MOOC Adventures in Signal Processing

Bringing DSP to the era of massive open online courses

Thomas A. Baran,
Richard G. Baraniuk,
Alan V. Oppenheim,
Paolo Prandoni, and
Martin Vetterli

In higher education circles, 2012 may be known as the "year of the MOOC"; the launch of several high-profile initiatives, both for profit (Coursera, Udacity) and not for profit (edX), created an electrified feeling in the community, with massive open online courses (MOOCs) becoming the hottest new topic in academic conversation. The sudden attention was perhaps slightly forgetful of many notable attempts at distance learning that occurred before, from campus TV networks to well-organized online repositories of teaching material. The new mode of delivery, however, was ushered in by a few large-scale computer science courses, whose broad success triggered significant media attention [1].

In the debate, some were quick to predict the end of traditional brick-and-mortar universities and marveled at the eagerness with which hallowed institutions wanted to be in the game. Some hailed the free and open MOOCs as the great equalizer, capable of providing underprivileged learners around the world with access to the

Digital Object Identifier 10.1109/MSP.2016.2556004
Date of publication: 1 July 2016

©ISTOCKPHOTO.COM/MILINDRI

highest-quality education. Of course, after a few sobering years of experience, neither extreme actually happened; in this article, we share our collective opinion on this topic.

Introduction

As teachers of various signal processing courses, we felt intrigued by the opportunity to bring our subject to this new teaching paradigm. Of course, many online resources for signal processing education exist already, including video courses developed in the 1980s by the Massachusetts Institute of Technology (MIT) Center for Advanced Engineering Studies [2], [3], various courses available on MIT's OpenCourseWare (OCW), a variety of online interactive demos, hands-on tutorials and code samples, and, of course, many collections of lecture notes and slides. Still, there was no interactive, full-fledged online course on signal processing available as the MOOC revolution was unfolding. This presented an interesting challenge and opportunity.

We approached online teaching from different backgrounds and with different perspectives, and we all started from our respective residential classes that range from mandatory undergraduate-, to master of science- and doctoral-level courses. Specifically, during the last three years, we created three MOOCs based on the following residential courses:

- Signal Processing for Communications (COM303), a mandatory undergraduate course taken in the third year at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, became Digital Signal Processing (DSP) on Coursera and has been offered five times since spring 2013.
- Discrete-Time Signals and Systems (ELEC301), a mandatory undergraduate course taken in the second/third year at Rice University, is the basis for the corresponding edX course offered twice since fall 2014.
- Discrete-Time Signal Processing (6.341), a first-year graduate course at MIT that is the basis for 6.341x, made available on edX and offered twice since fall 2014: once in a limited industrial beta format and once as a fully open online course.

The goal of this article is to relate the experience gained in moving "on-campus" material to the MOOC format and share the lessons learned, the influence this has had on our on-campus teaching. We hope that our MOOC experiences are valuable to the signal processing community.

EPFL's Digital Signal Processing on Coursera

Overview and goals

The course Digital Signal Processing, by Paolo Prandoni and Martin Vetterli, was first offered on the Coursera platform in February 2013. At the time of this writing, the class has completed its fifth edition. We authors are with the EPFL, and the course is based on the residential class COM303 offered by the Communication Systems Department to third-year undergraduates. For many SysCom students at EPFL, COM303 represents the first exposure to a higher-level engineering class after two years focused primarily on introductory subjects. COM303 lists calculus and linear algebra as prerequisites and recommends familiarity with probability theory as well. The class is based on the freely available textbook, *Signal Processing for Communications* [4], that we have written.

COM303 is a standard undergraduate-level DSP class with a slight emphasis on telecommunication systems. The syllabus starts off in the discrete-time domain and uses vector spaces and linear algebra as the framework to introduce signals and signal transforms; as subsequent topics are introduced, the goal is to strike a balance between solid mathematical foundations and practical applications. When adapting COM303 to the online medium, we decided to closely mirror the residential class. We did this for two reasons: primarily, we wanted to produce a package that, although aimed at the general public, would retain its focus on theoretical foundations rather than deliver yet another hands-on approach to applied DSP, for which countless tutorials are available on the Internet. Additionally, we wanted to experiment with the concept of the "flipped classroom" and be able to minimize standard lecturing to the advantage of more question-and-answer (Q&A) interaction with on-campus students.

Course organization

Outline

COM303 is composed of 17 lecture days that occur during nine weeks, and, as shown in Table 1, it is structured around nine thematic modules; each module is split into a varying number of small units (the actual videos) with the intent of balancing the conflicting requirements of a fine-grained subdivision of the material with the "narrative" needed to provide reasonably self-contained mini-lectures. Each lecture day provides students with the following:

- three video units (with associated slides; we should mention that we found it very difficult to produce videos lasting ten minutes or fewer, as per the recommended best practices, and average video length is 17 minutes)

Table 1. The syllabus for the EPFL course.

Module	Number of Units
Introduction	1
Discrete-Time Signals	3
Hilbert Space	3
Fourier Analysis	10
Linear Filters	12
Interpolation and Sampling	6
Stochastic SP	3
Image Processing	6
Digital Communication Systems	6

- additional material in the form of a numerical example or a mini-lecture on signal processing applications (see the section "Course Evolution")
- an automatically graded homework set; the passing grade for the class is determined from the cumulative homework score.

Overall, the course delivers 14 hours of video lectures (using approximately 1,300 slides) and 126 graded quizzes.

Style and format

It is said that Pythagoras would impart his lectures hiding behind a curtain, so that his students would concentrate solely on his words. His teaching style was called *acousmatic*, a word indicating an intelligible sound whose source remains unseen. In the same spirit, we decided to produce streamlined video lectures by pairing a slideshow with a simple voiceover; dynamic annotations drawn by hand are used to underscore key passages and elucidate derivations. Production-wise, this choice also enabled us to record and edit the audio in an efficient way before "filming" the video annotations. Great effort has been placed into the design of a large number of illustrations. To achieve a "coherent visual grammar" in the illustrations, we designed a LaTeX package called DSPTricks to efficiently draw one-dimensional (1-D) and two-dimensional DSP figures in PostScript [5]. The package is a high-level graphics toolbox that programmatically produces parametric images from within the LaTeX document and allows users to easily repurpose and modify their illustrations; we refer readers to the supplementary material that appears in IEEE *Xplore*. Finally, to preserve the "human component" of the class, brief introductions and closing remarks are added to each module in the form of short videos, in which the instructor appears in person.

Homework and grading

Homework and exercises present multiple challenges to teaching signal processing online because theoretical proofs and free-form derivations remain beyond the scope of automated graders (although some progress is underway [S2], [6]); whereas peer review may work effectively in less technical subjects, we believe that in our case the pool of students with sufficient mastery of the subject would be too small to ensure the necessary critical mass. As a consequence, we only designed graded homework with either multiple-choice or numerical answers. This somewhat limits the palette of questions that can be effectively formulated but has the advantage of providing unambiguous results, which is important given that the passing criterion for the class is based exclusively on homework grades. To complement the homework, we provide solved problem sets, in which we tackle more articulated questions whose answers require derivations and proofs.

Course evolution

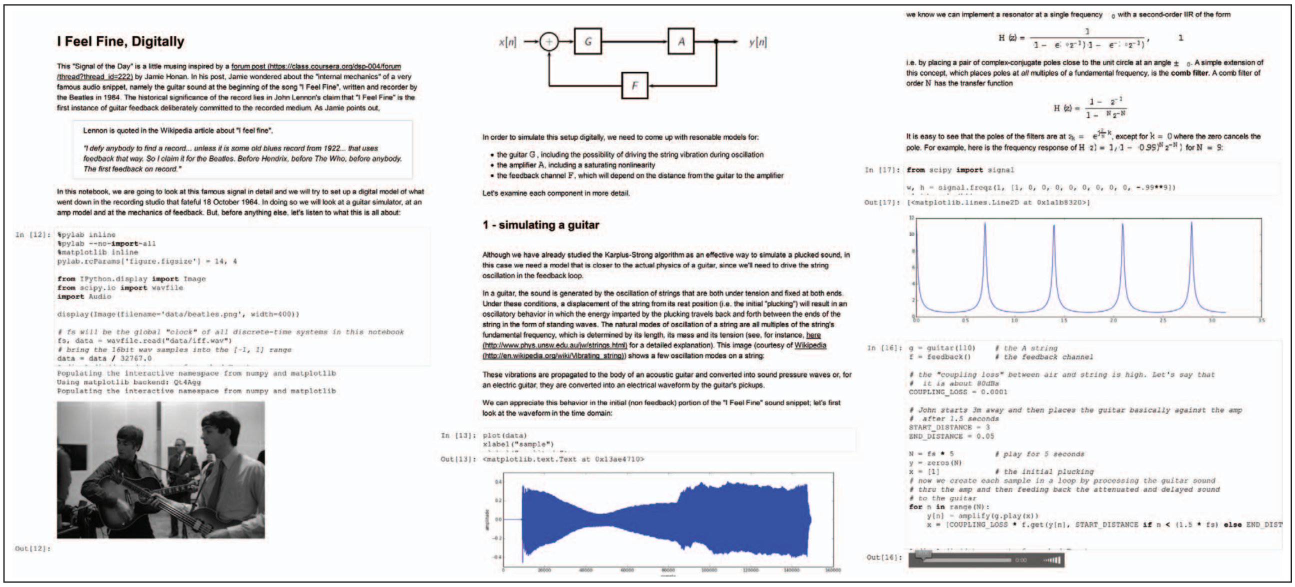
The first edition of the online class was offered in February 2013, at the height of MOOC hype. Although enrollment was definitely massive, so too was the dropout rate. The initial version of the class was produced under tight deadlines and required substantial effort and overtime. Successive editions (the fifth run ended in December 2015) have refined the original material in several respects, based on accumulated attendance data and feedback from students. We are currently in the process of reworking the material to reformat the class as a potentially self-paced course; a major part of the operation involves modifying the structure of the videos to fit in with the growing trend of bite-sized lectures.

Numerical examples

One of the most powerful features of DSP formalism is its independence from any specific programming language; this flexibility, however, also proved to be somewhat of a liability for online teaching. We knew that we wanted to provide working code with which students could play, but we also tried not to endorse one programming language specifically. To remain language agnostic, we realized that we could not assign programming homework because no realistic autograder could be put in place. Consequently, we initially decided to simply complement our lectures with a number of worked-out numerical examples, ranging from simple illustrations of signal processing algorithms to more ambitious mini-lectures with a clear focus on implementation. Originally, in the interest of expediency, we used MATLAB and encouraged students to translate the examples into their language of choice. This was aided by Mathworks' offer to provide a complimentary student license to all enrolled students. Starting with the fourth edition, we transitioned to a fully open-source solution and migrated all of our examples to Python by way of IPython notebooks [7], [8] (Figure 1). The notebooks allow us to write examples that can be either read as a worked-out exercise or downloaded, modified, and run locally by the students. This has been very well received, so much so that in the last edition we introduced a graded numerical homework in Python, where students have to code missing blocks in a fully functional MP3 encoder.

Personnel

The first edition of the class obviously required the greatest effort. We were fortunate to be able to rely on a great team of graduate students to develop exercises and troubleshoot both



I Feel Fine, Digitally

This "Signal of the Day" is a little musing inspired by a forum post (https://class.coursera.org/dsp-004/forum/thread?thread_id=222) by Jamie Honan. In his post, Jamie wondered about the "internal mechanics" of a very famous audio snippet, namely the guitar sound at the beginning of the song "I Feel Fine", written and recorder by the Beatles in 1964. The historical significance of the record lies in John Lennon's claim that "I Feel Fine" is the first instance of guitar feedback deliberately committed to the recorded medium. As Jamie points out,

> Lennon is quoted in the Wikipedia article about "I feel fine".
>
> *"I defy anybody to find a record... unless it is some old blues record from 1922... that uses feedback that way. So I claim it for the Beatles. Before Hendrix, before The Who, before anybody. The first feedback on record."*

In this notebook, we are going to look at this famous signal in detail and we will try to set up a digital model of what went down in the recording studio that fateful 18 October 1964. In doing so we will look at a guitar simulator, at an amp model and at the mechanics of feedback. But, before anything else, let's listen to what this is all about:

```
In [12]: %pylab inline
         %pylab --no-import-all
         %matplotlib inline
         pylab.rcParams['figure.figsize'] = 14, 4

         from IPython.display import Image
         from scipy.io import wavfile
         import Audio

         display(Image(filename='data/beatles.png', width=400))

         # fs will be the global "clock" of all discrete-time systems in this notebook
         fs, data = wavfile.read("data/iff.wav")
         # bring the 16bit wav samples into the [-1, 1] range
         data = data / 32767.0
         Populating the interactive namespace from numpy and matplotlib
         Using matplotlib backend: Qt4Agg
         Populating the interactive namespace from numpy and matplotlib
```

```
Out[12]:
```

In order to simulate this setup digitally, we need to come up with resonable models for:

- the guitar G, including the possibility of driving the string vibration during oscillation
- the amplifier A, including a saturating nonlinearity
- the feedback channel F, which will depend on the distance from the guitar to the amplifier

Let's examine each component in more detail.

1 - simulating a guitar

Although we have already studied the Karplus-Strong algorithm as an effective way to simulate a plucked sound, in this case we need a model that is closer to the actual physics of a guitar, since we'll need to drive the string oscillation in the feedback loop.

In a guitar, the sound is generated by the oscillation of strings that are both under tension and fixed at both ends. Under these conditions, a displacement of the string from its rest position (i.e. the initial "plucking") will result in an oscillatory behavior in which the energy imparted by the plucking travels back and forth between the ends of the string in the form of standing waves. The natural modes of oscillation of a string are all multiples of the string's fundamental frequency, which is determined by its length, its mass and its tension (see, for instance, here (http://www.phys.unsw.edu.au/jw/strings.html) for a detailed explanation). This image (courtesy of Wikipedia (http://en.wikipedia.org/wiki/Vibrating_string)) shows a few oscillation modes on a string:

These vibrations are propagated to the body of an acoustic guitar and converted into sound pressure waves or, for an electric guitar, they are converted into an electrical waveform by the guitar's pickups.

