Teaching Instrumentation and Data Analysis Using Python

1 author:

Matti Pastell
Natural Resources Institute Finland (Luke)
183 PUBLICATIONS   **2,710** CITATIONS

SEE PROFILE

Teaching Instrumentation and Data Analysis Using Python

Matti Pastell

CIGR2016

Luke
NATURAL RESOURCES
INSTITUTE FINLAND

Introduction

- **2010** we decided to integrate teaching of programming and data analysis in "Measurement Technology" course

- First masters level course for Agrotechnology students in the University of Helsinki. ~10 students / year, 7 ects

- Course contains lectures, practical measurement exercises and reports with data analysis

- Before that the course used mainly existing measurement software and MATLAB for data analysis.

- I taught the course annually between 2011 - 2014

The requirements for programming language

- Started with C# and MATLAB, but switched to Python
  - Motivation for using C# was to use .NET Micro for microcontrollers
- Drivers for DAQ hardware
- Scientific stack
  - Data analysis
  - Signal processing
  - Simulation
  - Computer vision

- Freely and easily available for homework on multiple operating systems
- Support for GUI development

The requirements for programming language

- Usable in later courses, projects and thesis work
  - Agricultural automation
  - Field robot event – project course
  - Data analysis

- "Easy" to learn
  - Simple and consistent syntax
  - Get started quickly with not too much boilerplate code

- External learning resources available

Python

- BSD licensed dynamic general purpose scripting language
- Minimal syntax
- Multiparadigm:
  - Imperative
  - Object oriented
  - Functional

- Used in several universities (MIT, Wageningen)
  - EDX and Coursera courses available
 - Several good free and non-free books exist

Very first program

```
 7
 8 import u3
 9 import matplotlib.pyplot as plt
10
11 LJ = u3.U3()
12
13 data = []
14
15 for i in range(10):
16     V = LJ.getAIN(0)
17     data.append(V)
18     print(V)
19
20 plt.plot(data)
21
22
23
```

Used hardware

Python setup and essential libraries

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:
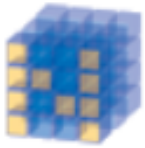
NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

OpenCV

ANACONDA

Freely available scientific Python distribution for Windows, OSX and Linux

Luke
NATURAL RESOURCES
INSTITUTE FINLAND

```
from scipy import signal
[b, a] = signal.cheby2(12, 80, 0.2)
plot_filterz(b,a)
```
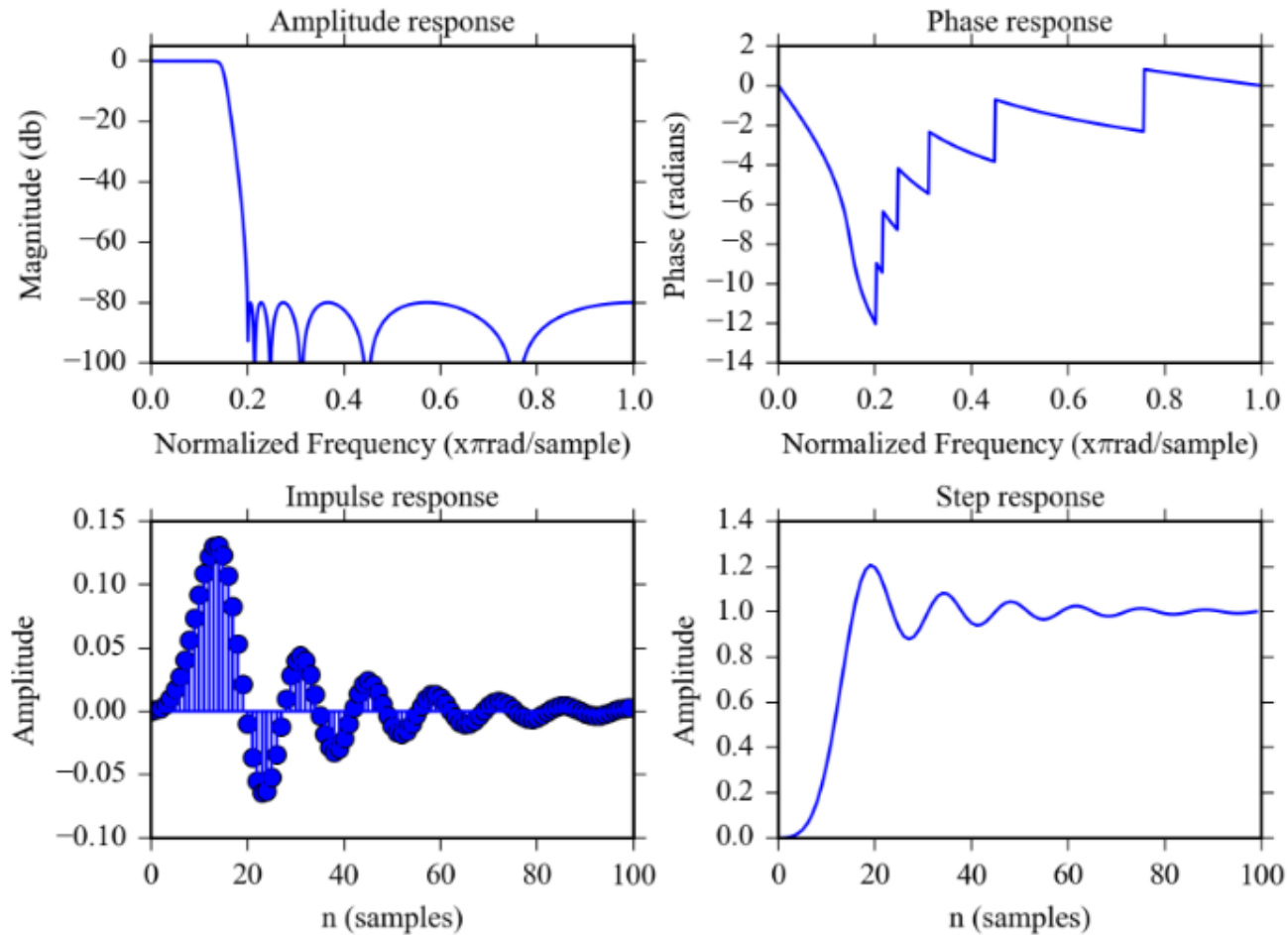

