2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51184

WIP: Bringing Modern Communication Systems to Juniors: A Work-in-Progress Course Using Python and SDR

Prof. Thomas Moon, University of Illinois at Urbana - Champaign

Thomas Moon received the B.S. degree in electrical electronic engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2008, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA in 2015. Between 2015 and 2017, he worked at IBM in Burlington, Vermont where he developed mmWave test equipment as a principle development engineer. He joined Coordinated Science Lab, University of Illinois at Urbana-Champaign (UIUC), IL, USA in 2017 as a post-doctoral researcher. He has been a Teaching Assistant Professor at Department of Electrical and Computer Engineering at UIUC. His current research interests include wireless sensing and communication in mmWave.

Juan Alvarez, University of Illinois at Urbana - Champaign

Juan Alvarez joined the Department of Electrical and Computer Engineering at University of Illinois faculty in Spring 2011 and is currently a Teaching Assistant Professor. Prior to that, he was a Postdoctoral Fellow in the Department of Mathematics and Statistics at York University, Canada, a Postdoctoral Fellow in the Chemical Physics Theory Group at the University of Toronto, Canada, and a Postdoctoral Fellow in the Department of Mathematics and Statistics at the University of Saskatchewan. He obtained his Ph.D. and M.S. from the Department of Electrical and Computer Engineering at the University of Illinois in 2004 and 2002, respectively. He teaches courses in communications, signal processing and probability.

WIP: Bringing Modern Communication Systems to Juniors: A Work-in-Progress Course Using Python and SDR

Abstract

This Work-in-Progress paper introduces a newly developed junior-level course designed to expose students to real-world communication systems earlier in the curriculum. The course blends foundational theory in analog and digital communication with hands-on lab experience using Python and Software-Defined Radio (SDR). Weekly modules include paper-based quizzes and lab assignments that transition from simulation to SDR-based implementation, covering tasks such as scanning RF spectrum, decoding FM broadcasts, and sending/receiving text messages. To evaluate the course's effectiveness, three early-stage assessment methods are employed: (1) student surveys measuring engagement and perceived learning; (2) performance-based metrics from lab milestones; and (3) cross-course enrollment tracking to monitor how many students pursue senior-level communication electives after taking this course. Results from the Spring 2025 pilot will be presented, highlighting student feedback and preliminary enrollment trends. These findings will inform future course improvements, including scaffolding theory, refining lab instructions, and enhancing support for debugging SDR-based projects. This paper is submitted under the curriculum design topic and is part of a long-term plan to develop a full paper that explores pedagogical strategies for integrating hands-on SDR learning into the undergraduate communications curriculum.

1 Introduction

The rapid growth of wireless mobile devices has significantly influenced both industry and academia. Wireless communication now boosts many emerging technologies in the fourth industrial revolution, including smart manufacturing, sensor networks, autonomous systems, cloud computing, and AI-driven applications. As wireless connectivity becomes increasingly central to modern engineering systems, there is a growing need to prepare students with both foundational knowledge and practical skills in communication technologies.

However, in the Electrical and Computer Engineering (ECE) program at our institution, dedicated communication systems courses are introduced relatively late in the curriculum, typically at the senior or even early graduate level. This delay—combined with the heavy prerequisite load in mathematics and signals—can make these courses difficult to access and less appealing to undergraduates. As a result, students may miss the opportunity to explore communication systems early enough to inform their academic interests or career paths.

Prior work [1, 2, 3, 4] has demonstrated that software-defined radio (SDR) can effectively support communication and signal processing education and enable project-based learning in upper-level courses. Lab-centric communication courses using SDR have also been shown to improve students' understanding and motivation [5, 6, 7, 8, 9, 10]. These efforts, however, primarily target senior-level or graduate students who already possess substantial background in communications and mathematics.

To address this difficulty, we have developed a new junior-level elective in communication systems with Python and SDR (hereafter, IntroSDR) designed to provide early exposure to core communication concepts through a blend of theory and hands-on lab projects. The course integrates Python programming and SDR hardware to build student intuition through real-world applications such as spectrum scanning, FM radio demodulating, and wireless text messaging. This structure aims to lower the barrier to entry, improve engagement, and align the course with industry-relevant tools and practices.

The course targets sophomore and junior ECE students who have completed basic coursework in signals and systems or digital logic. By introducing communication systems at this stage, we aim to scaffold foundational knowledge ahead of senior-level electives, while increasing student confidence and interest in the field.

