2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #50177

Integrating Artificial Intelligence into Signals and Systems Education: Bridging Classical and Modern Approache

Yifan Wu, California State Polytechnic University, Pomona

Dr. Yifan Wu received his Ph.D. in Electrical and Computer Engineering from University of California, San Diego. He was a postdoctoral fellow in Colorado School of Mines, and he worked as visiting assistant professor at Harvey Mudd College. Currently, he is an assistant professor at department of Electrical and Computer Engineering in California State Polytechnic University, Pomona. His expertise includes signal processing, machine learning, optimization theory, and tensor algebra and applications. He is an active reviewer for more than 10 academic journals and conference proceedings. Email: yifanwu@cpp.edu

WIP: Integrating Artificial Intelligence into Signals and Systems Education: Bridging Classical and Modern Approaches

Abstract

Signals and Systems is a foundational course in electrical and computer engineering, yet students often perceive it as abstract and disconnected from modern engineering practice. While traditional instruction emphasizes mathematical tools such as convolution, Fourier transforms, and LTI system analysis, students frequently struggle to apply these concepts to emerging technologies like Artificial Intelligence (AI). This Work-in-Progress (WIP) paper describes the development and initial implementation of three AI-enhanced learning modules designed to bridge this gap by framing neural network architectures as a natural extension of classical signal theory.

The proposed modules include: (1) visualizing linear neural networks and encoder-decoder structures from an LTI system perspective; (2) implementing circulant matrices and convolutional layers using Python and MATLAB; and (3) a comparative "competition" experiment between classical Butterworth filters and denoising autoencoders. At this stage, the theoretical framework and module content have been established, and pilot data collection is currently underway.

The proposed evaluation strategy utilizes a multi-method approach to measure pedagogical impact. Conceptual mastery will be assessed through pre- and post-tests aligned with the Signals and Systems Concept Inventory (SSCI), while student engagement and perceived career relevance will be measured via a 5-point Likert scale survey. This paper details the design of the modules, the mathematical isomorphisms used to connect LTI systems to neural layers, and the preliminary qualitative observations from the initial pilot group. Final results and a comprehensive statistical analysis of the student performance data will be presented in a future full paper.

1 Introduction

Signals and Systems serves as the mathematical bedrock of electrical and computer engineering. Concepts such as Linear Time-Invariant (LTI) system theory, convolution, transform-domain analysis (Fourier, Laplace, and Z-transforms), and sampling are essential for understanding everything from communication systems to radar imaging. Traditionally, these courses emphasize rigorous analytical derivations to provide students with a deep understanding of system behavior in both time and frequency domains.

Despite its importance, this class is frequently cited by students as one of the most difficult and "abstract" courses in the undergraduate curriculum[1]. The heavy reliance on complex variable calculus and formal proofs often obscures the physical intuition and practical utility of the material. In the current educational landscape, students are increasingly motivated by "hands-on" applications, particularly in emerging fields like Data Science[2] and Artificial Intelligence (AI)[3]. When the curriculum remains strictly focused on theoretical analysis without connecting to these modern drivers, a "relevance gap" forms, leading to decreased student engagement and perceived difficulty.

The rise of Machine Learning (ML)[4], specifically Deep Learning[5], offers a unique opportunity to modernize Signals and Systems education[6,7,8]. At their core, many AI architectures are sophisticated extensions of classical signal processing concepts. For example, a Convolutional Neural Network (CNN)[9] is fundamentally a series of learnable filters performing the very "flip-and-slide" operations taught in Signals and Systems[10]. By framing a neural network as a high-dimensional LTI system (or a series of non-linear systems), educators can provide a concrete, modern "why" for the abstract "how" of classical theory.

Current ECE curricula often treat Signal Processing and AI as siloed disciplines, with AI relegated to upper-level electives. This separation prevents students from seeing the continuity of engineering principles. There is a critical need for pedagogical frameworks that integrate AI concepts directly into the foundational Signals and Systems course.

This paper addresses this need by introducing three AI-enhanced learning modules designed to bridge the gap between classical theory and modern implementation. By using Python and MATLAB-based experiments, these modules allow students to:

- Visualize convolution through the lens of neural network kernels.
- Understand system identification by training simple linear networks.
- Compare the robustness of classical filters against deep-learning-based signal reconstruction.