We can appreciate this behavior in the initial (non feedback) portion of the "I Feel Fine" sound snippet; let's first look at the waveform in the time domain:

```
In [13]: plot(data)
         xlabel("sample")
Out[13]: <matplotlib.text.Text at 0x13ae4710>
```

we know we can implement a resonator at a single frequency $_0$ with a second-order IIR of the form

$$H(z) = \frac{1}{1-\ e^{\cdot}\, {}_0 z^{-1})\,1-\ e^{-\cdot}\, {}_0 z^{-1})}, \qquad 1$$

i.e. by placing a pair of complex-conjugate poles close to the unit circle at an angle $\pm\ _0$. A simple extension of this concept, which places poles at *all* multiples of a fundamental frequency, is the **comb filter**. A comb filter of order N has the transfer function

$$H(z) = \frac{1-\ z^{-1}}{1-\ {}^{N} z^{-N}}$$

It is easy to see that the poles of the filters are at $z_k =\ e^{j\frac{2}{N}k}$, except for $k = 0$ where the zero cancels the pole. For example, here is the frequency response of $H\ z) = 1/\,1-\ 0.99)^N z^{-N})$ for $N = 9$:

```
In [17]: from scipy import signal

         w, h = signal.freqz(1, [1, 0, 0, 0, 0, 0, 0, 0, 0, -.99**9])
Out[17]: [<matplotlib.lines.Line2D at 0x1a1b8320>]
```

```
In [16]: g = guitar(110)     # the A string
         f = feedback()      # the feedback channel

         # the "coupling loss" between air and string is high. Let's say that
         #  it is about 80dBs
         COUPLING_LOSS = 0.0001

         # John starts 3m away and then places the guitar basically against the amp
         #  after 1.5 seconds
         START_DISTANCE = 3
         END_DISTANCE = 0.05

         N = fs * 5          # play for 5 seconds
         y = zeros(N)
         x = [1]             # the initial plucking
         # now we create each sample in a loop by processing the guitar sound
         # thru the amp and then feeding back the attenuated and delayed sound
         # to the guitar
         for n in range(N):
             y[n] = amplify(g.play(x))
             x = [COUPLING_LOSS * f.get(y[n], START_DISTANCE if n < (1.5 * fs) else END_DIST
Out[16]:
```

FIGURE 1. A numerical example in the IPython Notebook format.

the software platform and the material itself. We estimate that, cumulatively, in excess of 1,800 hours went into the deployment of the first offering. Subsequent editions required progressively less effort, and most of the new development work has been devoted to the correction of typos and the recording of additional material. Mostly, our time has been spent monitoring the student forum: on average, forum conversations have totaled between 100 and 200 messages per week and represent the true nexus of the class; in fact, we think that the instructors' presence on the forum is what keeps students "on board" and manages to project a sense of cohesion onto the learning effort. Additionally, as the class progresses, a natural ordering tends to take place within the student body, and not all posts require an answer from the staff because more advanced learners are all too happy to prove their mastery of the material.

Unique elements

Leitmotifs

For most prospective students, the syllabus, and the first five modules in particular, prove to be rather challenging because of their theoretical aspect. We quickly learned that to improve retention, we needed to persistently underscore the links between mathematical abstraction and the applied side of the discipline, and we tried to do so by introducing recurring themes at different points in the course. For example, the Karplus–Strong algorithm is initially used to introduce the concept of block diagrams, then in the section on discreet Fourier transform (DFT) versus discreet Fourier series (DFS), and again in the section on infinite impulse response (IIR) filtering.

Signal of the day series

To lighten the more abstract side of the course, we started producing a growing collection of extra modules in which we introduce famous signals or equally famous signal processing algorithms; see "Signal of the Day Series (EPFL)" for some examples.

The class by the numbers

The final attendance figures for the five editions of the class are listed in Table 2, where active students are students who watched at least one video lecture. The final grade in each edition was based only on homework completion: 45% of correct solutions granted a passing grade, and 90% granted a certificate with distinction. Whereas the declining enrollment figures show that the "novelty effect" of MOOCs is understandably wearing off, we notice a rather stable ratio between enrollment and active students (around 50%) and between active students and successful completers (around 4%). This seems to be in line with similar reports from the field [12].

Retention

Just like its residential counterpart, the online class requires a working knowledge of calculus and linear algebra, and the largest number of nonincidental dropouts are imputable to a lack of minimal prerequisites. The second letter in "MOOC" stands for "open," and it is this openness that makes it all but impossible to filter enrollments; we do provide a voluntary entrance self-test, but few seem to take it seriously. Effective preselection remains a problem if overall retention is to improve. Interestingly, the dropout rate for students that pay approximately US$40 for a verified certification is about 50%, so perhaps even a minimal fee would eliminate many nominal enrollments with no attendance.

After the initial drop in attendance, the attrition rate is fundamentally dominated by external circumstances. The exit questionnaire, which is also answered by noncompleters, confirms that online classes are understandably a lower priority for most participants and vulnerable to unforeseen personal and professional events.

Signal of the Day Series (EPFL)

As we were preparing the third edition of the École Polytechnique Fédérale de Lausanne Digital Signal Processing (DSP) massive open online course, we thought it would be cute to start each lecture with a signal of general interest, tell its history, and show some notable examples of relevant processing. We went the extra mile to try to find "original" signals, either for their historical importance or with respect to their relevance to concepts taught in the class. The challenge was then to transform a cute idea into an attractive and engaging three- to five-minute video (or an IPython Notebook) that we call the *Signal of the Day* series.

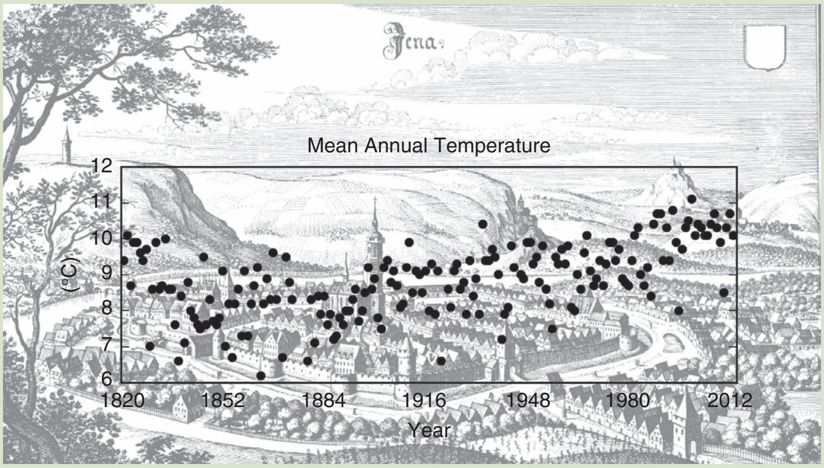


FIGURE S1. The Goethe time series.

There was no shortage of ideas, and colleagues from the lab suggested examples ranging from the mundane to the downright bizarre; after all, signals are everywhere. Our goal, in all of this, was to reach a broader community than "just" those involved in signal processing; we wanted a collection of signals from different scientific communities and those that involved diverse processing challenges. We are convinced, after all, that a lot of people in science do signal processing without realizing it; reaching out to these communities is both interesting and fun.

As an example, while talking to a German environmental scientist, we discovered that in 1821, Johann Wolfgang von Goethe, the famous German writer, had started taking daily temperature measurements in his hometown of Jena. This practice has been carried on to the present day by the Jena weather station (with only a few exceptions during World War II), and so Goethe's time series is probably one of the oldest "live" discrete-time records in existence [22]. We turned this into our first "signal of the day," applying a simple moving average filter to demonstrate that, despite claims by climate change skeptics, temperature is indeed rising (Figure S1).

Table S1. Signals of the day.

von Goethe's Temperature Measurements
The Tristan Chord
Lehman Brothers' stock value ca. 2008
Sputnik: The first man-made signal from outer space
I Feel Fine: The Beatles and guitar distortion
Can one hear the shape of a room? [10]
Moiré patterns
Camera resolution and space exploration
Exoplanet hunting
Safecast: Crowdsourced radioactivity measurements after Fukushima [9]

Currently, we have ten signals of the day, with a few more under construction (see Table S1). A fun one is certainly the one explaining why one should never wear a striped shirt when on television, a playful way to introduce aliasing and its effects. It was also interesting to discover that other communities have developed similar ideas (see, for instance, the "astronomy picture of the day" [23]); as signal processing moves more and more to the online medium, an ever-growing library of notable DSP examples, with contributions from the entire DSP community, would be a fantastic project.

Engagement

For a more dynamic perspective, the engagement data over time for the fourth edition of the class are shown in Figure 2. The curves are very similar in shape for all editions and allow us to draw the following quantitative analysis:

- Attendance measured as passive visits to the website displays a steep decay in the first weeks of class and levels off after week four. This roughly corresponds to the end of the Fourier analysis module, i.e., those who survive Fourier seem to keep their interest alive throughout the rest of the course.
- The percentage of students who submit homework seems to remain constant, i.e., from the start the number of students who attempt to obtain a certificate represents a small percentage of the total number of participants.

Table 2. Attendance figures on Coursera for the EPFL DSP class, with "Yield" representing passed/registered in percentage.

Edition	Date	Registered	Active	Passed	Yield
First	Feb. 2013	48,000	24,000	1,500	3.1
Second	Oct. 2013	35,000	20,000	1,000	2.9
Third	May 2014	19,000	10,000	280	1.5
Fourth	Jan. 2015	25,000	17,000	450	1.8
Fifth	Oct. 2015	16,000	7,500	360	2.2

- A peak of activity at the end of the class (April 2015) occurs, but only in terms of visits. Apparently, in online classes as in real classes, in the end students try to cram (and then give up).
- Interestingly enough, visits do not taper off to zero after the end of the class (April 2015 in Figure 2) but remain at the same level until the next offering of the course (homework submissions and forum participation obviously do stop). This strongly indicates that a MOOC model based on self-paced learning certainly has its place alongside monitored editions of the class.

Demographics

(Note: Although the following data refer to the fourth edition of the class, no significant differences have been remarked in the latest edition.) The student population showed a pronounced gender imbalance, with an 87% male component; a majority (42%) was in the 25–34 age range; and approximately half of the attendees were full-time employees and 37% full-time students. Overall, 31% held an M.S. degree and 34% a bachelor's degree, which suggests many take online classes of this kind more as a sort of refresher. The geographical distribution of students sees the United States at the top with 20%, followed by India (17%) and China (8%). As a whole, Asia leads with 38% of enrollments, followed by Europe (26%) and North America (25%).

Impact on the EPFL residential course

Initially, we thought about using MOOC recorded material to flip the classroom on campus. We attempted this during the first run of the online class, which coincided with a scheduled offering of the residential course. Students were instructed to watch the video lectures at home and prepare for Q&A sessions during the nominal lecture hours, but the experiment was unsuccessful. Students expressed an unmitigated dislike toward the absence of standard lecture time and considered their learning experience to be incomplete. We therefore decided not to repeat the format, and now we simply recommend that students enroll in the online classes and use the material to review and catch up on the standard lectures.

Feedback

Overall feedback from the online students was decidedly positive for every edition. The consensus held that the class was hard ("harder than I anticipated" was perhaps the most common commentary in the exit questionnaire), but the rigor was almost unanimously appreciated by the students who made it to the end of the class. In general, the more senior and more educated students tended to ask for more material and a longer, in-depth class. Younger participants advocated splitting the class into shorter independent units. We certainly cannot claim to have pleased everyone, and we had our share of constructive and nonconstructive criticism. But the real privilege, as teachers, is the wealth and diversity of direct feedback that a MOOC provides. Considering that approximately 70 students per year attend the residential class, we can now sift through a century's worth of class evaluations!

Rice University's Discrete-Time Signals and Systems on edX

Overview and goals

Discrete-Time Signals and Systems (301x) is a rigorous mathematical introduction to signal processing modeled on 50% of the Rice University course ELE301, Signal and Systems, a core undergraduate class taken by all electrical and computer engineering (ECE) majors, typically in the junior year. Rather than following the classical approach to teaching discrete-time signals and systems as discretized versions of continuous-time signals and circuits, the course approaches discrete-time signals and systems from first principles. The key overarching theme is the importance of linear algebraic concepts in signal processing: vector spaces, signals as vectors, linear systems as matrices/operators, linear time-invariant (LTI) systems as Toeplitz/circulant matrices, and the Fourier transform from the eigendecomposition of these LTI matrices. The course, which also covers the z transform and filter analysis and design, teaches students to analyze discrete-time signals and systems in both the time and frequency domains. Students continuously apply these concepts in interactive MATLAB programming exercises. The course has been taught twice on edX: the first edition in spring 2014 as a ten-week course [19] and the second edition in spring 2015 as two five-week mini-courses [14], [20].

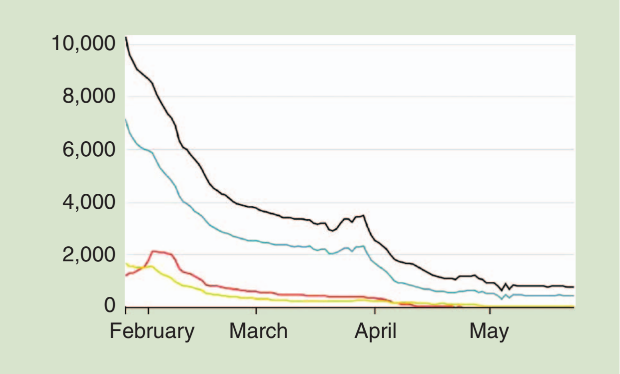


FIGURE 2. Engagement data for the fourth edition (19 January 2015–6 April 2015) of the EPFL course: visits to the course page (black), video views (blue), homework submission (red), and forum browsing (yellow).

Table 3. Topics by week in the first edition of Rice University's course, ELEC301x.

