Programski jezik Python: primena u merenjima

Predrag Pejović

ko sam ja?

redovni profesor na Elektrotehničkom fakultetu u Beogradu
Katedra za elektroniku, energetski elektroničar (uglavnom . . .)

- **B.S.** University of Belgrade, 1990.
  Naslov teme: „**Programska podrška** sistema za vektorsko upravljanje asinhronim motorom“
- **M.S.** University of Belgrade, 1992.
  Naslov teme: „Bilateralni pretvarač naizmeničnog napona u jednosmerni sa približno jediničnim faktorom snage“
- **Ph.D.** University of Colorado, Boulder, 1995.
  Thesis title: “A Method for **Simulation** of Power Electronic Systems Using Piecewise-Linear Device Models”

predajem . . .

šta nije pomenuto?

Srednje usmereno obrazovanje:

OVROMTS „Veljko Vlahović“
prevod: **Matematička gimnazija**

zanimanje: **Programer**

zainteresovani **korisnik** računara
. . . veoma zainteresovan . . .
. . . oni koji ne znaju da koriste računar bi rekli „zavisnik“

sve do 1995. . . .

. . . pauza . . .

dok se 2008. nije pojavio . . .

Hardy Heron!

comeback!

. . . i tada sam ponovo počeo da intenzivno koristim računar . . .

upozorenje!

U ovom predavanju iznosim svoje mišljenje!

- autor koristi i voli slobodni software
- u značenju “free software” po Richard Matthew Stallman
- nekima je to ideološki osetljivo . . .
- meni je korisno . . .
- zato i koristim free software . . .
- pa ako bas ne mozete da podnesete . . .
- sada je trenutak!

Python?

- programski jezik
- Wikipedia:
  - “Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Python claims to “[combine] **remarkable power with very clear syntax**”, and its standard **library is large and comprehensive**. Its use of indentation for block delimiters is unique among popular programming languages.”
  - “**The reference implementation of Python (CPython) is free and open source software** and has a community-based development model, as do all or nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation.”

Python??

- interpreter, scripting language
- po tome nalik na BASIC (nekada), GNU Octave, . . .
- nema kompilacije i linkovanja, vrlo brze probe
- sporije od C-a
- ali se dobro povezuje sa C-om
- jako moćne i raznovrsne biblioteke (numpy, matplotlib, sympy, pySerial, vxi-11, usbtmc . . .)
- jednostavna sintaksa
- opšta namena
- free!!!
- jako dobro podržan, razvija se, rasprostranjen
- Google, Youtube, . . .
- svaka distribucija GNU/Linux-a ga ima

Python???

- Guido van Rossum, December 1989, „novotarija“
- masovno se uči kao prvi programski jezik: MIT, . . .
- radi pod raznovrsnim platformama, sve koje se kod nas sreću obuhvaćene
- vrlo objektno orijentisan, mada ne mora da se koristi
- vrlo moćni tipovi podataka
- lako se prave novi tipovi podataka
- **gluing language**
- žao mi je ako vam kvari poslovni model, za mene odlično radi posao!

Python, dokumentacija

Ako ne učite programiranje, već programski jezik:

- `http://www.python.org/doc/`
- A4, pdf, zip, 9.5 MB
- Python 2.7.14, 16.12.2017.
- tutorial.pdf, Python Tutorial, **143** strane
- reference.pdf, The Python Language Reference, **134** strane
- library.pdf, The Python Library Reference, **1578** strana
- ↑ ovde je suština uspeha

Električna merenja . . .

- kako sam došao na Električna merenja . . .
- Caja i „Istorijski sporazum“
- interesovanje, sklonost, motivacija, entuzijazam . . .
- šta sam mislio o merenjima . . .
- i promene u dve faze . . .
- **sjajan marketing:**
  - Prof. Dragan Stanković:
  - . . . merenja su mukotrpna oblast . . .
  - . . . puno mukotrpnog rada . . . („glupog posla“)
  - . . . nezahvalna oblast . . .
  - . . . teško se publikuje . . .
  - . . . teško se nabavlja oprema . . .
  - ovo je marketing za oblast?
  - sve vreme se trudio da mi pomogne i pomagao mi je!

promene u dve faze . . .