This paper presents the motivation, design, and early implementation of the course, along with preliminary feedback from its first offering. We also discuss the challenges encountered in adapting communication theory and labs for a junior audience, and describe our assessment methods, including student surveys, performance-based metrics from the labs, and cross-course enrollment tracking, to evaluate the course's impact.

2 Course Design Overview

2.1 Course Structure

The topics of the course are organized into three main parts:

1. **Analog communication and basics of signal processing:** review of fundamental concepts in continuous- and discrete-time signal processing and linear systems, a brief preview of probability theory, and an introduction to analog modulation and demodulation.
2. **Digital communication:** introduction to key principles of digital communication, including baseband and passband representation, symbol mapping, pulse shaping, and basic receiver structures.
3. **Applications of modern communication:** exploration of modern wireless applications and IoT technologies, with an emphasis on practical use cases such as WiFi, Bluetooth, RFID, and LoRa.

Table 1 summarizes the week-by-week course schedule and associated topics. The course meets three times per week (Monday, Wednesday, and Friday) and follows a consistent structure

Table 1: Course schedule and topics.

Topic category	Week	Lecture topics (MW)	Lab / Project topics (F)
Analog comm	1	Signal processing review	Lab1: Setup Python and PlutoSDR
	2	AM modulation (DSB)	Lab2: DSB-SC, DSB-TC
	3	Baseband and bandpass signals, Hilbert transform	SDR Project 1: RF Scanner
	4	SSB, VSB, QAM	Lab3: SSB
	5	FM and PM	Lab4: FM
	6	Basic probability review	SDR Project 2: FM Stereo Transceiver
Digital comm	5	Symbol, BPSK, Pulse-shaping	Lab5: Pulse-shaping
	6	Matched-filter, Symbol-timing recovery	Lab6: Matched-filter, Symbol-timing recovery
	7	*Spring break*	*No lab*
	8	Channel, DBPSK, Frame-sync	Lab7: DBPSK, Frame-sync
	9	Signal space, ASK, FSK	Lab8: FSK
	10	Channel estimation/correction, CFO	Lab9: CFO and Channel Correction
	11	SNR vs BER, M-ary QAM and PSK	Lab10: M-ary QAM
Applications	12	OFDM, WiFi	SDR Project 3: Text message by OOK and 16QAM
	13	LoRa, Bluetooth, RFID, and course summary	SDR Project 3 continued
	14	*No lecture*	SDR Project 3 Demo

designed to support a simulation-to-hardware learning progression.

The lecture meetings are divided into two phases:

- **Concept Lectures (Monday and Wednesday):** These sessions focus on theoretical concepts, mathematical tools, and system-level ideas in communication and signal processing. The emphasis is on building intuition and connecting new topics to prior knowledge from signals and systems and basic circuits.

- **Lab-Integrated Session (Friday):** This session is organized around the upcoming lab or project. The instructor reviews the lab objectives, walks through key implementation details, and demonstrates relevant examples in Python or on the SDR hardware. Each Friday meeting also includes a short (approximately 10-minute) quiz to assess students' understanding of the current week's concepts and to provide immediate feedback.

This structure is intended to create a weekly rhythm in which students first encounter new ideas in a traditional lecture format, then see those ideas instantiated in concrete lab activities, and finally demonstrate their understanding through quizzes and laboratory work. The regular integration of lab preparation into the Friday session also reduces the overhead of students “figuring out” the lab on their own, which is particularly important for a junior-level course in which students may be working with SDR hardware for the first time.

2.2 Course Contents

The main course components include weekly Python-based labs, three SDR projects, and weekly lecture quizzes.

In a typical week (approximately 10 out of 14 weeks), students complete:

- a **Python lab** that reinforces the lecture material through simulation and signal visualization, and
- a **lecture quiz** administered during the Friday session.

In the remaining weeks (approximately 4 out of 14 weeks), students work on an **SDR project** instead of a Python lab. In these projects, they transfer algorithms developed in Python to the PlutoSDR platform and perform experiments with over-the-air signals. This simulation-to-hardware progression is a core design principle of the course: students first gain confidence in a controlled environment and then apply the same ideas to real RF signals.

The three SDR projects are designed to align with the course topics. For example, a kick-off SDR introductory project may involve scanning and characterizing the local RF spectrum, an analog communication project may focus on FM radio demodulation, and a digital communication project may implement a simple QAM link for text messaging. Weekly quizzes emphasize conceptual understanding and basic computations rather than lengthy derivations, which is appropriate for a junior-level audience that is still building its mathematical foundation.