Weeks	Topics
0	Preclass activities (optional)
1	Introduction
2	Signals are vectors
3	Linear systems
4	Convolution
5	Discrete fourier transform
6	Discrete-time fourier transform
7	*z* transform
8	Analysis and design of filters
9	Exam

Course organization

Outline

The class incorporates numerous learning elements to engage students, stressing the balance between rigorous mathematical theory and hands-on practical applications. The course flow of the first edition is detailed in Table 3. The second edition was split into two mini-courses, one covering time-domain tools and one frequency-domain tools (with the split occurring at week 5 in Table 3).

Both course editions include an optional one-week precourse refresher on the key prerequisites in mathematics (complex arithmetic and linear algebra) and programming (MATLAB). A collection of reference material was made available in the Rice University-based open access education platform OpenStax CNX [21]. See Table 4 for an overview of the key course elements.

Style and format

Our perusal of the cognitive science literature indicated that a "talking head" video lecture did not lead to improved learning outcomes in an online course, and so we produced lecture videos consisting of the voice of the instructor as he manipulated the slides and MATLAB windows on a tablet. See Figure 3 for sample screenshots from the course. To personify the course, the instructor appeared in a lighthearted video introducing each week's concepts. To broaden student experience and supplement the course, we produced a range of *Office Hours* videos conducted by Rice University graduate student Raajen Patel. Video production support was provided by Rice Online, a major MOOC initiative of Rice University.

Homework and grading

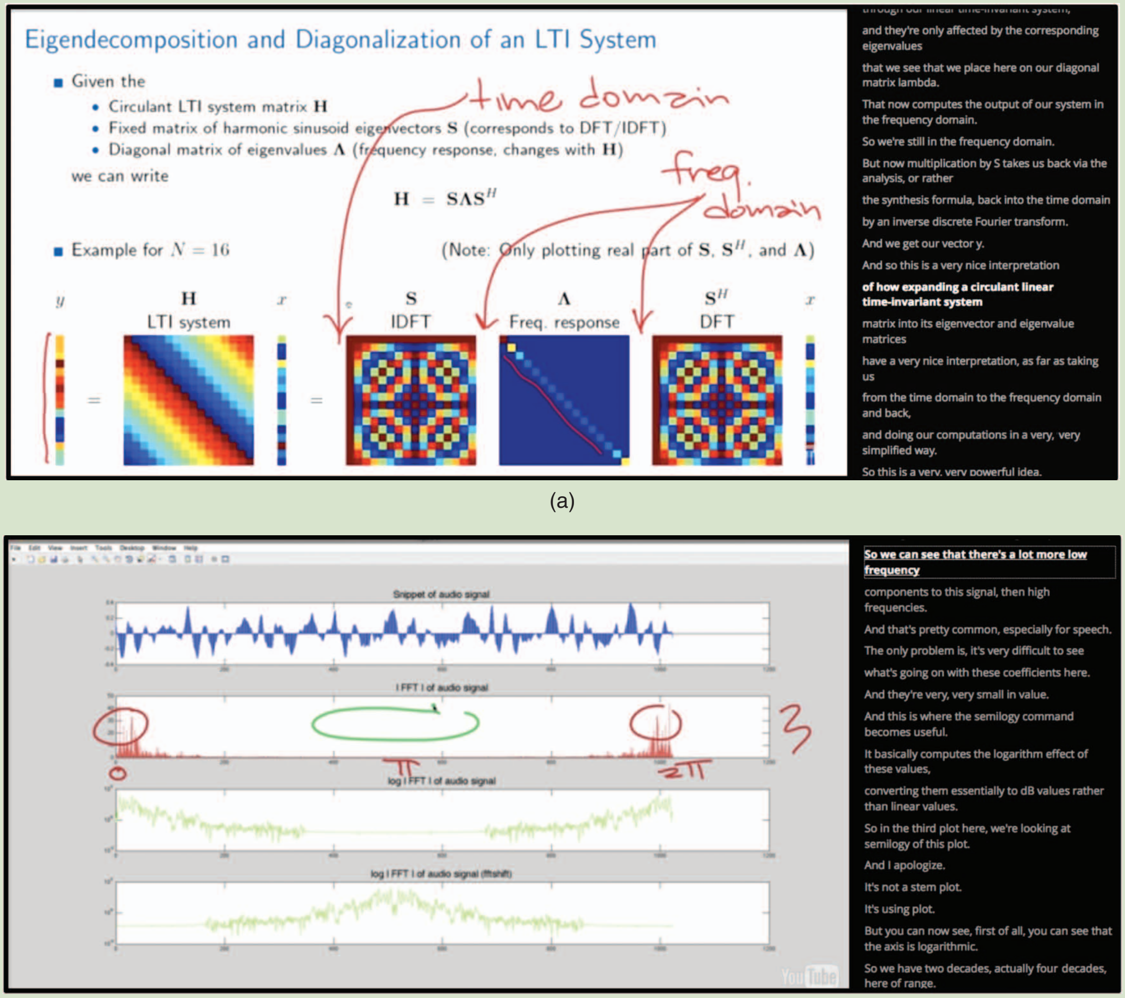
On the theory and analysis side, given the current limitations of the edX platform, we assess students primarily using multiple-choice questions. In the second edition of the course, each weekly homework contains one open-form response question whose response is input via MathJax and peer graded by three other students. A model solution and grading rubric are made available after each homework is due. And as discussed next, each homework also includes numerical problems in MATLAB that are assessed algorithmically via the edX platform. The final student grade combines performance on the weekly homework, case studies (recall Table 4), and the final exam. A score of 60% is required to pass the course.

Course evolution

After research by the Rice Online team suggested that MOOCs are more successful when they are shorter rather than

Table 4. Course elements in Rice ELEC301x in its first edition. The second edition split the course into two minicourses (breaking at week 5) and integrated the case studies into the weekly homework.

Precourse math refresher	Students can self-review the required mathematical skills with practice exercises and tutorials before the start of the formal course.
Preclass MATLAB tutorial	Introductory tutorials on the MATLAB programming language enable students to get up to speed.
Introduction videos	Each week kicks off with a light-hearted overview of the week's material featuring Richard Baraniuk, Mr. Lan, and BIBO, the bear.
Lecture videos	Each week features several hours of lecture videos recorded by Baraniuk specifically for an online format. Lectures were chunked into 5- to 20-minute segments.
Quick questions	Conceptual knowledge-check questions follow each segment of lecture video to test students' understanding and maintain engagement.
Supplemental resources	Links to additional learning content, exciting related applications, and additional information add depth to the learning experience.
Office Hours videos	Videos of a TA working out homework-type problems prepare students for the assignments and encourage appropriate problem-solving techniques.
Homework	Each week, a rigorous problem set challenges students to apply what they have learned, graded via multiple choice and peer review. For numerical problems, students program in the MATLAB language using an integrated development environment built into the edX platform.
Discussion forum	An active discussion forum enables students to ask and answer questions and receive feedback from course staff.
MATLAB case studies	Biweekly programming case studies enable students to apply concepts learned to practical programming exercises designed to show how signal processing is used.
Final exam	A traditional final exam tests students' comprehensive course knowledge.
Final case study	Serving as a final project, the final case study expands on the previous case studies as students gain programming proficiency.



Eigendecomposition and Diagonalization of an LTI System

- Given the
  - Circulant LTI system matrix $\mathbf{H}$
  - Fixed matrix of harmonic sinusoid eigenvectors $\mathbf{S}$ (corresponds to DFT/IDFT)
  - Diagonal matrix of eigenvalues $\mathbf{\Lambda}$ (frequency response, changes with $\mathbf{H}$)

we can write

$$\mathbf{H} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^H$$

- Example for $N = 16$

(Note: Only plotting real part of $\mathbf{S}$, $\mathbf{S}^H$, and $\mathbf{\Lambda}$)

through our linear time-invariant system,

and they're only affected by the corresponding eigenvalues

that we see that we place here on our diagonal matrix lambda.

That now computes the output of our system in the frequency domain.

So we're still in the frequency domain.

But now multiplication by S takes us back via the analysis, or rather

the synthesis formula, back into the time domain

by an inverse discrete Fourier transform.

And we get our vector y.

And so this is a very nice interpretation

of how expanding a circulant linear time-invariant system

matrix into its eigenvector and eigenvalue matrices

have a very nice interpretation, as far as taking us

from the time domain to the frequency domain and back,

and doing our computations in a very, very simplified way.

So this is a very, very powerful idea.

(a)

So we can see that there's a lot more low frequency

components to this signal, then high frequencies.

And that's pretty common, especially for speech.

The only problem is, it's very difficult to see

what's going on with these coefficients here.

And they're very, very small in value.

And this is where the semilogy command becomes useful.

It basically computes the logarithm effect of these values,

converting them essentially to dB values rather than linear values.

So in the third plot here, we're looking at semilogy of this plot.

And I apologize.

It's not a stem plot.

It's using plot.

But you can now see, first of all, you can see that the axis is logarithmic.

So we have two decades, actually four decades, here of range.

(b)

FIGURE 3. Sample screenshots from Rice University ELEC301x lecture videos showing (a) a content slide and (b) a MATLAB demonstration.

longer, the Rice administration recommended that we split the first-edition MOOC into two mini-MOOCs, each lasting five weeks. The split was accomplished by breaking the original course into one mini-course covering time-domain concepts and one mini-course covering frequency-domain concepts.

Numerical examples

As discussed previously, we provide a variety of opportunities for students to explore the numerical/application side of signal processing through 1) MATLAB examples interspersed in the lecture videos, 2) *Office Hour* videos, 3) case studies, and 4) automatically graded MATLAB problems in homework.

Personnel

Realizing the MOOC took a village, instructor Richard Baraniuk was assisted by Rice DSP research engineer Heather Seeba, DSP consultant Matthew Moravec, graduate students Raajen Patel (*Office Hours*) and Eva Dyer (case studies), undergraduate student Tan Nguyen (MATLAB homework), and six undergraduates (forum moderation).

Unique elements

Week 0

Because students come to the class with wildly varying backgrounds, both editions of the course include an optional one-week precourse refresher on the key prerequisites in mathematics (primarily complex arithmetic and linear algebra) and programming (MATLAB). Students can test their preparedness with a pretest (for zero credit).

Integrated MATLAB

ELE301x was one of the first MOOCs to exploit a collaboration with MathWorks that made MATLAB freely available to students. Through the edX platform, students can access, within their web browser, a server running MATLAB and even submit their code for autograding.

Community Teaching Assistants (Rice University)

Conventional residential courses are typically staffed by one or more teaching assistants, who grade and sometimes create homework and exams, hold office hours, and facilitate laboratories. The concept is typically also ported over to online courses such as massive open online courses (MOOCs). The Rice ELE301x discussion forum quickly became vibrant in both editions of the course. Moreover, certain students began taking a leadership role when answering forum questions and offering advice. One of these students, John Coppens (see https://www.jcoppens.com), a practicing engineer with a wealth of real-world experience, was so active in the first edition of the class that we elevated him onto the course team as a community teaching assistant for the second edition.

This is an intriguing exemplar of how the "openness" of an MOOC can lead to emergent behaviors that are unseen in residential courses. Moreover, this is a preliminary indication that the dream of MOOC scalability could in fact be realizable. With so many of the ELEC301x students having advanced degrees (see the section "The Class by the Numbers"), we plan to continue to encourage and reward such positive contributions in the future.

John Coppen's reflections on ELEC301x provide a number of insights into why he took such initiative: "I was very interested in the subject, and, over the years, have been studying and implementing small projects for myself. The course coincided with the first semester of the year, when I normally have a little more time to spare, and the general feel of the course through its videos and discussion forum was 'open,' 'stimulating,' and 'inviting.' Moreover, reactions to problems posted to the forum were prompt and helpful. Minor issues with the tasks and exercises actually forced me to do more investigation than was called for in the course, which was a great learning experience. Perhaps MOOC instructors should leave some such issues on purpose."

Case studies

In addition to the usual analysis and calculation assessment, Rice graduate student Eva Dyer developed a suite of case studies that challenge students on the core concepts of the class through a series of real-world application programming exercises. These studies enable students to explore first hand how signal processing is used in a wide range of real applications. Case study topics include audio synthesis using sinusoids, predicting financial time series using moving average filters, audio synthesis of a clarinet using attack-decay-sustain-release curves, filter design for Karplus–Strong string synthesis, and neural spike sorting. In the second edition of the course, case studies were integrated into the weekly homework.

Office Hours videos

Rice graduate student Raajen Patel prepared videos of worked examples that give students more insight into the material and encourage strong problem-solving skills.

Post-MOOC community

Because interest in both editions of the course continued even after they were closed and archived, we worked to engage students in an ongoing, post-MOOC community. Many students were interested in staying in touch with both course staff and other students. One particularly useful contribution from 200 members of the community was to work out step-by-step solutions to a number of signal processing problems. These solutions were then used as test data for a Rice project on mathematical language processing [S2] that aims to automatically grade mathematical calculations and provide appropriate feedback to students. For two additional unique elements, see "Community Teaching Assistants (Rice University)" and "MOOCs as an Experimental Platform (Rice University)," respectively.

The class by the numbers

We ran an interesting experiment purely by accident. For the second edition, very close to when standard practice would dictate that we announce the course (several months in advance), we were strongly recommended to split the ten-week course into two five-week courses. Accomplishing the split took several months, and so the course was announced late (approximately one month in advance). This reduced the number of registrants significantly (see Table 5). However, the proportion of students that actually completed at least the first part of the course was more than twice that in the first edition. This is evidence for the hypothesis that registration numbers are not very informative for MOOCs, because many potential students will register after reading the announcement of a course without ever truly intending to complete the work required to finish it.

Retention and engagement

The yield of enrolled to passed students was above the average for edX courses at the times our two editions were offered; see Table 5.