1. na ispitu nema pisanja priča, memorisanja, precrtavanja, bojenja, . . . samo zadaci! (završeno 1997. godine)
2. laboratorija sa ispravnom primenom računara, započeto 2016. godine, nastavljeno 2017. godine kada smo dobili nove osciloskope . . .

zanimljivo, bilo je ljudi koji su više voleli prepričavanje, crtanje, bojenje, pisanje eseja od zadataka!

mada, nije sve bilo po planu . . .

- plan: **izbaciću** Lisažuove figure!
- realizacija: **proširio sam** nastavu iz Lisažuovih figura!

lični pristup . . .

- oblasti privlače ljude . . .
- kad mogu da biraju!
- oduvek sam mrzeo dosadne poslove!
- to me je i privolelo računarima, primena!
- Ewald Fuchs i domaći zadaci . . .
- a nije baš ni da smo imali izbora . . . ni instrumenata
- slučaj Terzija i Power Analyzer
- videćete da računari merenjima daju nov kvalitet
- možete nešto što nikako niste mogli ranije!
- pokupite odbirke, ostalo je matematika koju radi računar!
- . . . a Python je baš idealan za to!

Ko povezuje . . .

. . . i . . .

čovek!

- multimetar je danas mikrokontroler, računar ...
- danas se podaci obrađuju na računaru ...
- prenos podataka radi čovek: sveščica, olovčica, vreme, greške, popravljanje grešaka ...
- sjajna prilika za zapošljavanje čoveka izuzetnih intelektualnih sposobnosti i kreativnosti ...
- pobuna ako se takav posao automatizuje, izmišljanje razloga zašto je „po starom“ bilo bolje ...
- ljudi uvek vide sebe i svoj interes, štite svoje pozicije, treba da ih razumemo ...
- pa tako i ja, nadam se da ćete me razumeti; imamo ista prava!
- bluetooth, ideja za patent?

kasno!!!

- **kasno:** http://wiki.seeed.cc/Bluetooth_Multimeter/
- ili https://www.amazon.com/Digital-Multimeter-Temperature-Bluetooth-Interface/dp/B071JN59DW
- prikaz: https://www.youtube.com/watch?v=3yylv0_o1tg
- prikaz: https://www.youtube.com/watch?v=lhCx9rexvdI
- menja li nam Internet život?

idealna oblast za primenu računara!!!

- puno posla koji treba automatizovati
- i nisam se toga ja setio, još od 60's se radi na tome ... i to jako dobro i ispravno!
- „slobodna“ oblast, ljudi koji vole računare nisu u nju ulazili, naprotiv, „oblast bira ljude“
- dominirao skup proprietary software koji mi se nikada nije dopadao
- kada se malo zagrebe, sve je već bilo tu!
- samo nije bilo reklamirano ...
- posto “free” nema ko da reklamira
- opet sociologija!
- a ja sam ušao u engineering pošto Kirhofove zakone nije doneo Bundestag!

kolege ...

- moj (moj?) pristup užasno iritia „ideologe“, posebno one koji baš nikakvog racionalnog interesa nemaju, ali iracionalno uvek dominira nad racionalnim ...
- Skoplje ...
- ovdašnji ideolog, nedavno, ne radi merenja, ali „se razume”: „Pa šta imaš u Linux-u za merenja? Nemaš ništa!“
- da vidimo, ...
    1. Python
    2. NumPy
    3. SciPy
    4. matplotlib
    5. pySerial, python-serial
    6. vxi-11, python-vxi11
    7. python-usbtmc
    8. convert
    9. LaTeX
    10. tk-inter, Qt i Qt Designer
    11. ...

na čemu se zasniva automatizacija merenja?

standardizovana i dokumentovana komunikacija sa instrumentima!

1. standardizovane komande
   SCPI: Standard Commands for Programmable Instruments
   ovde je link na Wikipedia članak
   novotarija: prva verzija objavljena 1990. godine
2. standardizovani komunikacioni protokoli:
   2.1 nekada HPIB, GPIB, IEEE 488, IEEE 488.1, IEEE 488.2
       60's and 70's, još jedna novotarija!
   2.2 RS-232 (python-serial)
   2.3 USB (python-usbtmc)
   2.4 Ethernet/LAN (python-vxi11)

znam li šta o ovome? malo! onoliko koliko mi treba! razmenjujem stringove sa instrumentima, to je sve!

stav proizvođača uređaja?

dva pristupa:

1. otvorena komunikacija, prodam hardware, ponudim software
2. zatvorena komunikacija, prodam hardware, prodam software

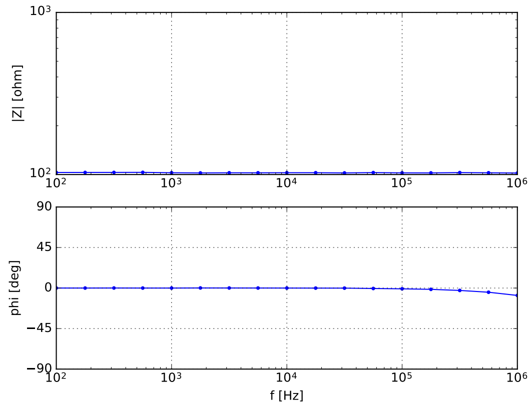
iskustvo od ove godine: zatvorena komunikacija: ne kupujem hardware jer hoću da pišem svoj software!

zanimljivo: isti proizvođač ima različite pristupe, od instrumenta do instrumenta

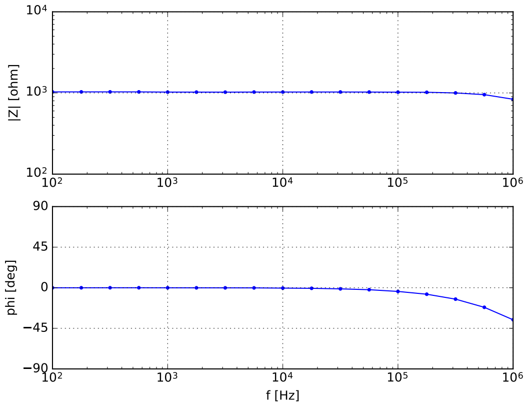
moj stav: SCPI obavezno! Ako je LXI, savršeno!

PRIMERI PRIMENE

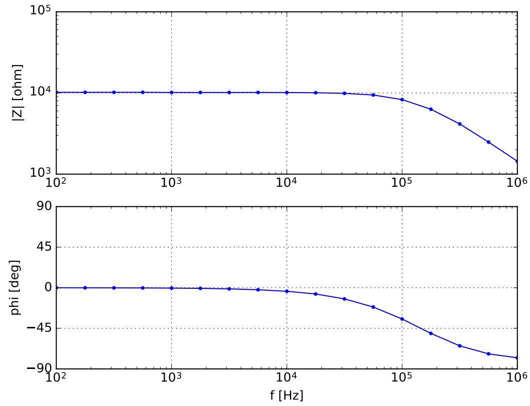
frekvencijska karakteristika, $R = 100\,\Omega$



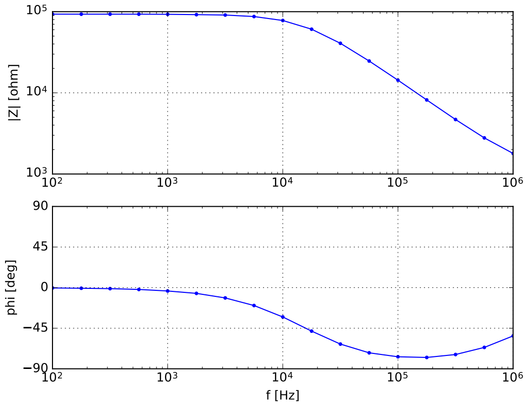
frekvencijska karakteristika, $R = 1\,\mathrm{k}\Omega$



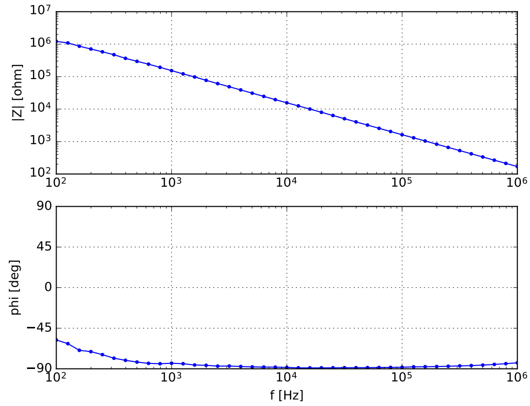
frekvencijska karakteristika, $R = 10\,\mathrm{k}\Omega$