In the following subsections, we illustrate the design of each module type with a representative example.

2.2.1 Example Python Lab

A typical Python lab begins with a short, guided notebook that reviews the relevant mathematical relationships (e.g., complex exponentials, spectra, sampling) and then leads students through progressively more open-ended tasks. Throughout the labs, students are encouraged to annotate their code, interpret plots, and connect their observations back to the concepts discussed in lecture.

As an example, an early lab focuses on single-sideband (SSB) amplitude modulation. In lecture, students learn that SSB signals occupy only half of the bandwidth of a conventional double-sideband (DSB) signal while still conveying the same information. The lab guides students to implement SSB modulation in Python, visualize the resulting signals in both time and frequency, and compare upper- and lower-sideband spectra. Listing 1 shows a snippet of the provided Python code, and Figure 1 illustrates a representative student result demonstrating the spectra of upper-SSB and lower-SSB signals.

```python
# Demodulate using synchronized carrier
phase_offset = 0
carrier_rx = np.cos(2 * np.pi * fc * t + phase_offset)

# Apply low-pass filter to remove high-frequency components
b, a = butter(15, Wn = fs_audio, btype = 'low', fs = fs)

# Demodulate by multiplying carrier_rx signal
ssb_demod = # Your code here

# Pass sc_demod through the low-pass filter (use filtfilt)
ssb_demod_filt = # Your code here
ssb_demod_filt_fft = # Your code here
```

Listing 1: A snippet of the provided Python code in SSB lab.

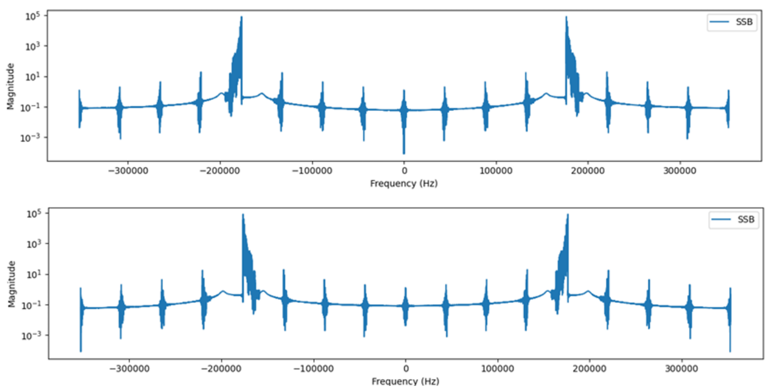


Figure 1: Example output of SSB lab.

The same lab also introduces practical receiver considerations. Students implement an SSB demodulator to recover an audio signal and then investigate the effect of phase mismatch between the transmitter and the receiver local oscillator. They vary the receiver phase offset from 0 to 2π, listen to the demodulated audio, and reason about when and why the signal remains intelligible. This structure reinforces the connection between mathematical models and perceptible system behavior.

The following are example lab assignments from this module:

- **Assignment 1:**
  - Implement and submit the complete SSB modulation code.
  - Plot the time-domain waveforms and magnitude spectra of both upper-sideband (USB) and lower-sideband (LSB) signals.
- **Assignment 2:**

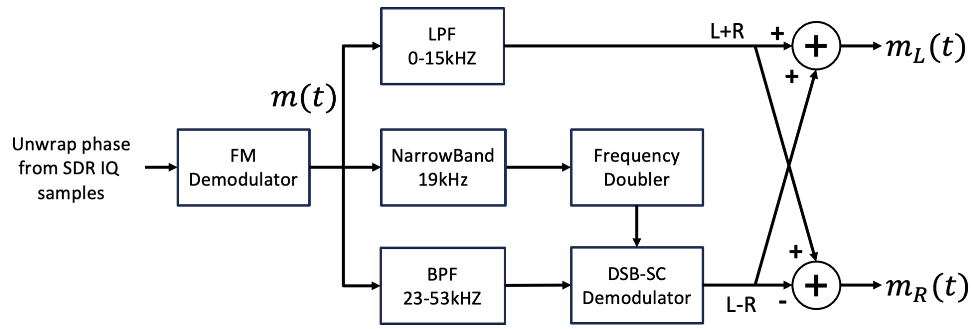


Figure 2: Main components of FM receiver in the lab handout.