Demographics

The median age of residential Rice students taking ELEC301 is 20, but the median age for students taking the two editions of the course was 27. Interestingly, 30% of registrants had a high school diploma or less, 39% had a

MOOCs as an Experimental Platform (Rice University)

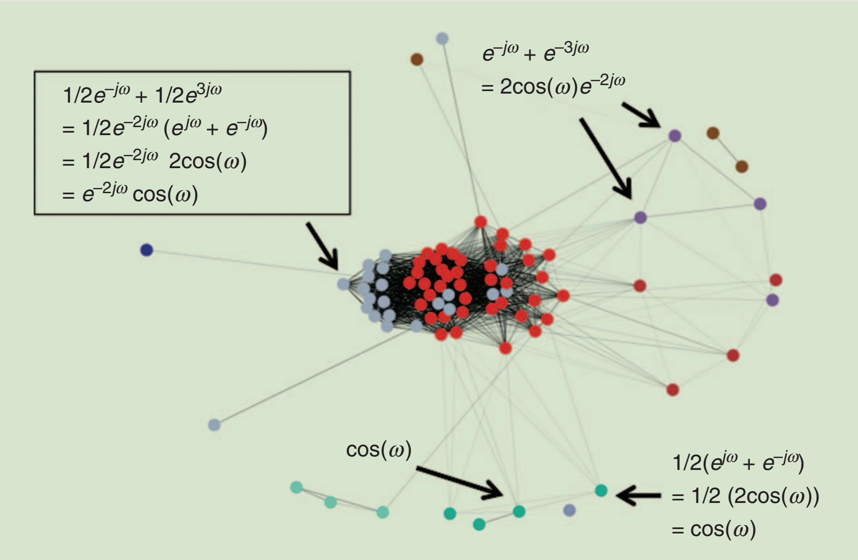


FIGURE S2. An illustration of the clusters obtained by MLP to 100 students' solutions to a signal processing problem. Each node corresponds to a solution. Nodes with the same color correspond to solutions that are estimated to be in the same cluster. The thickness of the edge between two solutions is proportional to their similarity score. The boxed solution is correct; all others are in varying degrees of (in)correctness.

To help support the burgeoning learning analytics and cognitive science research program at Rice (see http://openstaxtutor.org), we used ELEC301x to conduct a range of experiments.

Learning analytics experiment

A study in the final week of the second edition of ELEC301x assessed whether students believe in or agree with learning analytics data that were presented to them in a dashboard. (Learning analytics involves measuring, collecting, and analyzing data about learners to understand and optimize learning.) More specifically, we aimed to determine whether students would use the analytics to guide future learning activities, the analytics were "better" than the students' own metacognitive judgments, and our sparse factor analysis for learning and content analytics [S1] was accurate for all students. A research paper on our findings is in preparation.

Mathematics language processing experiment

The weekly open-form response question in the second edition of ELEC301x collected valuable test data for a Rice project on mathematical language processing (MLP) [S2] that aims to autograde mathematical computations and provide appropriate feedback to students. These data were augmented with additional step-by-step solutions from the 200 members of the post-MOOC community.

MLP leverages solution data from a large number of learners to evaluate the correctness of their solutions, assign partial-credit scores, and provide feedback to each learner on the likely locations of any errors. MLP takes inspiration from the success of natural language processing for text data and comprises three main steps.

1) Convert each solution to an open response mathematical question into a series of numerical features.
2) Cluster the features from several solutions to uncover the structures of correct, partially correct, and incorrect solutions.
3) Autograde the remaining (potentially large number of) solutions based on their assigned cluster and one instructor-provided grade per cluster.

As a bonus, we can track the cluster assignment of each step of a multistep solution and determine when it departs from a cluster of correct solutions, which enables us to indicate to learners the likely locations of errors. Figure S2 illustrates the clusters of (correct and incorrect) solutions to the following signal processing problem.

Question: *A discrete-time linear time-invariant system has the impulse response shown in the figure (omitted). Calculate* $H(e^{iw})$*, the discrete-time Fourier transform of the impulse response* $h[n]$*. Simplify your answer as much as possible until it has no summations.*

References

[S1] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics," *J. Mach. Learning Res.*, vol. 15, June 2014.

[S2] A. S. Lan, D. Vats, A. E. Waters, and R. G. Baraniuk. (2015, Jan. 18). Mathematical language processing: Automatic grading and feedback for open response mathematical questions. *ACM Learning at Scale.* [Online]. Available: http://arxiv.org/abs/1501.04346

college degree, and 28% had an advanced degree. As with the EPFL and MIT courses, the student population was predominantly male (83%).

The geographic distribution of the first edition of the MOOC was India (25%), United States (20%), United Kingdom (3%), Germany (3%), and China (3%), followed by 168

Table 5. Enrollments in Rice ELEC301x on edX in spring 2014.

Edition	Enrollment	Active	Passed	Yield (%)
First, spring 2014	22,819	1,145	583	2.6
Second, spring 2015, part 1	5,522		302	5.5
Second, spring 2015, part 2	4,376		161	3.7

In edX, an active student is one who submitted an assessment during the second week of the course and received a nonzero score.

other countries. In the second edition, the United States and India switched top places.

Impact on the Rice residential course

Rice ELEC301 is taught to approximately 40–60 students once per year. Our desire to make signal processing as accessible as possible in the MOOC led to a redesign of how discrete-time concepts are taught; in particular, we increased focus around linear algebra. This new approach was well received by the Rice students in fall 2015. Moreover, on-campus students took advantage of the recorded lectures as a supplemental resource and were disappointed that the lectures were unavailable for the continuous-time portion of the class.

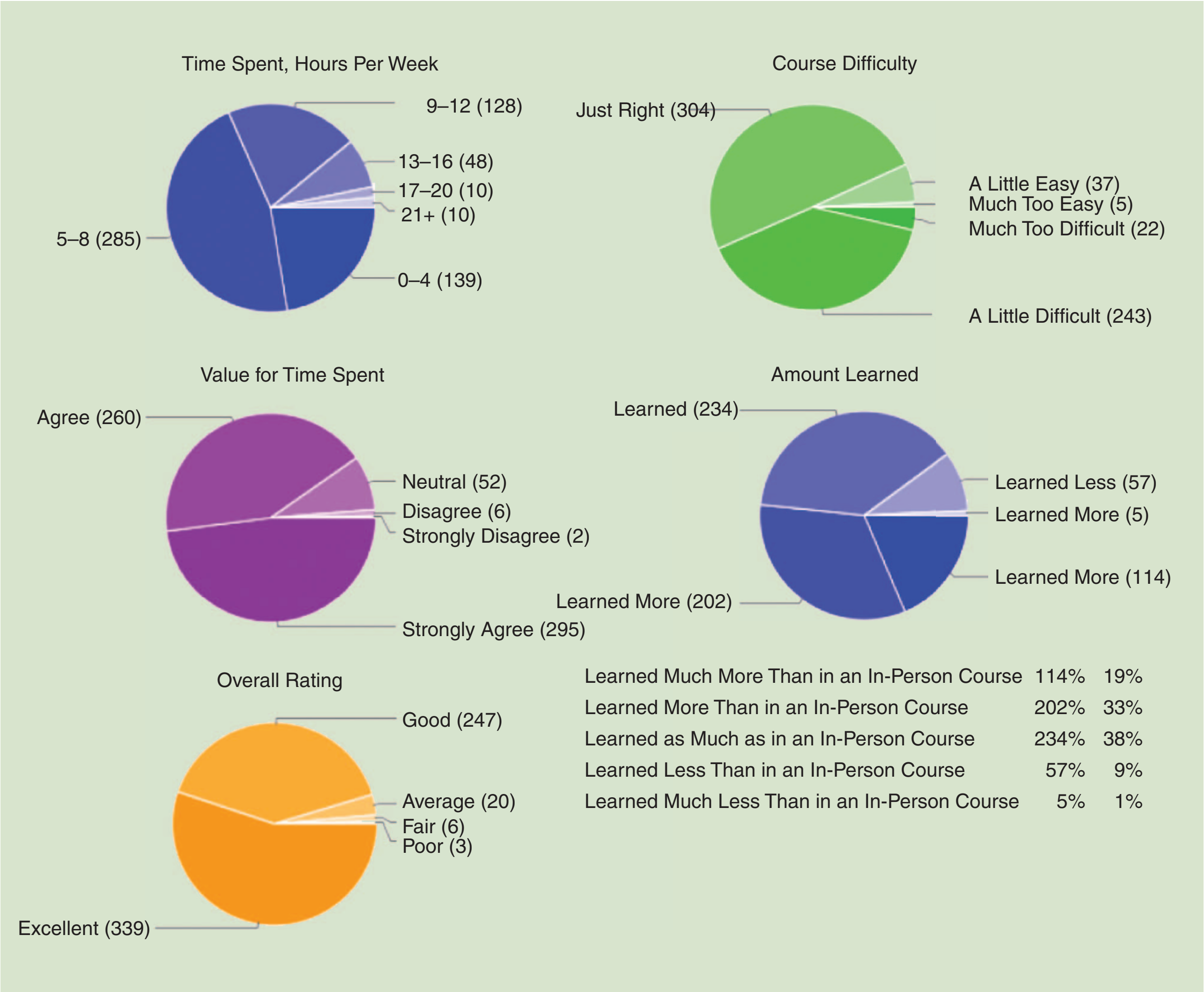
Feedback

We conducted a survey at the conclusion of both editions of the course using Google Forms and Qualtrics. A sampling of the results from 628 responses from the first edition is given in Figure 4. Students appreciated the vector-space approach to signals and systems, saying that it made the key concepts more accessible. Students also appreciated the optional week 0 refresher on the required mathematics and MATLAB.

MIT's Discrete-Time Signal Processing on edX

Overview and goals

The MIT Discrete-Time Signal Processing MOOC 6.341x on edX.org [15], coauthored by Alan Oppenheim and Tom Baran, is an outgrowth of and very strongly parallels the



Learned Much More Than in an In-Person Course	114%	19%
Learned More Than in an In-Person Course	202%	33%
Learned as Much as in an In-Person Course	234%	38%
Learned Less Than in an In-Person Course	57%	9%
Learned Much Less Than in an In-Person Course	5%	1%

FIGURE 4. A postcourse survey of Rice University 628 ELEC301x students after the first edition.

MIT residential introductory graduate subject 6.341, the textbook for which is Discrete-Time Signal Processing [17]. The residential course carries as a prerequisite the undergraduate subject 6.011, the textbook for which is Signals, Systems, and Inference [18], which itself has as a prerequisite of the basic background of Signals and Systems.

The residential course 6.341 has a lineage and evolution at MIT that go back to the 1970s, and a key goal in creating 6.341x has been to extend that lineage and evolution forward into the online realm. It is commonly expressed that the MOOC and online teaching and learning landscape is currently "the wild west." The territory is rapidly evolving and currently in a highly experimental stage. With this in mind, our overall goals with 6.341x included experimenting with online teaching and the edX platform with primary emphasis on 1) having a positive impact on the residential course at MIT and potentially residential courses elsewhere, 2) exercising and pushing the boundaries of the online edX infrastructure and platform, and 3) making the content of and experience with the MIT graduate subject more widely accessible worldwide at the level at which it is presented residentially at MIT. To accomplish these goals, the development of 6.341x evolved in four phases on which we elaborate shortly, after first describing the basic organization of the course.

Course organization

Outline

The course 6.341x is an 11-week graduate-level class divided into 18 units. Each unit consists of several topic segments, outlined in Table 6. In a typical week of the spring 2015 MOOC, students were provided with the following:

- *Multiple courseware topics*. Each topic consisted of a combination of brief exercises, text comments, and video segments recorded in a "chalk and talk" format, extracted and edited from lectures of the residential course at MIT, and with slides that were digitally animated specifically for the MOOC. The exercises were interspersed among the video segments specifically to allow students to verify their progress before moving on to the next segment. Consistent with the MIT residential course, lectures contained both mathematically oriented discussions and live demonstrations of signal processing concepts.
- *An overview video from the staff each week*. Videos were recorded in "talking head" style, providing an outline of the week's topics, in addition to brief, high-level audio signal processing demonstrations illustrating associated concepts.
- *A set of homework problems*. Problems included project-style numerical signal processing problems that students completed using in-browser tools developed specifically for 6.341x.

In addition, three exams were given to evaluate performance. Accompanying the courseware was an online discussion forum on which we comment in more detail below.

Style and format

Elements of the courseware are depicted in Figure 5. Figure 5(b) shows still frames captured from the lecture video segments, featuring chalk and talk-style video clips interspersed with lecture slides animated specifically for 6.341x, as well as in-class signal processing demos. The figure also depicts an example exercise that would fall between video segments, designed for students to verify their understanding. Weekly introductory videos were recorded in a talking head format, shown in Figure 5(a). Figure 5(c) and (d) illustrate interactive elements of two homework problems, in which students were prompted to choose spectral analysis parameters and enter block diagrams.

The basic staff–student interaction model used in the spring 2015 MOOC is depicted in Figure 6. Referring to this figure, the lowest-latency method for staff–student interaction was through the online discussion forum. From the perspective of students, homework problems and exercises provided instant feedback about performance, although aggregate results about student performance were viewable by the staff on a delayed basis. There was also typically a one-week delay between filming and deploying week overview videos, due to the time associated with editing and audio transcription.

Table 6. MIT 6.341x course outline.