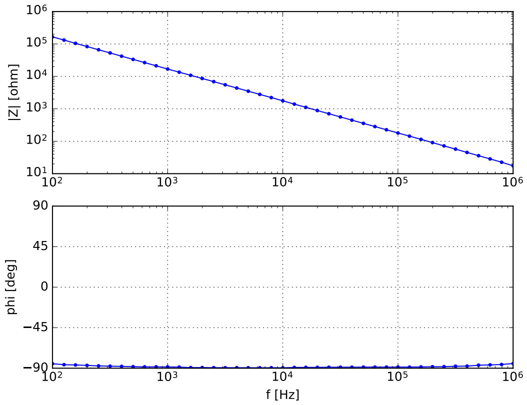
frekvencijska karakteristika, $R = 100\,\mathrm{k}\Omega$



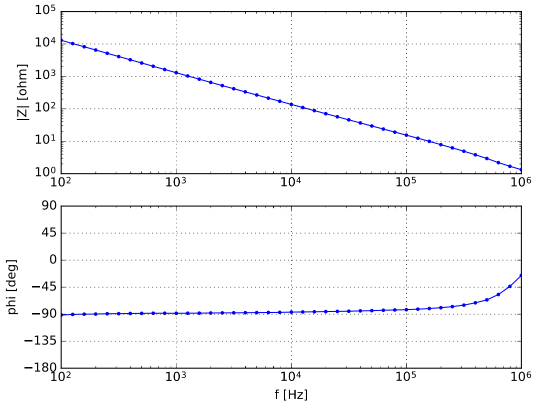
frekvencijska karakteristika, $C = 1\,\mathrm{nF}$



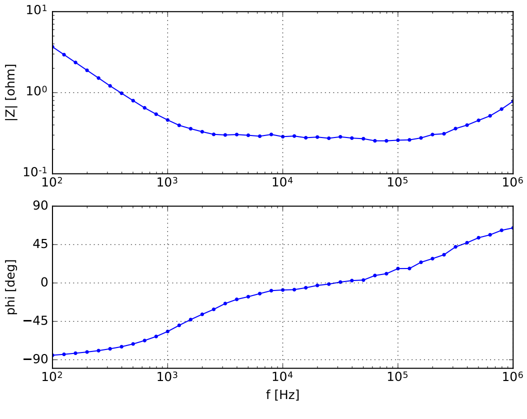
frekvencijska karakteristika, $C = 10\,\mathrm{nF}$



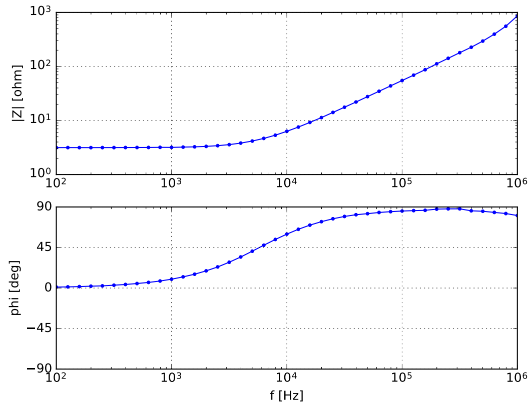
frekvencijska karakteristika, $C = 100\,\mathrm{nF}$



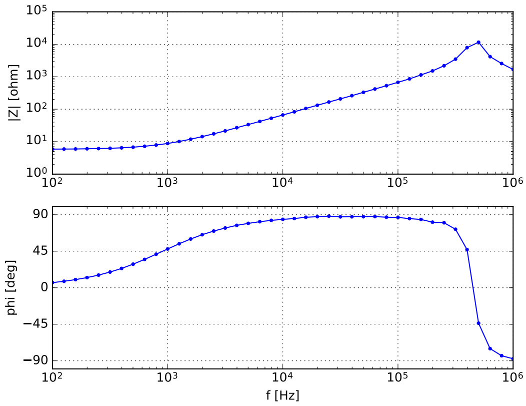
frekvencijska karakteristika, $C = 470\,\mu\mathrm{F}$, elektrolitski