- Vary the receiver phase offset to observe its effect on the demodulated signal.
- Discuss the changes observed in the time-domain waveforms and in the perceived audio quality.
- If the audio remains intelligible for certain phase offsets, explain why based on the underlying signal model.

2.2.2 Example SDR Project

An example SDR project focuses on implementing an FM radio transmitter (Tx) and receiver (Rx). Students begin from a Python-only simulation of frequency modulation and demodulation developed in earlier labs, and then adapt their code to interface with the PlutoSDR hardware. In the first stage, they implement an FM transmitter and use a loopback configuration, connecting the PlutoSDR transmit and receive ports with an SMA cable. Students configure the SDR to use a common carrier frequency for both Tx and Rx, generate an FM-modulated signal at the transmitter, capture complex baseband samples at the receiver, and implement the necessary processing blocks to recover the audio signal (e.g., resampling, filtering, demodulation, and audio playback). Figure 2 shows an example block-diagram-style figure from the lab handout that outlines the main components of the FM receiver.

After successfully recovering the audio signal in the controlled loopback setup, students proceed to over-the-air reception of commercial FM broadcasts. In this second stage, they replace the SMA loopback cable with an RF antenna and configure the PlutoSDR to tune to a local FM station. They then extend their receiver implementation to handle a real broadcast signal, which includes stereo and additional pilot signals.

The project requires students to address several practical issues, such as adjusting gain settings, tuning the carrier frequency, and locking onto the pilot tone in stereo FM. At the same time, it reinforces theoretical concepts introduced in lecture, including the Hilbert transform, frequency modulation and demodulation, phase unwrapping, and the design and use of digital filters. By the end of the project, students have implemented a complete end-to-end FM radio system and experienced firsthand how abstract signal processing operations manifest in a realistic communication scenario.

2.2.3 Example Lecture Quiz

The weekly lecture quiz typically consists of a small number of short/long-answer or multiple-choice questions that probe students' understanding of the key ideas from that week. For example, a quiz in the analog communication unit may ask students to identify the spectrum of a given amplitude-modulated signal, sketch the instantaneous frequency of phase modulated signal, or identify the in-phase and quadrature components of a bandpass signal. In the digital communication unit, quizzes may involve interpreting constellation diagrams or reasoning about symbol timing. These quizzes serve both as formative assessment tools and as a mechanism to keep students engaged with the material on a weekly basis.

3 Preliminary Results

3.1 Student Survey

An anonymous student survey was conducted during the last month of the semester. Nine out of thirteen students responded to the survey.

We first asked about students' prior coursework in digital signal processing (DSP Core) and probability (Prob Core), neither of which is a prerequisite for this course. Among the nine respondents, four had not taken DSP Core, and the same four students had also not taken Prob Core. We also inquired whether students had taken any upper-level communication courses (Comm Theory, Wireless Comm, Digital Comm, Comm Lab); seven out of nine respondents had not taken any of these courses. Figure 3 summarizes the distribution of students' prior background.

To gauge the perceived difficulty of the course, we asked students to rate the following items on a five-point Likert scale (1 = much too light / easy, 5 = much too heavy / difficult):

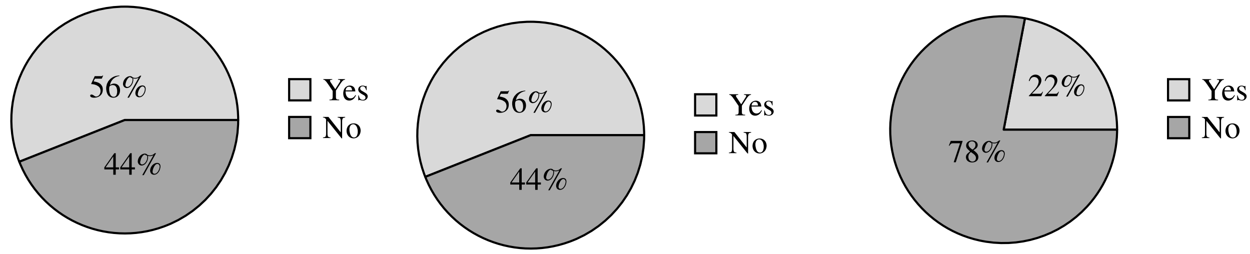
- Q1. How manageable do you find the overall workload?
- Q2. How manageable do you find the difficulty of labs?
- Q3. How manageable do you find the difficulty of SDR projects?
- Q4. How manageable do you find the difficulty of quizzes?