The goal of this integration is not to replace classical theory, but to reinforce it by showing that the "abstract" math of the 1800s is the engine driving the "intelligent" technologies of the 2020s.

2 Theoretical Framework: The Bridge

The integration of Artificial Intelligence (AI) into the Signals and Systems curriculum is grounded in the mathematical isomorphism between LTI systems and the architecture of Deep Learning. By framing neural layers as adaptive signal processors, we provide students with a familiar entry point into modern computational methods.

2.1 The Impulse Basis and the LTI Property

The "bridge" begins with the fundamental representation of a discrete sequence $x[n]$. As established in classical LTI theory, any sequence can be uniquely represented as a linear combination of shifted impulse sequences $\delta[n-k]$:

$$x[n] = \sum_{k} x[k]\delta[n-k] \tag{1}$$

This basis is orthonormal, satisfying the condition $\langle \delta[n-k], \delta[n-j] \rangle = 1$ if $k = j$ and 0 otherwise. Because an LTI system is characterized by linearity and time-invariance, the system output $y[n]$ is determined solely by its response to the unit impulse, denoted as the impulse response $h[n]$. This leads to the convolution sum:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n] \tag{2}$$

2.2 Matrix Representation: Systems as Networks

To connect this to Neural Networks, we transition from the summation form to the matrix-vector representation. An LTI system can be viewed as a linear operator $y = Hx$. For a system with finite support, the transformation matrix H is a **circulant matrix**, where each row is a cyclically shifted version of the impulse response $h[n]$.

This provides two distinct "Network Views" essential for understanding AI layers:

- **The Row View (Filtering):** The n-th output $y[n]$ is the dot-product of the input x and the n-th row of H (the reversed and shifted impulse response). In AI, this corresponds to the "receptive field" of a neuron.
- **The Column View (Synthesis):** The output y is a linear combination of the columns of H. Each column is a shifted version of $h[n]$. In AI, this represents the "contribution" of an input feature to the various outputs.

Neural networks exploit this structure by enforcing weight sharing; instead of a fully connected matrix with D^2 independent weights, a convolutional layer replicates a small set of weights (the kernel) across the matrix, directly mimicking the time-invariance of an LTI system.

2.3 Eigenvectors and the Fourier Basis

The bridge extends to the frequency domain by examining the "eigenfunctions" of LTI systems. A complex exponential $e^{j\omega n}$ passed through an LTI system remains a complex exponential of the same frequency, scaled by a complex value $H(\omega)$:

$$y[n] = H(\omega)e^{j\omega n} \quad \text{where} \quad H(\omega) = \sum_k h[k]e^{-j\omega k} \tag{3}$$

In this context, the Fourier basis functions $b_k[n]$ are the eigenvectors of the system, and the frequency response $H(\omega)$ represents the corresponding eigenvalues.

2.4 Systems vs. Networks Interpretation

As shown in Figure 1, we can interpret the network's operation as a similarity measure. Each neuron in a layer computes the similarity (dot-product) between the input signal $x[n]$ and a basis function $b_k[n]$. This "similarity" is then magnified or attenuated by the gain $g_k = H(\frac{2\pi}{D}k)$.

The system output is thus dominated by the frequencies that satisfy two conditions:

1. The input signal $x[n]$ contains significant energy at that frequency.
2. The system "prefers" that frequency (the gain $H(\omega)$ is large).

By training a neural network, students are essentially performing **Automated Filter Design**, where the network learns a frequency response $H(\omega)$ that amplifies features relevant to the task (e.g., edges in an image) while suppressing noise.