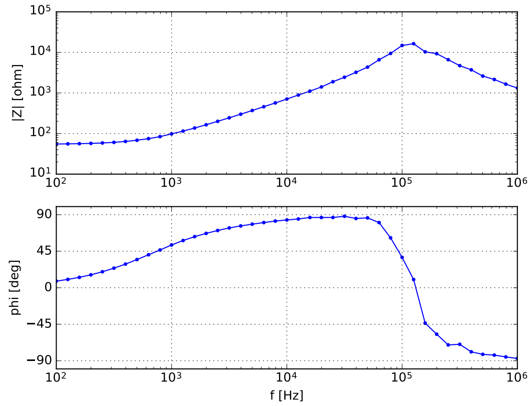
frekvencijska karakteristika, $L = 100\,\mu\mathrm{H}$



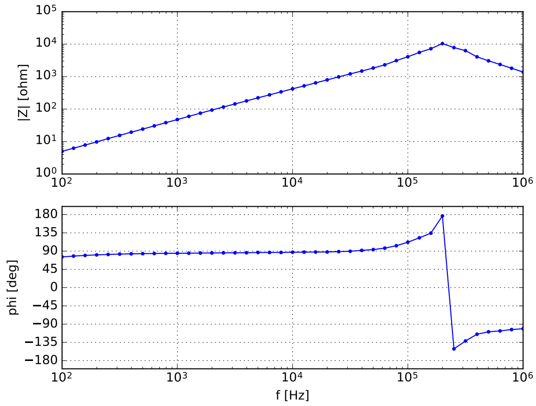
frekvencijska karakteristika, $L = 1\,\mathrm{mH}$



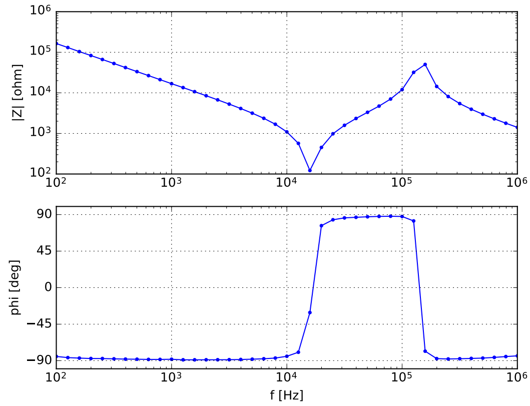
frekvencijska karakteristika, $L = 10\,\mathrm{mH}$



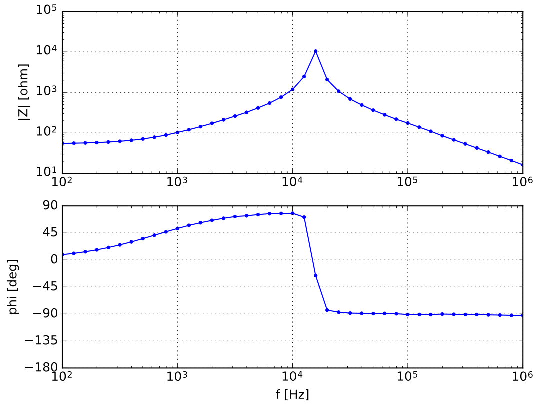
frekvencijska karakteristika, ring core



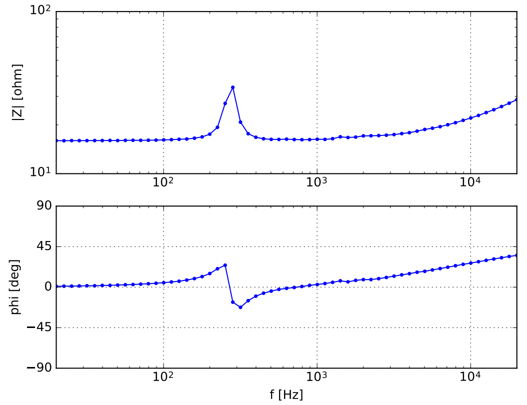
frekvencijska karakteristika, RLC redno



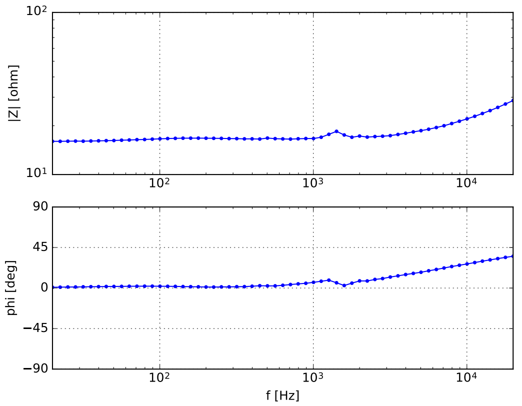
frekvencijska karakteristika, RLC paralelno