The response distributions for these four items are shown in Figure 4. Although the sample size is small, the results suggest that most students viewed the workload and difficulty as between "manageable" and "somewhat heavy," with few extreme responses.

We also included two open-ended questions:

- Q5. What aspects of the course do you find most helpful?
- Q6. What aspects of the course do you find most challenging?

For Q5, most students highlighted the hands-on elements: five of nine respondents mentioned the Python labs, and three mentioned the SDR projects as especially helpful. Several students (two of



Previously taken DSP Core

Previously taken Prob Core

Previously taken upper-level comm courses

Figure 3: Distribution of prior coursework among surveyed students.

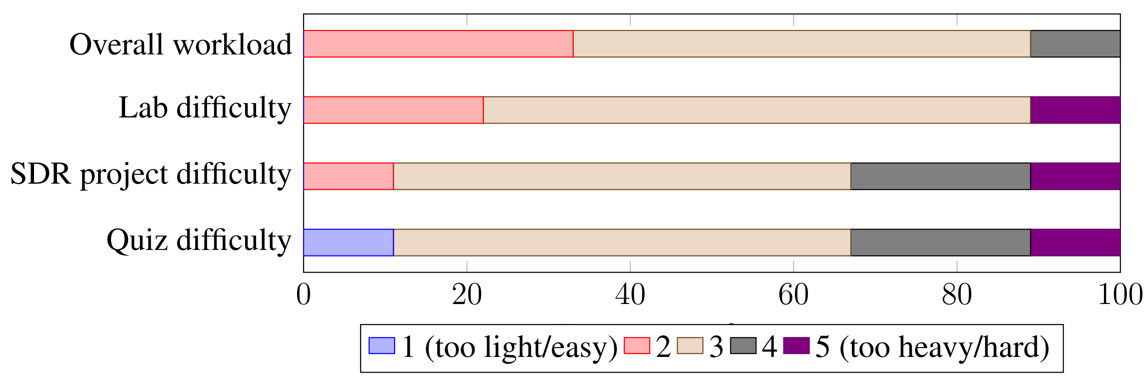


Figure 4: Distribution of student responses to four workload-related Likert items (1–5).

nine) also noted that concise lectures combined with practical demonstrations were useful in preparing for the labs.

For Q6, four of nine respondents pointed to the difficulty of the early quizzes as a primary challenge. A few students (two of nine) also commented that the analog modulation concepts were more challenging to grasp than the digital communication material later in the course.

3.2 Performance-Based Metrics

In addition to the survey data, we examined performance-based metrics from the weekly labs and SDR projects. For each major outcome, student submissions were categorized as *not completed*, *partially completed*, or *fully completed*. Table 2 summarizes the distribution across the students (note that the total number of students varies by assignment due to course drops during the semester).

Overall, the majority of students were able to complete the core analog and digital communication tasks. In contrast, fewer students fully completed the more advanced text-messaging project using on–off keying (OOK) and quadrature amplitude modulation (QAM).

Table 2: Summary of performance-based lab outcomes (n=13-15).

Outcome	Not completed	Partial	Completed
Implement AM (DSB) systems in Python	2	2	11
Implement AM (SSB) systems in Python	1	1	11
Implement FM, PM systems in Python	1	3	8
Implement non-coherent digital comm system in Python	3	3	6
Implement coherent digital comm system in Python	1	1	11
Implement a working RF scanner on PlutoSDR	1	2	11
Implement a working FM receiver on PlutoSDR	0	3	10
Implement a digital text-messaging link on PlutoSDR	0	7	6

3.3 Cross-course Enrollment

We also performed a preliminary cross-course enrollment analysis to examine whether students who took IntroSDR subsequently enrolled in upper-level communication courses. Among the 13 students in the Spring 2025 cohort, four (31%) enrolled in at least one 400-level communication course (Comm Theory, Wireless Comm, Comm Lab, and Digital Comm) in Fall 2025 and Spring 2026: 2 in Comm Theory, 2 in Wireless Comm, 3 in Comm Lab, 2 in Digital Comm (some students appear in more than one course).

Because the observation window covers only two subsequent semesters and the initial offering of IntroSDR included several seniors (eleven of thirteen, with the remaining two being juniors), these numbers should be interpreted as an early baseline rather than a definitive measure of long-term impact. We plan to track additional cohorts and later semesters to obtain a more complete picture of how the course influences students' choices of advanced communication electives.