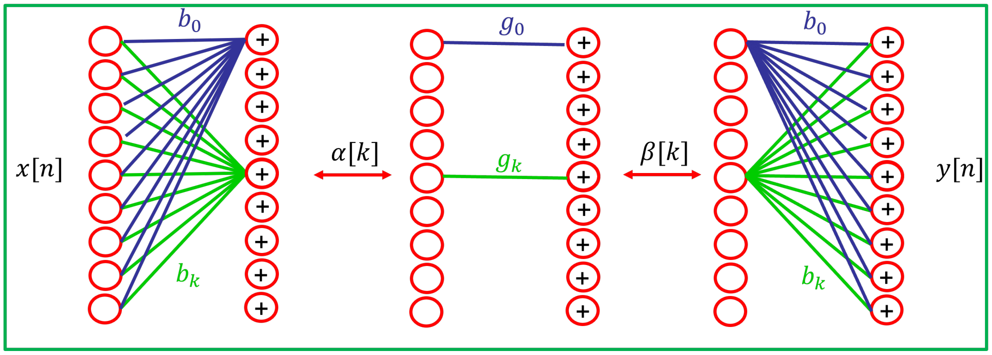

Figure 1: The illustration figure of the network's operation

Table 1: Conceptual Mapping: LTI Systems to Neural Networks

LTI System Concept	Network Equivalent	Physical Meaning
Impulse Response $h[n]$	Kernel Weights	The system's "memory" or "feature"
Convolution $x[n] * h[n]$	Forward Pass	The matching of patterns in a signal
Circulant Matrix H	Conv Layer	Enforced time-invariance and weight sharing
Eigenvalues $H(\omega)$	Spectral Gains	How the network filters specific patterns

3 Methodology

The pedagogical core of this work consists of three modules implemented in a laboratory setting. These modules are designed to reinforce classical LTI theory by demonstrating its utility in modern Artificial Intelligence frameworks.

3.1 Module 1: The Encoder-Decoder as a Sampling and Reconstruction System

This module visualizes the "Autoencoder" architecture through the lens of classical sampling and reconstruction theory. Students are taught to view the **Encoder** as a downsampling or dimensionality-reduction operator that maps a signal $x[n]$ to a compressed latent representation z. The **Decoder** is then presented as an interpolation or reconstruction filter that seeks to recover the original signal.

Drawing from the "Systems vs. Networks" perspective, students explore how the hidden layers act as a series of dot-products with basis functions. They analyze the *reconstruction error*—the difference between the input and the reconstructed output—as a function of the number of neurons (basis functions) used in the bottleneck. This mirrors the Nyquist-Shannon criteria, where insufficient “sampling” of the feature space leads to aliasing or loss of information in the reconstructed signal.

3.2 Module 2: Implementation via Python and MATLAB

To lower the barrier to entry, the modules utilize high-level programming environments familiar to ECE students. The setup is designed to be “plug-and-play,” requiring minimal installation of additional resources.

- **Python Environment:** Students utilize the `NumPy` and `SciPy` libraries to construct circulant matrices and perform classical convolution. For the neural components, `PyTorch` is used due to its “tensor-centric” design, which allows students to see the direct mapping between signal vectors and network layers.
- **MATLAB Environment:** Students leverage the `Signal Processing Toolbox` for classical filter design (Butterworth, FIR) and the `Deep Learning Toolbox` for training simple linear networks. MATLAB’s `deepNetworkDesigner` is specifically used to visualize the “flow” of signals through the LTI-equivalent layers.

3.3 Module 3: The “Competition” — Classical vs. Deep Learning Denoising

The final module involves a comparative computer experiment where students evaluate the trade-offs between hand-engineered filters and learned models.

Task: Denoise a signal (audio clip or sensor data) corrupted by additive white Gaussian noise (AWGN).

1. **The Classical Approach:** Students design a high-order *Butterworth Filter*. They must manually select the cutoff frequency ω_c and order N based on the spectral characteristics of the noise and signal.
2. **The AI Approach:** Students train a simple *Denoising Autoencoder* (DAE). Unlike the Butterworth filter, the DAE is not told the cutoff frequency; it must learn to “prefer” the signal’s frequency components and “reject” the noise by observing input-output pairs.

Students compare the results using both quantitative metrics (Signal-to-Noise Ratio improvement, Mean Squared Error) and qualitative analysis (audio playback or visual inspection of the residuals). This “competition” demonstrates that while classical filters are highly predictable and require no training data, deep learning models can adapt to complex noise profiles where a simple frequency cutoff may be insufficient.