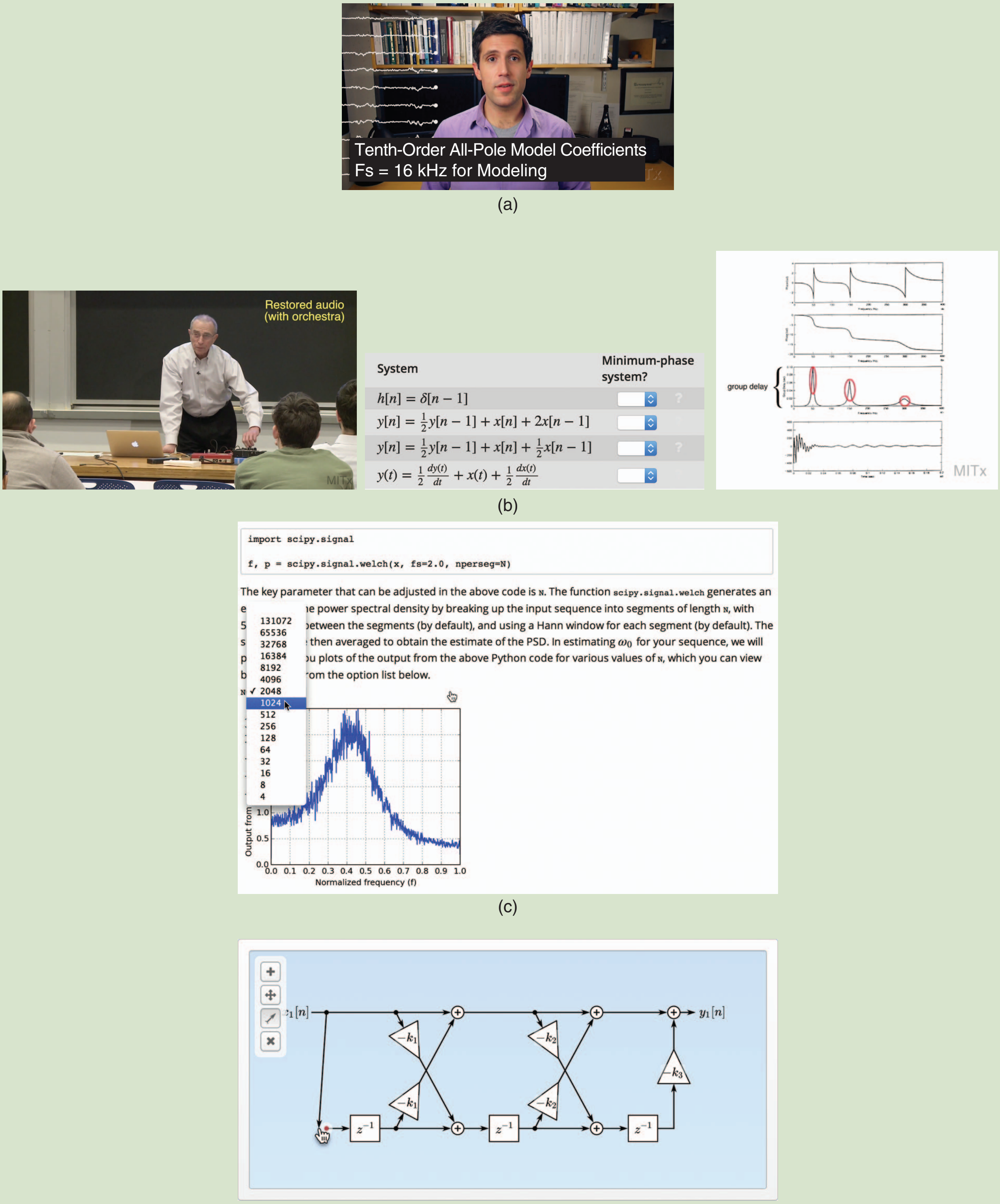
Course Unit	Release Date
Unit 1: Signals and systems in the time and frequency domains	Week 1
Unit 2: Allpass and minimum-phase systems	Week 2
Unit 3: Discrete-time processing of continuous-time signals	Week 2
Unit 4: Sampling rate conversion	Week 3
Unit 5: Quantization and oversampling	Week 3
Unit 6: Signal-flow graph implementations of LCCDEs[a]	Week 4
Unit 7: Lattice structures	Week 4
Unit 8: IIR filter design	Week 5
Unit 9: FIR filter design	Week 5
Unit 10: Parametric signal modeling	Week 6
Unit 11: The Levinson recursion	Week 6
Unit 12: Multirate systems and polyphase structures	Week 7
Unit 13: The DFT	Week 8
Unit 14: Computation of the DFT	Week 9
Unit 15: Spectral analysis	Week 10
Unit 16: The TDDTFT[b] and modulated filter banks	Week 11
Unit 17: Multirate and critically sampled filter banks	Week 11
Unit E: Enrichment lectures	Weeks 7 and 10

[a]LCCDEs: Linear constant-coefficient difference equations; [b]TDDTFT: time-dependent discrete-time Fourier transform.

Course evolution

The development of 6.341x consisted of four phases that began in the spring of 2013 and continued through the fall of 2015.

- *Phase I.* Work began in spring 2013 on the translation of significant content of 6.341, as used residentially, into digital form to be used initially as an online augmentation to the residential course. The online platform for digital content was initially Open edX, the open-source platform used residentially by MIT and other schools that mirrors the edX.org infrastructure. It was recognized at the outset that because the residential course covers a graduate-level subject, it would inherently



(a)

System	Minimum-phase system?
$h[n] = \delta[n-1]$	☐
$y[n] = \frac{1}{2}y[n-1] + x[n] + 2x[n-1]$	☐
$y[n] = \frac{1}{2}y[n-1] + x[n] + \frac{1}{2}x[n-1]$	☐
$y(t) = \frac{1}{2}\frac{dy(t)}{dt} + x(t) + \frac{1}{2}\frac{dx(t)}{dt}$	☐

(b)

```
import scipy.signal

f, p = scipy.signal.welch(x, fs=2.0, nperseg=N)
```

The key parameter that can be adjusted in the above code is `N`. The function `scipy.signal.welch` generates an
e ne power spectral density by breaking up the input sequence into segments of length `N`, with
5 between the segments (by default), and using a Hann window for each segment (by default). The
s then averaged to obtain the estimate of the PSD. In estimating ω_0 for your sequence, we will
p u plots of the output from the above Python code for various values of `N`, which you can view
b rom the option list below.

(c)

(d)

FIGURE 5. Screen captures of various elements in the MIT 6.341x courseware. (a) Week overview video with brief audio-based signal processing demonstration. (b) Course topic sequence, composed of in-class lecture videos, interactive problems, and animated slides. (c) The interactive homework problem related to spectral analysis. (d) The interactive portion of homework problem for which students are asked to graphically apply the transposition theorem.

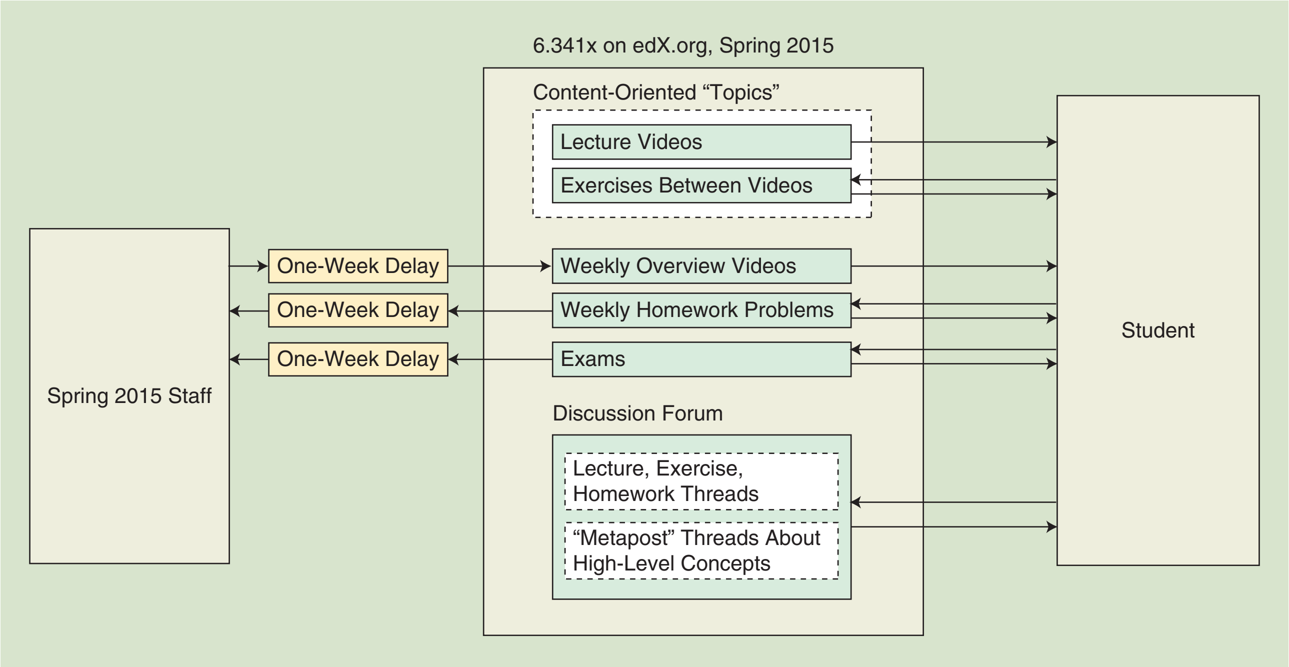


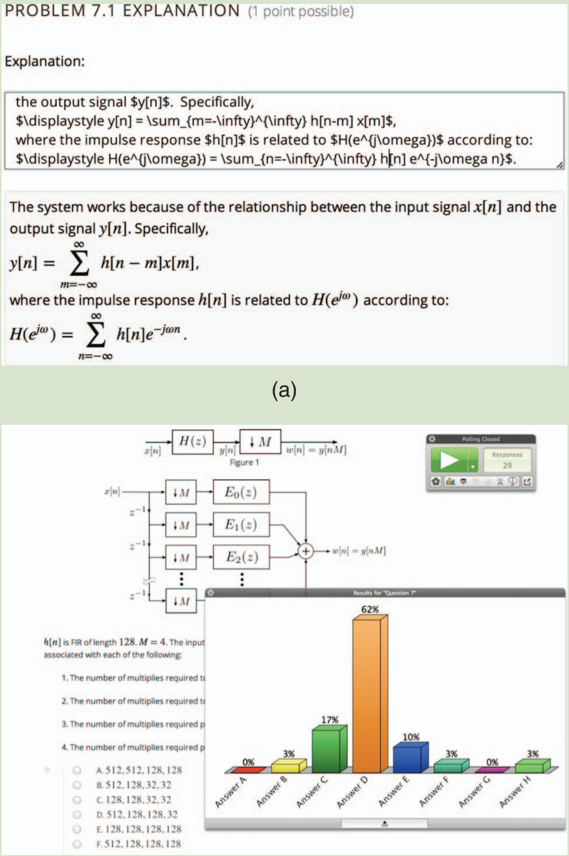
FIGURE 6. A diagram indicating the modes of interaction and flow of information between MIT staff and students, facilitated by the learning platform as it existed in spring 2015.

be more demanding than more introductory online courses. In developing the 6.341x content for this platform, we chose to work "close to the bare metal" of the edX platform to push the boundaries and help to improve the platform.

The process included adapting existing problem sets and a background exam for the online platform, autograding, and creating "finger exercises" to be given to the class and responded to anonymously with fixed-function wireless "clickers" during lecture. A histogram of responses to the exercises, displayed immediately in class, impacted subsequent discussion and pace during the lecture [see Figure 7(b)]. The decision to use fixed-function clickers, as opposed to, e.g., a phone or tablet app, was motivated by the desire to keep the attention of students out of their e-mail and browsers and focused on the lecturer and class discussion. The residential offering of the course in fall 2013 incorporated all of this online content to augment the live lectures. In addition, the live lectures were video recorded.

- *Phase II.* In spring 2014, the video recordings of fall 2013 lectures were heavily edited for crisper pacing, generating video segments of appropriate content and length, and sequencing with the finger exercises. In the residential offering, fall 2013 students were given a background exam on Open edX and completed autograded problem sets on the platform. The problem sets also used an in-browser "explanation box" system that was developed for the residential course, in which students could enter symbolic equations, proofs, and reasoning about their answers [see Figure 7(a)]. These were then electronically distributed to the course staff for manual grading. Given the emphasis on a deep conceptual understanding that has traditionally been a key part of the residential course, the comment box system provided a way to assess this understanding without diluting the problems to fit within the constraints of an autograding system.

  The success of the online experience in the residential course was a key motivation for carrying the material forward to a MOOC. Toward this end, the digital content was continually expanded and refined until it was ready to be run as a private beta for a limited number of participants from industry in fall 2014, and then publicly deployed on edx.org in spring 2015. This transition was a significant effort, funded largely by the MIT electrical engineering and computer science (EECS) department and the MIT Office of Digital Learning (ODL), together with edX. A significant effort was required to edit the in-class video recordings for the MOOC environment, which we found works best with short, ten-to 20-minute, well-paced segments. The in-class exercises were also augmented and in some cases modified for a better match to the edX environment.
- *Phase III.* This phase consisted of first offering 6.341x as a MOOC in fall 2014 in a beta version limited to 200 participants from industry, and then in spring 2015 as a fully open online course. For all registered students, the autograded online background exam was made available before the start of the MOOC to allow participants to assess their background relative to the course content.
- *Phase IV.* In the fall semester of 2015, residential 6.341 was offered at MIT with the usual structure of three hours of live class time per week, one hour of live recitation discussion with the teaching assistants (TAs), and a handwritten midterm and final exam graded by the staff. The course made full use of all of the digital online 6.341x content


PROBLEM 7.1 EXPLANATION (1 point possible)

Explanation:

```
the output signal $y[n]$. Specifically,
$\displaystyle y[n] = \sum_{m=-\infty}^{\infty} h[n-m] x[m]$,
where the impulse response $h[n]$ is related to $H(e^{j\omega})$ according to:
$\displaystyle H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n}$.
```

The system works because of the relationship between the input signal $x[n]$ and the output signal $y[n]$. Specifically,

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]x[m],$$

where the impulse response $h[n]$ is related to $H(e^{j\omega})$ according to:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}.$$

(a)

$h[n]$ is FIR of length 128. $M = 4$. The input
associated with each of the following:

1. The number of multiplies required t
2. The number of multiplies required t
3. The number of multiplies required p
4. The number of multiplies required p

○ A. 512, 512, 128, 128
○ B. 512, 128, 32, 32
○ C. 128, 128, 32, 32
○ D. 512, 128, 128, 32
○ E. 128, 128, 128, 128
○ F. 512, 128, 128, 128

(b)

FIGURE 7. Additional online tools used specifically for residential deployment of MIT 6.341x. (a) The MathJax-enabled student "explanation box," in which residential students provided staff with reasoning and derivations behind their autograded answers. (b) A polling system used with in-class "finger exercises," designed to provide self-assessment of student understanding during lectures.

running on the MITx platform. Participants were asked and strongly encouraged to watch the videos and work the finger exercises before coming to the live class sessions. Class sessions were structured to give an overview and incorporate many subtleties about the topic under the assumption that students had previewed the content online. During the class sessions, a number of finger exercises on the topic were presented as clicker questions with instant feedback to the staff and class through display of the histogram of responses. The syllabus schedule was also structured under the assumption that the students would preview the topics online before class sessions. Consequently, online content for week N was made available online to the class late Thursday evening in week $N - 1$. All homework on the topics in week N was due at the end of that week and was autograded online, but each problem also included an explanation box in which the student could elaborate on the solution. The staff reviewed these explanation boxes and student feedback was given quickly. Staff solutions were posted online immediately after the due date.