4 Early Insights and Future Directions

The first offering of IntroSDR provided several early insights into how junior-level students engage with an SDR-based introductory communication course, as well as directions for improving future iterations.

First, although approximately 50% of the students had prior coursework in digital signal processing and probability, many struggled with the early quizzes on analog communication. It is not yet clear whether this difficulty reflects the intrinsic challenge of the analog material, the specific design of the quiz items, or a combination of both. In any case, the results indicate that additional support is needed at the beginning of the course. Potential strategies include adding more scaffolded examples in lecture, and integrating short in-class exercises that reinforce key analog concepts.

Second, in contrast to the generally solid performance on the core labs and early SDR tasks, fewer students were able to fully complete the later stages of the SDR project sequence. In particular, only about half of the students implemented a fully functional digital text-messaging link using OOK and QAM. This suggests that the most advanced project may currently sit at or slightly beyond the comfort zone of many students within the available timeframe. In future offerings, we plan to introduce additional intermediate checkpoints, provide template code for non-central components (e.g., plotting or I/O), and allocate more dedicated project work time to help students reach a complete end-to-end implementation.

Third, despite these challenges, student feedback on the overall course design was largely positive. Many respondents highlighted the Python labs as especially helpful for visualizing theoretical concepts and experimenting with communication components in a controlled environment. Students also expressed satisfaction with the hands-on experience of implementing real-world examples on SDR hardware, such as spectrum scanning and FM radio reception. These comments suggest that the core idea of combining Python-based simulation with SDR-based practice is effective in engaging students and making abstract concepts more concrete.

Looking ahead, we plan to offer IntroSDR every spring semester and to continue collecting both survey data and performance-based metrics from each cohort. We also intend to extend the cross-course enrollment analysis to multiple subsequent semesters in order to obtain a clearer picture of how the course influences students' choices of upper-level communication electives. On the content side, we are developing additional lab and project modules, such as Automatic Dependent Surveillance–Broadcast (ADS-B) signal reception and a simple digital image transmission link, to broaden the range of application examples. As the course matures, we aim to refine the balance between conceptual depth and project complexity and to explore more systematic assessment methods for evaluating its long-term impact on student learning and curricular pathways.

References

[1] Sharlene Katz and James Flynn. Using software defined radio (sdr) to demonstrate concepts in communications and signal processing courses. In *2009 39th IEEE Frontiers in Education Conference*, pages 1–6. IEEE, 2009.

[2] Alexander M Wyglinski and Daniel J Cullen. Digital communication systems education via software-defined radio experimentation. In *2011 ASEE Annual Conference & Exposition*, pages 22–502, 2011.

[3] Ismail Guvenc, Natalie Paul, Rhys Bowley, and Maria Lorelei Fernandez. Hands-on learning of wireless communication principles using software-defined radio experiments and labview. In *2016 ASEE Annual Conference & Exposition*, 2016.

[4] Patrick Cutno, Chi-Hao Cheng, Zhiqiang Wu, Bin Wang, and Deng Cao. Software defined radio for digital signal processing related courses. In *2016 ASEE Annual Conference & Exposition*, 2016.

[5] Sven Bilen. Implementing a hands on course in software defined radio. In *2006 Annual Conference & Exposition*, pages 11–724, 2006.

[6] Shiwen Mao, Yingsong Huang, Yihan Li, Prathima Agrawal, and Jitendra K Tugnait. Introducing software defined radio into undergraduate wireless engineering curriculum through a hands-on approach. In *2013 ASEE Annual Conference & Exposition*, pages 23–822, 2013.

[7] Shiwen Mao, Yingsong Huang, and Yihan Li. On developing a software defined radio laboratory course for undergraduate wireless engineering curriculum. In *2014 ASEE Annual Conference & Exposition*, pages 24–947, 2014.

[8] Sabih Guzelgoz and Hüseyin Arslan. A wireless communications systems laboratory course. *IEEE Transactions on Education*, 53(4):532–541, 2009.

[9] Cory J Prust. An introductory communication systems course with matlab/simulink-based software-defined radio laboratory. In *2019 ASEE Annual Conference & Exposition*, 2019.

[10] Yair Linn. An ultra low cost wireless communications laboratory for education and research. *IEEE Transactions on Education*, 55(2):169–179, 2011.