Overview

Eric Jonas (jonas@eecs.berkeley.edu)

June 8, 2015

This is an experiment. I need your feedback to make sure it's working for you.

1 Background

Have you taken

1. A linear algebra course
2. A differential equations course
3. A course on signals and systems
4. A DSP course
5. A stochastic process theory course
6. Have you called fft in the past year

2 Introduction

The importance of consistent pedagogy – $x(t)$ vs $x[n]$

The unimportance of formalism – this is a rapid survey (drinking from the firehose) and the formalism sometimes gets in the way. That said, if you wish to do actual work in this area, it's important to at least collaborate with someone who knows it.

3 About Me

I'm not a mathematician, or really even a computer scientist, but rather just a guy who wants to build cool shit.

4 What we don't cover

- **Control theory**: Control theory is one of the great successes of 20th century engineering, and has enabled so much of the amazing devices we have today. Laurent Lessard put together a great introduction that I can refer you to if you're interested. There are need connections between control theory and recent active learning work (which we will not talk about)
- **Stochastic processes**: The real world is stochastic, as is most of your data. Stochastic process theory formalizes this, and is something even CS students should know, as it underlies large swaths of machine learning. That said, LTI systems like those we discuss impact simplistic stochastic processes in about the way you would expect, so there's not a lot of "surprising" material here.

- **Filter design**. Of course, once you have this incredible framework for signal processing, you want to then process some signals, and to do this you often ened to engineer filters. This used to be a very deep topic that has been mostly reduced to a series of matlab function calls.
- **advanced modulation schemes / information theory** The bleeding edge of the PHY layer involves state-of-the-art modulation schemes as well as lots of recent work in coding theory. We skip all of this .

5 Format

- 7 lectures
- heavy on background at first
- Lots of neat examples (i hope)
- If you want supplemental material let me know

6 What I hope to achieve

1. Get people excited about these problems
2. in two years you remember enough keywords to intelligently google the right thing and not have the wikipedia page scare you
3. Realize that computer science has been doing a lot of this stuff forever, and that you really do have the background and skills to contribute.