The overall experience for both students and staff in incorporating all of the digital online content and emphasizing the importance of previewing in both the structure of the schedule and in the level and pacing of the course was extremely positive.

Numerical examples and tools

In the residential course, multiple numerically focused class projects are traditionally assigned to students to provide practice with the use of IIR and finite impulse response (FIR) filter design and order estimation tools and with the use of various methods for spectral estimation. These projects have been very well received by students taking the residential courses at MIT. For the MOOC, we attempted to replicate or at least approximate the experience by creating a set of in-browser tools that learners were able to use to perform many of the numerical tasks associated with these projects.

As one example, we created a set of in-browser tools, discussed in greater detail in "Numerical Tools for Filter Design (MIT)," for performing the numerical tasks commonly associated with FIR and IIR filter design. The tools allowed learners to estimate design parameters, compute filter coefficients, and compare the performance of the resulting filters against stated specifications. The numerical output from the tools was then evaluated in the context of various assigned problems, which provided textual as well as graphical feedback about the submitted filter designs. For example, any regions of the magnitude response violating the stated constraints would automatically be highlighted, indicating to the learner where specifications had been violated. Automated textual messages about the submitted designs were also programmed to respond to common pitfalls that we identified as having commonly occurred in class projects during past semesters of the residential course.

Personnel

Through its four phases, the development of 6.341x required considerable resources and support from the MIT EECS department and ODL and the edX team. The responsibility for content development and for incorporation of the content into the online platform was ours along with Tarek Lahlou, an EECS graduate student, who also was an instructor for the industrial beta version of the MOOC. In developing interactive content and incorporating it into the online platform, we collaborated closely with the edX and ODL teams and, in particular, with TC Haldi, Tsinu Hermano, Joe Martis, and Peter Pinch.

A major effort in developing content was required for editing and reformatting the in-class live video recordings into segments with good pacing and length. This editing was the responsibility of Alan Oppenheim, Tom Baran, and Isaac Chuang, together with video editors Jim Ohm and Edwin Cabrera. The video segments were also further reviewed for accuracy and appropriate highlighting by Tarek Lahlou, as well as EECS graduate students Anuran Makur and Lucas

Numerical Tools for Filter Design (MIT)

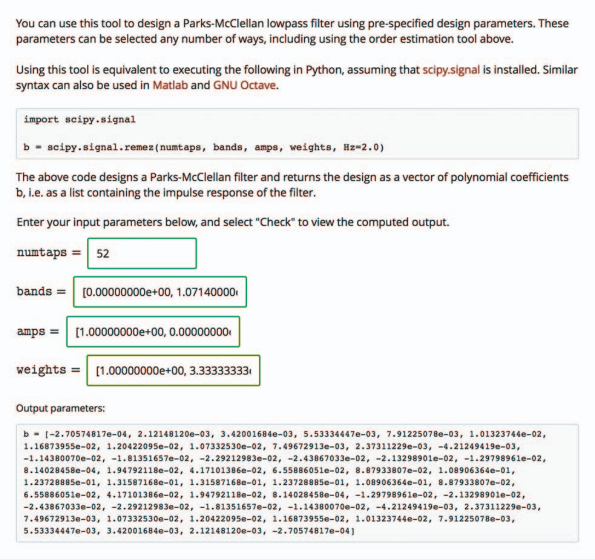
In 6.341 at the Massachusetts Institute of Technology (MIT), a numerically focused class project on IIR and FIR filter design was traditionally assigned as a key component of the course. In developing 6.341x, one of our goals was to provide a project in this theme for online learners. With this in mind, we wanted to deliver a numerical project experience that was contained entirely in the browser, provided graphical and textual feedback to the learner about how and where their numerical input might be incorrect, and did not require the learner to have extensive knowledge of a particular numerical signal processing language, yet still provided practice dealing with many of the numerical issues associated with using such packages.

With these goals in mind, our approach in 6.341x was to write a series of server-side Python libraries that allowed learners to perform order estimation and filter design by submitting design parameters to the edX server, which would then return the numerical output that was computed from the parameters. Learners were also provided with associated Python/SciPy code to reproduce these results on their own machine if they wished, although very little knowledge of Python was required to use the online tools.

Using the numerical designs obtained from the in-browser tools, learners were able to complete various assessment problems, and basic code used in performing the assessment was provided to learners to use as a reference if they wished. Graphical feedback based on learner input was generated dynamically on the server and passed to the browser, e.g., highlighting regions of the magnitude response where a numerical design might not have met the stated specifications.

From the perspective of the student, the process of designing a particular filter typically involved the following sequence. First, an order estimation tool would be used, as depicted in Figure S3. The returned values could then be used to select parameters in a corresponding filter design tool, shown in Figure S4. By transferring the resulting numerical design to the appropriate assessment problem, the learner was provided with automated, graphical feedback about his or her design, as is depicted in Figure S5.

The overall reaction to the exercises using these tools was positive. Learners indicated, in particular, that the freedom to explore various design methods and parameter choices was a key part of the learning experience, highlighting to us the value of providing access to numerical tools and problems in an online signal processing course.



You can use this tool to design a Parks-McClellan lowpass filter using pre-specified design parameters. These parameters can be selected any number of ways, including using the order estimation tool above.

Using this tool is equivalent to executing the following in Python, assuming that scipy.signal is installed. Similar syntax can also be used in Matlab and GNU Octave.

```
import scipy.signal

b = scipy.signal.remez(numtaps, bands, amps, weights, Hz=2.0)
```

The above code designs a Parks-McClellan filter and returns the design as a vector of polynomial coefficients b, i.e. as a list containing the impulse response of the filter.

Enter your input parameters below, and select "Check" to view the computed output.

numtaps = 52

bands = [0.00000000e+00, 1.07140000

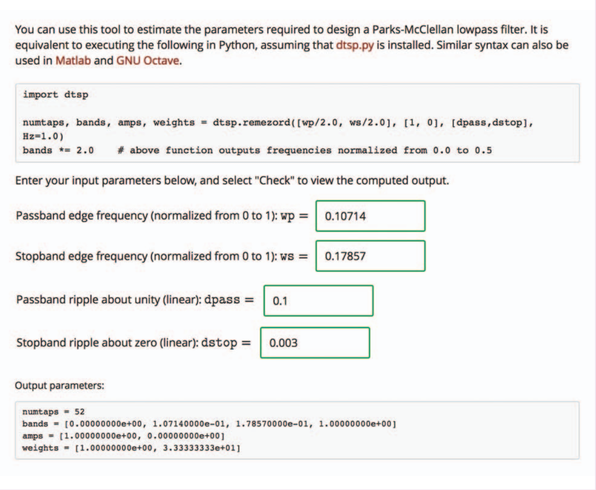
amps = [1.00000000e+00, 0.00000000

weights = [1.00000000e+00, 3.33333333

Output parameters:

```
b = [-2.70574817e-04, 2.12148120e-03, 3.42001684e-03, 5.53334447e-03, 7.91225078e-03, 1.01323744e-02,
1.16873955e-02, 1.20422095e-02, 1.07332530e-02, 7.49672913e-03, 2.37311229e-03, -4.21249419e-03,
-1.14380070e-02, -1.81351657e-02, -2.29212983e-02, -2.43867033e-02, -2.13298901e-02, -1.29798961e-02,
8.14028458e-04, 1.94792118e-02, 4.17101386e-02, 6.55886051e-02, 8.87933807e-02, 1.08906364e-01,
1.23728885e-01, 1.31587168e-01, 1.31587168e-01, 1.23728885e-01, 1.08906364e-01, 8.87933807e-02,
6.55886051e-02, 4.17101386e-02, 1.94792118e-02, 8.14028458e-04, -1.29798961e-02, -2.13298901e-02,
-2.43867033e-02, -2.29212983e-02, -1.81351657e-02, -1.14380070e-02, -4.21249419e-03, 2.37311229e-03,
7.49672913e-03, 1.07332530e-02, 1.20422095e-02, 1.16873955e-02, 1.01323744e-02, 7.91225078e-03,
5.53334447e-03, 3.42001684e-03, 2.12148120e-03, -2.70574817e-04]
```

FIGURE S4. An in-browser tool for performing minimax-optimal FIR filter design.



You can use this tool to estimate the parameters required to design a Parks-McClellan lowpass filter. It is equivalent to executing the following in Python, assuming that dtsp.py is installed. Similar syntax can also be used in Matlab and GNU Octave.

```
import dtsp

numtaps, bands, amps, weights = dtsp.remezord([wp/2.0, ws/2.0], [1, 0], [dpass,dstop],
Hz=1.0)
bands *= 2.0    # above function outputs frequencies normalized from 0.0 to 0.5
```

Enter your input parameters below, and select "Check" to view the computed output.

Passband edge frequency (normalized from 0 to 1): wp = 0.10714

Stopband edge frequency (normalized from 0 to 1): ws = 0.17857

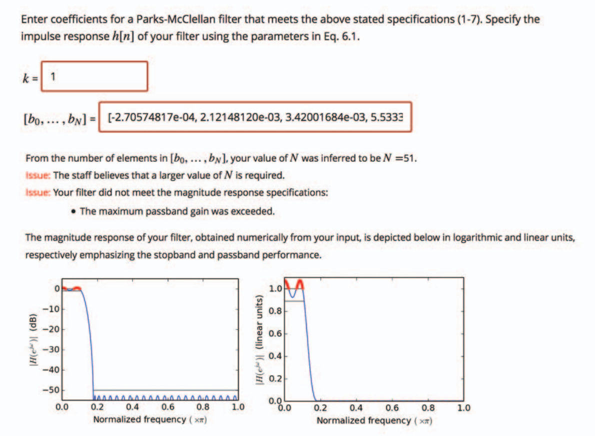
Passband ripple about unity (linear): dpass = 0.1

Stopband ripple about zero (linear): dstop = 0.003

Output parameters:

```
numtaps = 52
bands = [0.00000000e+00, 1.07140000e-01, 1.78570000e-01, 1.00000000e+00]
amps = [1.00000000e+00, 0.00000000e+00]
weights = [1.00000000e+00, 3.33333333e+01]
```

FIGURE S3. An in-browser tool for performing FIR filter order estimation.



Enter coefficients for a Parks-McClellan filter that meets the above stated specifications (1-7). Specify the impulse response $h[n]$ of your filter using the parameters in Eq. 6.1.

$k =$ 1

$[b_0, \ldots, b_N] =$ [-2.70574817e-04, 2.12148120e-03, 3.42001684e-03, 5.5333

From the number of elements in $[b_0, \ldots, b_N]$, your value of N was inferred to be N =51.

Issue: The staff believes that a larger value of N is required.

Issue: Your filter did not meet the magnitude response specifications:

- The maximum passband gain was exceeded.

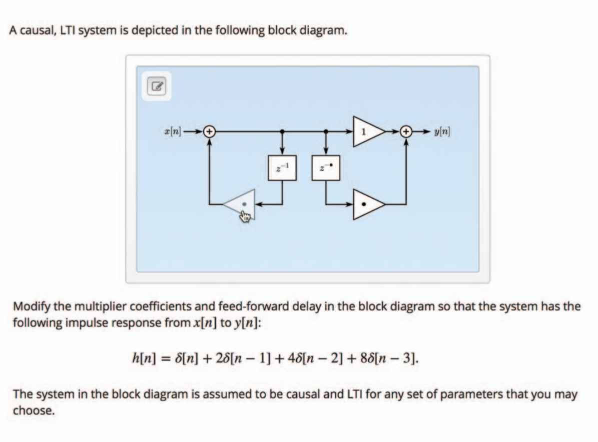
The magnitude response of your filter, obtained numerically from your input, is depicted below in logarithmic and linear units, respectively emphasizing the stopband and passband performance.

FIGURE S5. An exercise assessing a particular filter design entered by a learner, providing automated feedback about where the design exceeded specifications stated previously in the problem.

The Editor: A Browser-Based Tool for Manipulating Signal-Flow Diagrams (MIT)

In the residential course at MIT, on which 6.341x was based, a key focus was traditionally placed on using both symbolic and numerical exercises in assessing understanding and many times, in particular, using signal-flow and block diagrams in doing so. With these goals in mind, we wrote a variety of assessment exercises around what we referred to as "The Editor": an in-browser graphical tool that we built, designed specifically to give learners the ability to create and manipulate numerical and symbolic signal-flow diagrams directly inside the courseware.