4 Data Analysis Strategy

To evaluate the efficacy of the AI-enhanced modules in bridging the gap between abstract theory and practical application, a multi-faceted assessment strategy is proposed. This plan focuses on measuring three key dimensions: conceptual mastery of LTI theory, student engagement with modern engineering tools, and the perceived relevance of the curriculum.

4.1 Quantitative Assessment: Conceptual Mastery

The primary metric for student learning will be a series of pre- and post-tests aligned with the Signals and Systems Concept Inventory (SSCI)[1]. These assessments will specifically target the mathematical isomorphisms discussed in Section 2. Here is an example of pre-test and post-test assessment questions:

Part 1: Pre-Test (Classical LTI Framework)

Focus: Core mathematical properties of impulse bases, circulant matrices, and frequency preference.

1. **Impulse Basis Representation**
   Any discrete sequence $x[n]$ can be uniquely represented as a linear combination of shifted impulses $\delta[n-k]$. If $x[n] = [2, -1, 0, 3]$, what are the coefficients for the basis functions $\delta[n-0], \delta[n-1], \delta[n-2]$, and $\delta[n-3]$?
   (a) $2, -1, 0, 3$
   (b) $3, 0, -1, 2$
   (c) $1, 1, 1, 1$
   (d) $0, 0, 0, 0$
2. **Circulant Matrix Structure**
   An LTI system is characterized by an impulse response $h[n] = [1, 0.5]$. For an input of length $D = 4$, the system can be represented by a matrix H. Which property defines the structure of this matrix?
   (a) Every row is a replica of the input $x[n]$.
   (b) Every column is a cyclically shifted version of the impulse response $h[n]$.
   (c) The matrix is identity, meaning $y[n] = x[n]$.
   (d) The matrix is diagonal, containing only the values of $H(\omega)$.
3. **The Similarity Metric**
   When an LTI system computes the dot-product $\langle x[n], b_k[n] \rangle$ where $b_k[n]$ is a complex exponential basis function, the resulting value physically represents:
   (a) The total energy of the input signal.
   (b) The "similarity" or content of the signal at that specific frequency.
   (c) The phase shift of the system at $\omega = 0$.
   (d) The causal delay of the system.

Part 2: Post-Test (Neural Network Bridge)

Focus: Isomorphic application of LTI concepts to modern AI architectures.

1. **Convolutional Kernels as Impulse Responses**
   In a 1D Convolutional Neural Network (CNN) layer, a kernel $w = [3, -2, 1]$ is applied to an input. In classical signal processing, this kernel is mathematically equivalent to which of the following?
   (a) The system's frequency response $H(\omega)$.
   (b) The system's impulse response $h[n]$.
   (c) The system's transfer function $H(z)$.
   (d) The system's Region of Convergence (ROC).

2. **Weight Sharing and Time-Invariance**
   Neural networks use "weight sharing" in convolutional layers to reduce parameters. Which classical LTI property is being enforced by using a circulant weight matrix with shared coefficients?
   (a) Linearity
   (b) Stability (BIBO)
   (c) Time-Invariance
   (d) Causality

3. **Autoencoders and Frequency Preference**
   A Denoising Autoencoder learns to reconstruct a signal from noisy data. If the autoencoder successfully filters out high-frequency white noise, what can we conclude about the "gains" g_k of its internal basis functions?
   (a) The gains for high-frequency basis functions are near zero.
   (b) The gains for all frequencies are exactly 1.0.
   (c) The gains for low-frequency basis functions are zero.
   (d) The gains are randomly distributed regardless of frequency.

Statistical Method: A paired t-test will be conducted on the scores to determine if the improvement in conceptual understanding is statistically significant. The null hypothesis (H_0) assumes no difference in mean scores before and after the AI modules. We define the t-statistic as:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{4}$$

where $\bar{d}$ is the mean of the differences between post- and pre-test scores, s_d is the standard deviation of those differences, and n is the number of students. To measure the magnitude of the intervention's impact, Cohen's d will be calculated to provide an effect size, where a value > 0.8 indicates a large practical significance.

4.2 Qualitative Assessment: Student Perception and Engagement

To capture the "relevance gap" mentioned in the Problem Statement, a 5-point Likert scale survey will be administered. This survey focuses on student confidence in applying classical LTI theory to modern technologies.