Figure 8.5:   Properties 12th order Chebyshev II lowpass filter

Experiences

- Python is very suitable for teaching the course and received good feedback from the students
  - The selected libraries are proven in real world applications, robust and stable
  - "Easy to get started", "understandable"
  - Interacting with hardware is seen as motivating way to be introduced to programming

- Large difference in the learning speed for the students
  - It would better to have a sepate programming course or increase the hours in the course

NATURAL RESOURCES
INSTITUTE FINLAND

Future prospects

- Use Python also for exercises in other courses
- Teaching mathematics via Python? e.g at Coursera *"Coding the Matrix: Linear Algebra through Computer Science Applications"* by Philip Klein
- A lot of low cost embedded hardware using Python has been introduced after the course
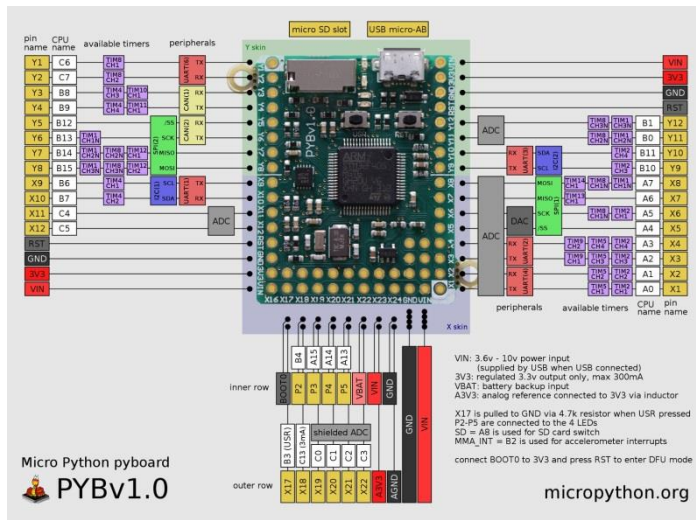
Kirjanmerkit

- Preface
- Measurement basics
- Datalogger hardware and sensor connections
- Connecting loggers to computers
- Introduction to Programming and Python
- Scientific Python
- Plotting
- Frequency Analysis
- Digital Signal Processing
- References

MEASUREMENTS AND
DATA ANALYSIS
for Agricultural Engineers
Using Python

Matti Pastell

The lecture notes are available at http://pyageng.mpastell.com for free

Luke
NATURAL RESOURCES
INSTITUTE FINLAND