The Editor is a JavaScript library that couples a declarative representation of a signal-flow structure with a graphical interface in which the representation can be manipulated. The signal processing representation (SPR) is based on extensible markup language (XML; referred to as SPRXML) and encodes the topology and parameters associated with a particular signal-flow block diagram. Using associated server-side libraries also written for the course, the edX server can dynamically generate an SPRXML system, pass the system to the browser where it is displayed



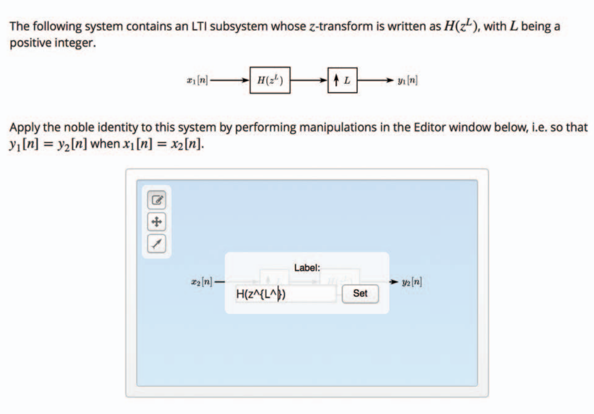
A causal, LTI system is depicted in the following block diagram.

Modify the multiplier coefficients and feed-forward delay in the block diagram so that the system has the following impulse response from $x[n]$ to $y[n]$:

$$h[n] = \delta[n] + 2\delta[n-1] + 4\delta[n-2] + 8\delta[n-3].$$

The system in the block diagram is assumed to be causal and LTI for any set of parameters that you may choose.

FIGURE S6. A problem for which learners specify block diagram parameters using The Editor.



The following system contains an LTI subsystem whose z-transform is written as $H(z^L)$, with L being a positive integer.

Apply the noble identity to this system by performing manipulations in the Editor window below, i.e. so that $y_1[n] = y_2[n]$ when $x_1[n] = x_2[n]$.

Label: H(z^{L^})

FIGURE S7. A problem assessing the application of the noble identity by performing manipulations involving symbolic expressions using The Editor.

Nissenbaum. Further behind the scenes are many others at edX and ODL, without whose active involvement and support the interactive numerical content would not have been able to run on the platform.

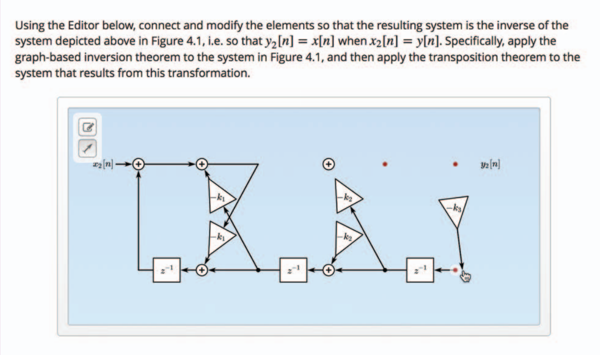
Unique elements

Discussion forum

A key component of 6.341x was the edX discussion forum that was very actively monitored by the course instructors (Tom Baran and Alan Oppenheim for the public MOOC, and Tarek Lahlou for the private industry beta), as well as by several community TAs (CTAs) from industry, who had participated in the limited beta run in fall 2014. On the forum, students engaged with one another and the staff, discussing course content and how it might be applied to their own engineering problems. By the end of the spring 2015 MOOC, a lively community of engineering professionals, students, independent learners, and educators had emerged on the 6.341x discussion forum.

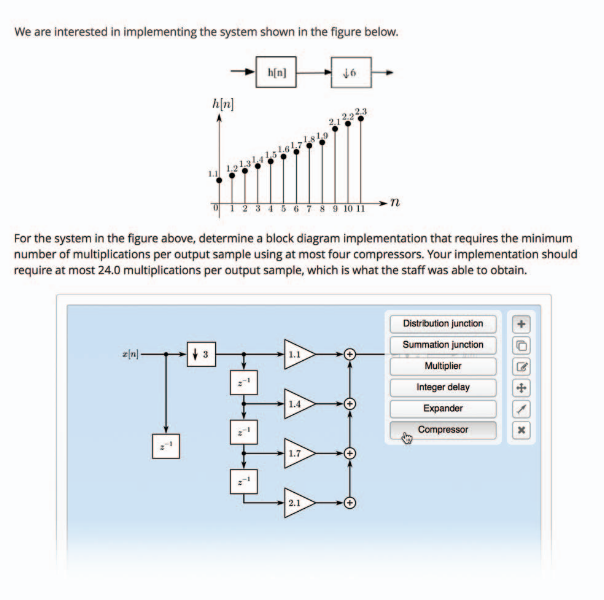
As indicated in Figure 6, the online discussion forum was the lowest-latency mode of student–staff interaction and, as such, was very actively used throughout the duration of the course. There were typically two types of content-oriented posts on the forum: specific questions about homework problems (typically generated by students) and regular more-elaborate posts written by the staff, designed as a springboard for higher-level discussion about various signal processing concepts. For the spring 2015 MOOC, a total of six CTAs were also available on the forum, selected from those students who performed well previously in the industry beta version.

Based on course feedback, students generally felt 6.341x staff to be very accessible via the online forum. The staff regularly monitored the progress of threads on the forum and encouraged discussion among the students, e.g., by posting comments and follow-up questions. Staff responses to questions about homework problems were intentionally delayed somewhat unless an error had been identified, giving students the opportunity to respond first and further encouraging students to view the forum as a collaborative meeting place among a community of learners, as opposed to as a resource for homework help from the staff.



Using the Editor below, connect and modify the elements so that the resulting system is the inverse of the system depicted above in Figure 4.1, i.e. so that $y_2[n] = x[n]$ when $x_2[n] = y[n]$. Specifically, apply the graph-based inversion theorem to the system in Figure 4.1, and then apply the transposition theorem to the system that results from this transformation.

FIGURE S8. An Editor-based problem assessing the detailed application of graph theorems.



We are interested in implementing the system shown in the figure below.

For the system in the figure above, determine a block diagram implementation that requires the minimum number of multiplications per output sample using at most four compressors. Your implementation should require at most 24.0 multiplications per output sample, which is what the staff was able to obtain.

FIGURE S9. A system implementation exercise in which learners use The Editor to specify an efficient multirate system.

by The Editor, and after the learner manipulates the system graphically, the server grades the modified SPRXML.

In 6.341x, a variety of problem types were written around The Editor. For example, The Editor was used in several problems to display a dynamically generated block diagram having parameters that were updated on each attempt. This was used by learners who desired repeated practice in computing transfer functions. Learners could also use The Editor to modify parameters in a block diagram having a fixed topology, as shown in Figure S6. By using the LaTeX-like equation syntax supported by The Editor, in conjunction with a symbolic grader, the course was also able to assess the ability of a learner to apply key signal processing identities, such as the noble identity as depicted in Figure S7. Detailed assessments of the use of identities were also possible using The Editor, such as in the problem shown in Figure S8, which assesses the application of the transposition and graph-based inversion theorems. Using The Editor, block diagrams could also be created from scratch. The problem depicted in Figure S9 uses The Editor in assessing the ability of a learner to create an efficient implementation of a multirate system.

Platform augmentation and development

A key goal in developing 6.341x was to design content that pushed the boundaries of and extended the edX platform. This goal was made in collaboration with the MIT ODL and edX and was done for several reasons. For example, we considered it important to provide online access to numerical tools for students who might not have had specific signal processing packages available on their machine (e.g., MATLAB, LabView, etc.). Another key reason was that material in 6.341x is intended to teach concepts and not specific languages. This has historically been true, even with those problems and projects in the residential course that are numerically focused. With this in mind, on the 6.341x site, various numerical tools were provided. Parameters could be entered into the browser, the input would be evaluated on the server, and in addition to providing the numerical result, Python code would be provided so that students could deploy the result on their own system, if they wished. An example of the workflow associated with using these tools in the context of FIR and IIR filter design is depicted in Figures S7–S9 and discussed in "Numerical Tools for Filter Design (MIT)."

Another tool developed for 6.341x is what became known as The Editor, a JavaScript-based interface for graphically entering symbolic block diagrams that was used in a variety of problems and exercises. The use of The Editor is depicted in Figure 5(d) and Figures S3–S6, and is discussed in greater detail in "The Editor: A Browser-Based Tool for Manipulating Signal-Flow Diagrams (MIT)." Using The Editor, a student could enter systems composed of standard signal processing blocks such as summation nodes, coefficient multipliers, expanders, and decimators. Unlike various traditional signal-flow entry tools, The Editor supported the entry of blocks having symbolic parameters. A learner could be asked, for example, to symbolically apply a noble identity or transposition theorem to a preloaded signal-flow system, and the autograder would symbolically evaluate whether the properties had been correctly applied.

There were also problems in which learners used The Editor to input block diagrams from scratch, implementing, for example, a multirate system having a desired response. In this case, the entered block diagram would be scheduled and implemented on the edX server, with the resulting automatically generated implementation used as a basis for evaluation. Motivated by the goal of providing detailed feedback to students, the scheduling algorithm was also able to automatically analyze and reduce algebraic loops, forming an online implementation of the algorithm as shown in [16].

Class by the numbers

Retention and engagement

In the spring 2015 run of 6.341x, the course began with approximately 10,500 learners registered and closed with approximately 9,500 learners registered and 110 receiving certificates. Early in the course, about 2,500 registrants were clearly active, with approximately 500 active at the close of the course. Perhaps not surprisingly, the largest spikes and subsequent drop offs in the number of registrants that were active coincided with the three exams.

Demographics

The residential course 6.341 is taught once each academic year at MIT, with typical end-of-semester enrollment of 30–40 (mostly) graduate students. With the acceptance rate of 2–3% in the MIT EECS graduate program, it is reasonable to assume that the students in the residential course have strong backgrounds and are well qualified. Attrition rates in 6.341 are typically on the order of 30% between the first and last weeks of the semester.

In the industry beta version of 6.341x, there were a total of approximately 170 registered students, with 36 completing the course with a passing grade (i.e., performance at the level of an A or B grade). In the spring 2015 open run of the MOOC, the initial registrants represented a total of 136 countries. Among those who successfully completed the spring 2015 MOOC, the overwhelming majority were those who joined from industry. Keeping in mind our goal of reaching a broader number of individuals than we were able to reach residentially at MIT, we were delighted that while maintaining the same level of content depth, difficulty, and sophistication, we were able in a single run of the course to impact the same number of students as would be impacted residentially during approximately six–seven years of teaching at MIT. In addition, as further evidenced by the participant quotes and the modest percentage of participants able to successfully complete the course, it was difficult and demanding in terms of time and background, as would be expected with a course at the graduate level. It was also especially encouraging that a significant percentage of those impacted by the MOOC were individuals who do not traditionally constitute a major component of the MIT EECS graduate-level student demographic; specifically, those completing 6.341x were primarily university faculty and senior-level engineers working on projects at well-known high-tech firms.

Much in the same way, the openness of online courses often causes prospective learners to underestimate the importance of prerequisites, which leads to the low yield rates that most other instructors experienced.

Impact on the MIT residential course

We have a very clear and strong sense of how MOOC content has enhanced the residential experience and its impact for the future. Although MOOC content by its nature is designed to stand alone, in the residential context it becomes a very strong multiplier on the in-class time that students have with the staff. In the residential course, students were strongly encouraged to preview the online content before coming to class, and the assumption during class time was that students had previewed the videos and worked at least some of the online exercises. In the context of a graduate-level course that carries an assumption of a high level of maturity and commitment on the part of students, we chose not to make previewing explicitly mandatory, nor to have a graded mini-quiz at the beginning of each in-class session. Furthermore, class sessions were not simply discussions or Q&A sessions. However, the level and pace of the presentation was predicated on the assumption of previewing. The approximate experience was that about one-third of the class regularly previewed the topics, one-third sporadically or superficially did so, and about one-third almost never did at all. Those who did uniformly (anecdotally) felt that it made a significant difference. And whether or not a student regularly previewed the content, virtually all actively used the online content after in-class sessions.

Our conclusion is that the residential course benefited enormously from the availability of the rich online content and that students actively used it both before and after the in-class interaction. The MOOC by itself is no substitute for a well-taught residential course. However, it can be a significant enhancement to any residential course, and in our view it is a strong substitute for any poorly taught residential course.

Feedback

Overall feedback about 6.341x was strongly positive, and perhaps the most common negative feedback was that the level of sophistication and time commitment required to take 6.341x was higher than expected. However, for those who met the background prerequisites, this was viewed as an asset, commenting that 6.341x stood apart from other MOOCs in this regard. Those students whose background was slightly weak but who actively engaged with the forum generally found that the availability and encouragement of course staff and CTAs allowed them to brush up on their weak spots and stay engaged with the course.

As expressed by one of the students, “Right before the course started, I thought I was well prepared for this course. After all, I have a strong background in [signals and systems], and I am very familiar with digital signal processing. But after the first few weeks…I found the exercises and problem sets challenging…. I almost gave up [were it not for] the helpful feedback from instructors, TAs, and other kind peers." Regarding the forum, a student commented, "The instructors did a fantastic job in interacting with the students. I cannot recall one single question that was not properly addressed. They are very kind and responsive." To us, this indicated that better learning outcomes in 6.341x were strongly facilitated by a high level of staff involvement on the forum.

In terms of the organization of the courseware, students reacted especially positively to the organization of the course topics, verifying the staff's intuition that weaving video segments with short exercises would provide a natural mechanism for students to check their understanding before moving too far along. The interactive numerical content in the courseware was well received and generally viewed as a unique feature of 6.341x. Some students had been hoping for an opportunity to write and test their own signal processing code, which fell outside of the scope of the interactive problems. Many of those students desiring additional code practice were typically sufficiently motivated to do so on their own and post the results of their efforts on the discussion forum, significantly enriching the discussion among the community of learners.

A number of individuals also commented that, overall, 6.341x had a significant impact on them both personally and professionally. In particular, students indicated the immediate applicability of 6.341x to their professional work in fields ranging from software-defined radio to the design of particle accelerators. A recurrent theme in student feedback was also that 6.341x helped them to decide to change their field to signal processing, which the staff was delighted to hear. Overall, these comments indicate to us that 6.341x is a challenging course, but for those who complete it, it is also a very rewarding experience.