Sample survey items include:

- "I can explain the physical meaning of convolution more clearly after seeing it implemented as a neural network kernel."
- "The use of Python/PyTorch made the abstract mathematical concepts of this course feel more applicable to my future career."
- "I feel confident in choosing between a classical filter and an AI-based approach for a specific denoising task."

4.3 Project-Based Evaluation

In Module 3 (The Competition), student performance will be evaluated using a technical rubric. This rubric measures the ability of students to perform "System Identification"—comparing the learned frequency response of their autoencoder against the designed response of a Butterworth filter.

Success will be quantified by the students' ability to:

1. Correctly visualize and interpret the circulant matrix of their neural layer.
2. Quantify denoising performance using Signal-to-Noise Ratio (SNR) improvements.
3. Justify their choice of architecture based on the spectral characteristics of the input signal.

4.4 Preliminary Observations

While formal data collection is ongoing, initial pilot observations suggest a high level of "aha!" moments. For instance, students who struggled with the "flip-and-shift" graphical method of convolution reported that visualizing the operation as a sparse, replicated weight matrix in Python (the "Network View") made the time-invariance property of LTI systems immediately intuitive.

5 Conclusion

This Work-in-Progress paper presented a pedagogical framework designed to modernize the undergraduate Signals and Systems curriculum by establishing a formal mathematical bridge to Artificial Intelligence. By reframing LTI theory through the lens of neural network architectures, we provide students with a concrete application for abstract concepts such as the impulse basis, circulant matrices, and frequency-domain similarity.

The preliminary development of the three learning modules—ranging from matrix-based convolutional layers to competitive denoising experiments—demonstrates that modern AI topics can be integrated into the ECE curriculum without sacrificing the mathematical rigor of classical signal theory. Initial qualitative observations from pilot sessions suggest that this "Systems vs. Networks" perspective helps demystify the "black box" nature of deep learning while simultaneously reinforcing the utility of LTI system properties.

The next phase of this research will focus on a full-scale longitudinal study. Future work will involve:

- **Statistical Validation:** Conducting a formal paired t-test analysis on a larger student cohort ($n > 30$) to quantify improvements in conceptual mastery via the Signals and Systems Concept Inventory (SSCI).
- **Longitudinal Tracking:** Assessing whether students who complete these modules show increased self-efficacy and performance in subsequent senior-level elective courses such as Digital Signal Processing and Machine Learning.
- **Cross-Institutional Deployment:** Adapting the Python-based computer experiments for use at peer institutions to evaluate the portability of the modules across different student demographics.

Ultimately, this effort seeks to narrow the "relevance gap" in ECE education, ensuring that the next generation of engineers views classical signal processing not as a relic of the past, but as the essential mathematical language of modern artificial intelligence.

References

[1] Kathleen E Wage, John R Buck, Cameron HG Wright, and Thad B Welch. The signals and systems concept inventory. *IEEE Transactions on Education*, 48(3):448–461, 2005.

[2] Longbing Cao. Data science: a comprehensive overview. *ACM Computing Surveys (CSUR)*, 50(3):1–42, 2017.

[3] Earl B Hunt. *Artificial intelligence*. Academic Press, 2014.

[4] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349 (6245):255–260, 2015.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] Kenan Baltaci, Monika Herrmann, and Ahmet Turkmen. Integrating artificial intelligence into electrical engineering education: A paradigm shift in teaching and learning. In *2024 ASEE Annual Conference & Exposition*, 2024.

[7] Thomas Franklin Hallmark. Integrating artificial intelligence in engineering education: A work-in-progress systematic review of applications and challenges. In *2025 ASEE-GSW Annual Conference*, 2025.

[8] Gerald Fudge, Anika Rimu, William Zorn, July Ringle, and Cody Barnett. Teaching machine learning to undergraduate electrical engineering students. *Computers*, 14(11):465, 2025.

[9] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020*

international conference on emerging trends in information technology and engineering (ic-ETITE), pages 1–5. IEEE, 2020.

[10] Michael Unser. A representer theorem for deep neural networks. *Journal of Machine Learning Research*, 20 (110):1–30, 2019.