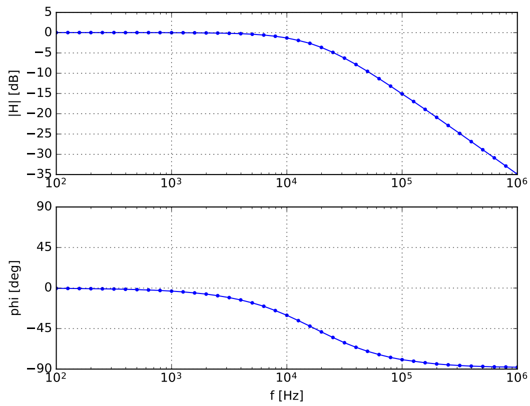
frekvencijska karakteristika, zvučnik, gore



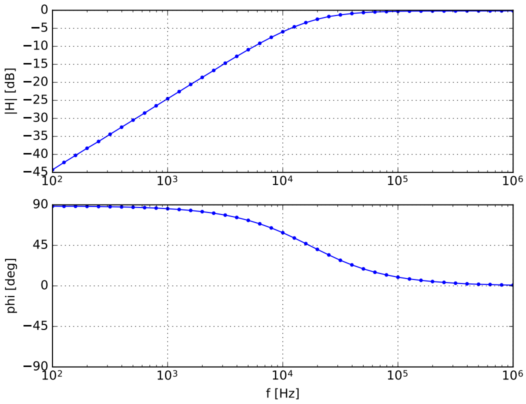
frekvencijska karakteristika, zvučnik, dole