Next steps

Going forward, there are several potential modes and roles for 6.341x content. In the context of its incorporation into the MIT residential course, there is no question that it has contributed significantly to the educational experience, and it will continue to be incorporated and developed further in the context of the MIT residential graduate course. We welcome the opportunity for it to find a similar role in many residential courses at this level elsewhere.

In the form that 6.341x ran as a MOOC in spring 2015, the content was released on a fixed schedule, and learners were expected to commit to that schedule. Although that mode of delivering difficult course content of this depth and sophistication is typical in a university environment, rigid pacing is not necessarily well matched to participants outside of the university environment or to those with other significant time constraints and deadlines. In future deployments of 6.341x, we envision a self-paced mode on a more local platform so that pacing of the content can accommodate the needs of particular groups of participants. As one model for use in an industry environment, a company could perhaps subscribe to the content and platform for use internally. The oversight, pacing, and staffing of the course could then be managed internally to match the needs and schedule constraints of participants.

Our current plan in the near term is to release the total 6.341x course content in a form that is freely available to learners on the edX platform, for the purpose of self-study. In this mode, autograding the exercises and problem sets will be activated, but no discussion forum or support staff will be available. Specifically, it will be accessible in a manner similar to that of course content on MIT OCW, although with a more interactive component. When it becomes available, the material will be accessible at http://www.rle.mit.edu/dspg/6.341x.

Conclusions

The experience of the three DSP courses presented in this article clearly suggests that online platforms and content offer rich opportunities for teaching signal processing. How to best affect this is not yet clear, and "best practices" can be very dependent on the demographics of the learners and the objectives and personal style and preferences of the course developers and instructors in adapting residential course content to an online environment.

Commonalities

Perhaps the most important commonality across the three courses is the focus on solid theoretical foundations. In that and a number of other respects, they have a different purpose and target audience than many other online courses, for which content is primarily oriented toward a high-level overview of a topic area. This difference is clearly a key factor in the drop-off level of active involvement from registration (i.e., many registrants are motivated mainly by curiosity) to course completion. This is inevitable for online courses that attempt, to the extent possible, to provide participants with the same depth and sophistication as a residential course. It is also important to recognize that a MOOC is no substitute for a well-taught residential course that incorporates significant interactive face time with a knowledgeable and motivated staff.

Because the three courses are based on residential classes at different levels, the backgrounds and expectations of the participants somewhat differ. However, in a broad sense, a serious background in signals and systems at some level was common, and quite often, a more advanced background including industrial project experience with a partial motivation to refresh that background was helpful.

Differences

Two of the courses were offered on edX and one on Coursera. The differences between platforms are certainly many but not profound enough to significantly affect the way the material was structured and presented. The three courses were, in fact, more distinct in their handling of numerical exercises and examples. EPFL ultimately gravitated toward Python (via IPython Notebooks), Rice experimented with a tight integration between MATLAB and edX, and MIT developed specific extensions to the edX platform to provide in-browser numerical exercises independent of any specific package or programming language.

Each of these approaches has potential advantages and drawbacks. MATLAB offers perhaps the most complete signal processing sandbox and a very user-friendly learning curve, but its scripting language does not please those students with a more rigorous background in computer science.

Furthermore, it is a commercial solution, and although free alternatives do exist, they are not as complete and robust. As an alternative, Python is becoming increasingly popular in scientific programming, but the language itself is still embroiled in a difficult version transition. IPython Notebooks are a very versatile didactic tool, but they do not scale well to large projects and do not offer easy version control. Adhoc code and browser extensions are clearly the most attractive approach with respect to integration with lectures and from the point of view of user experience; but, inevitably, they impose a very high development and maintenance cost on teaching staff.

Lessons learned from our MOOC experiences

- A very clear and strong lesson we all learned is that the resources needed in terms of effort, financing, and platform backup to successfully develop and run an online course with serious depth and content are enormous. Video segments need to be short and crisp and, even if extracted from in-class video recordings, major editing is essential. Exercises, problems, and projects all need to be carefully designed and restructured, even when based on residential course content.
- The backgrounds of MOOC participants are typically very diverse. For our courses, they included educators, experienced engineers, high school and college students, and retirees. Making clear to potential participants the assumed background required and providing a preliminary background exam for their calibration before registering would seem essential.
- In stark contrast to a residential setting using MOOC and other online resources, in a MOOC setting, it is extremely difficult to exercise and test deep understanding of concepts.
- The opportunity for students to receive immediate feedback as they work through exercises, problems, and projects is a key defining feature of teaching when using online resources, whether residentially or as MOOCs.
- With the large number of participants in a MOOC and the analytics that the platforms can capture and provide, there are unprecedented amounts of data on what works and what can be improved; these data are in the form of direct feedback from the students on the forums, indirect observation of self-regulating conversations among students, performance on the various elements, and information on the use of the videos. MOOC platforms in use today log every interaction between the learner and the interface: timestamps, number of views, fragmentation of video consumption, and access to previous material are just a few of the variables to which we now have access. For now, these data remain largely untapped. Clearly, the insight from such a vast data set would benefit not only online teaching but residential courses as well.
- The emergence of volunteer CTAs is one of the things that makes MOOCs truly different from usual residential courses. It illustrates that educational communities have the potential for emergent behavior, where students mentor and tutor each other with little interaction from an instructor. Moving forward, it is important to find ways to incentivize and support this very positive and useful behavior.

The role of certification

It seems clear anecdotally that a high percentage of MOOC participants value some form of certification of their successful completion of the course. What we find to be less clear is the inherent value of a statement of accomplishment, how it might fit into a student's curriculum, and how a professional can leverage its value in the workplace. Our institutions have been very prudent in the wording of certificates and have made sure to prevent any association between MOOCs and the actual on-campus curriculum. Clearly, any other course of action would be difficult in the absence of a reliable method to assess the identity (and the proficiency) of online students. If it is already hard to manage the test administration process on campus, the difficulties online are close to insurmountable: Multiple identities are easy to forge, cheating is easy, and unless exams are constantly rewritten, solutions from previous editions of the class are just a click away. In this sense, the "massive" and "open" characteristics of MOOCs are also their liability as far as proper certification is concerned. Much in the same way, the openness of online courses often causes prospective learners to underestimate the importance of prerequisites, which leads to the low yield rates that we (and most other instructors) experienced. Once again, it will be difficult to arrive at a compromise for which the original spirit of the MOOC "revolution" coexists with a preselection process.

The big picture, with an eye on the future

We started by evoking "the year of the MOOC" and conclude with some reflections on the future of MOOCs, given our collective experience so far. Perhaps the first observation is that, as is widely recognized in the community, a clear business model for making MOOCs financially viable and sustainable has yet to emerge. Online courses require a huge amount of work to design, realize, and sustain, which adds up to a significant financial investment on the part of the sponsoring institutions. Yet completion rates are so low that any residential class with similar drop-off rates would be unsustainable. Potential solutions lie in directions such as the evolution toward specialization classes with fewer and prescreened participants and/or the targeted professional market. Such directions, of course, are no longer "massively open" but continue to take advantage of the enormous benefits of the online environment. Perhaps the harsher realization is that a deep understanding of a topic is built from a solid foundational background and then serious and hard work to advance that background into a deeper and richer understanding. Absorbing difficult content is, well, difficult, no matter which delivery channel is used to reach the students.

These difficulties should not overshadow the enormous potential of MOOCs and the content that they contain. They are typically based on packaging extremely high-quality material from on-campus classes into an attractive format accessible by anyone from anywhere. There is a strong similarity to the process of evolving course notes, often hastily typed and poorly photocopied, into high-quality textbooks that then impact a much wider audience. And, of course, the significant effort that goes into polishing course notes into widely available textbooks, often underestimated, has a major impact on the residential teaching of that content, both locally and more broadly. Use of online MOOC content to enhance residential teaching of the material appears to have enormous potential and, as with well-written textbooks, can provide enormous leverage to dedicated residential courses worldwide. However, there is often the misconception that incorporation of MOOC content and other online content will lead to cost savings and reduced required effort. In our experience, this is not the case if teaching quality remains important. Incorporation of this content into residential courses has the potential for enormous leveraging and enhancement to materially increase the quality of the education. These are indeed exciting times for education!

Authors

Thomas A. Baran (tbaran@alum.mit.edu) received his M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT). He has been a visiting lecturer and MIT and edX (MITx) fellow at MIT and is a cofounder and chief executive officer of Lumii, Inc. His research interests include the intersection of signal processing with optimization theory as well as light-field signal processing and audio signal processing. He has received awards from the IEEE and MIT in recognition of research, teaching, and mentorship.

Richard G. Baraniuk (richb@ece.rice.edu) is the Cameron Professor of Electrical and Computer Engineering at Rice University in Houston, Texas. He received the IEEE Signal Processing Society Best Paper Award, Best Column Award, Education Award, and Technical Achievement Award. For his education projects, including Connexions and OpenStax, he received the National Outstanding Teaching Award from Eta Kappa Nu, the Tech Museum Laureate Award, the Harvard Law School Internet Pioneer Award, the World Technology Education Award, and the IEEE James H. Mulligan, Jr. Medal for Education. His research interests include signal processing, machine learning, and open education. He is a Fellow of the IEEE and the American Association for the Advancement of Science.

Alan V. Oppenheim (avo@mit.edu) is Ford Professor of Engineering in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology (MIT). He is a coauthor of a number of widely used textbooks on signal processing. He received an honorary doctorate from Tel Aviv University and was a Guggenheim fellow and a Sackler fellow. He received multiple IEEE awards, IEEE Signal Processing Society awards, and MIT awards in recognition of outstanding research, teaching, and mentoring. He is a member of the National Academy of Engineering. He is a Life Fellow of the IEEE.

Paolo Prandoni (paolo.prandoni@epfl.ch) is a lecturer at the École Polytechnique Fédérale de Lausanne, Switzerland. He coauthored the textbook *Signal Processing for Communication* and designed the first digital signal processing massive open online course for Coursera in 2012. He is a cofounder and chief science author of Quividi, a leading video analytics company for out-of-home audience measurement.

Martin Vetterli (martin.vetterli@epfl.ch) received his Dipl.-Ing degree from the Swiss Federal Institute of Technology, Zürich, in 1981; his M.S. degree from Stanford University, California, in 1982; and his Ph.D. degree from the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 1986. He held faculty positions at Columbia University, the University of California, Berkeley, and EPFL, where he is a professor. Since 2013, he has been the president of the Swiss National Science Foundation. His research interests include electrical engineering, computer sciences, and applied mathematics. He is the coauthor of three textbooks and numerous papers and patents and received several awards for his research. He is a fellow of the Association for Computing Machinery and a foreign member of the National Academy of Engineering. He is a Fellow of the IEEE.

References

[1] T. Friedman, Revolution Hits the Universities, *The New York Times*, Jan. 26, 2013.

[2] A. V. Oppenheim, “Videotape lecture series and study guide on digital signal processing,” MIT Center for Advanced Engineering Studies, Sept. 1975. [Online]. Available: http://ocw.mit.edu/resources/res-6-008-digital-signal-processing-spring-2011/

[3] A. V. Oppenheim. (1987, Apr.). Videotape lecture series and study guide on signals and systems: An introduction to analog and digital signal processing. *MIT Center for Advanced Engineering Studies*. [Online]. Available: http://ocw.mit.edu/resources/res-6-007-signals-and- systems-spring-2011/

[4] P. Prandoni and M. Vetterli, *Signal Processing for Communications*, 2008. [Online]. Available: http://www.sp4comm.org

[5] [Online]. Available: https://www.ctan.org/pkg/dsptricks

[6] H. J. Trussell and D. Baron, “Creating analytic online homework for digital signal processing,” *IEEE Signal Processing Mag.*, vol. 32, no. 5, pp. 112–118, Sept. 2015.

[7] [Online]. Available: http://ipython.org/notebook.html

[8] [Online]. Available: https://github.com/LCAV/SignalsOfTheDay.

[9] M. Martinez-Camara et al., “The Fukushima Inverse problem,” in *Proc. 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* Vancouver, British Columbia, Canada, pp. 4330-4334.

[10] I. Dokmani, “Listening to distances and hearing shapes: Inverse problems in room acoustics and beyond,” Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2015.

[11] [Online]. Available: https://en.wikipedia.org/wiki/Johann_Heinrich_Pestalozzi

[12] [Online]. Available http://www.katyjordan.com/MOOCproject.html

[13] [Online]. Available: http://openstax.org

[14] R. G. Baraniuk. (2016). ELEC301.2x: Discrete-time signals and systems, part 2: Frequency domain. Rice Univ. [Online]. Available: https://courses.edx.org/courses/RiceX/ELEC301.2x/2015Q3/info

[15] A. V. Oppenheim and T. A. Baran. (2015). 6.341x Discrete-time signal processing. *edX*. [Online]. Available: https://www.edx.org/course/mitx/mitx-6-341x-discrete-time-signal-4396

[16] T. A. Baran and T. A. Lahlou, “Implementation of interconnective systems,” in *Proc. 2015 IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP),* South Brisbane, Australia, pp. 1101-1105.

[17] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2010.

[18] A. V. Oppenheim and G. Verghese, *Signals, Systems and Inference*. Englewood Cliffs, NJ: Prentice Hall, 2016.

[19] R. G. Baraniuk. (2015). 6.341x Discrete-time signal processing. *edX*. [Online]. Available: https://courses.edx.org/courses/RiceX/ELEC301x/T1_2014/info

[20] R. G. Baraniuk. (2016). RiceX: ELEC301.1x discrete time signals and systems, part 1: Time domain. *edX*. [Online]. Available: https://courses.edx.org/courses/RiceX/ELEC301x_/2015Q3/info

[21] Signals and Systems. *OpenStax CNX*. [Online]. Available http://cnx.org/contents/d2CEAGW5@15.4:l-e_wSqM@24/Signal-Classifications-and-Pro.

[22] Weather data from Jena, Germany. [Online]. Available: http://www.bgc-jena.mpg.de/martin.heimann/weather/

[23] Astronomy Picture of the Day. [Online]. Available: http://apod.nasa.gov/apod/astropix.html