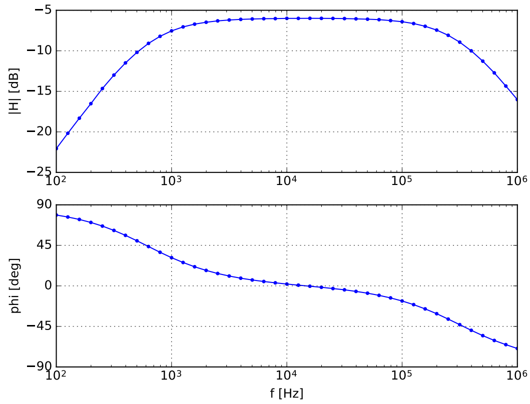
filter propusnik niskih frekvencija



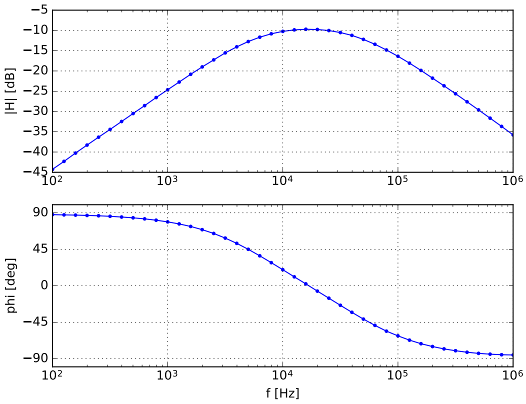
filter propusnik visokih frekvencija



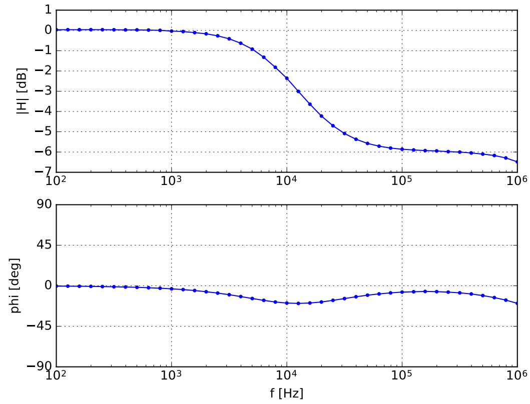
filter propusnik opsega



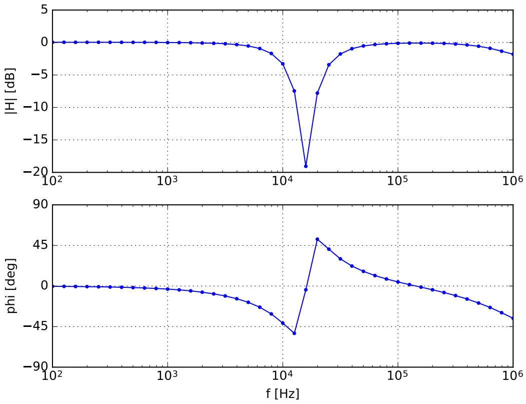
filter propusnik opsega, Wien bridge



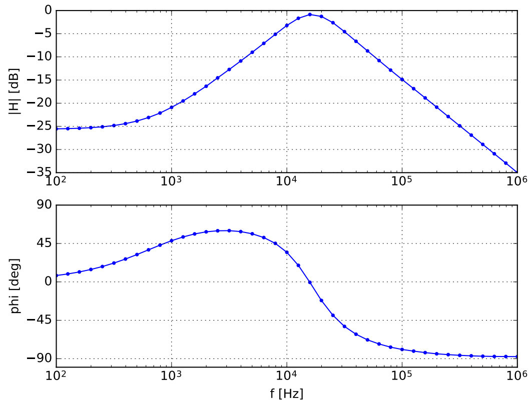
ponašanje kondenzatora . . .



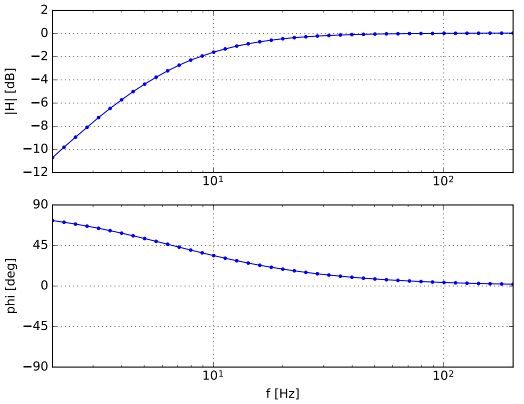
RLC nepropusnik



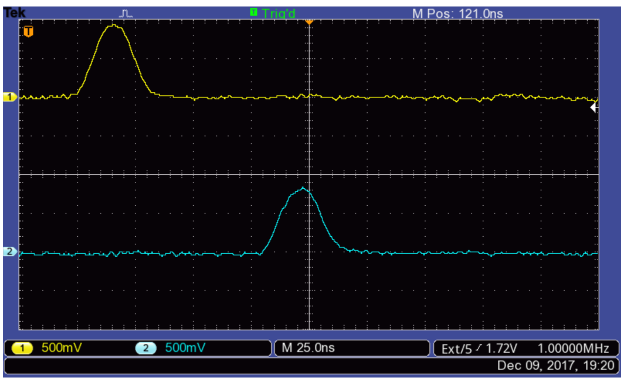
RLC propusnik



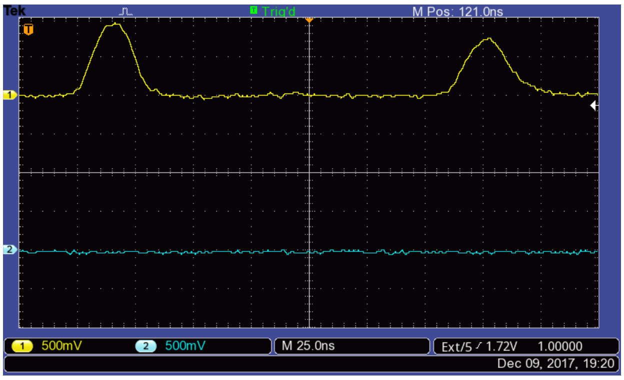
AC/DC filter osciloskopa



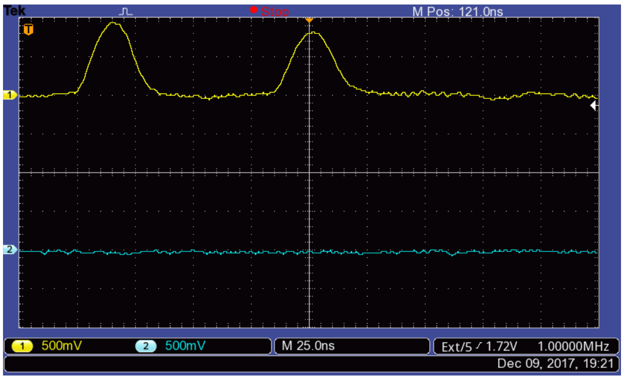
dobar prenos signala po vodu



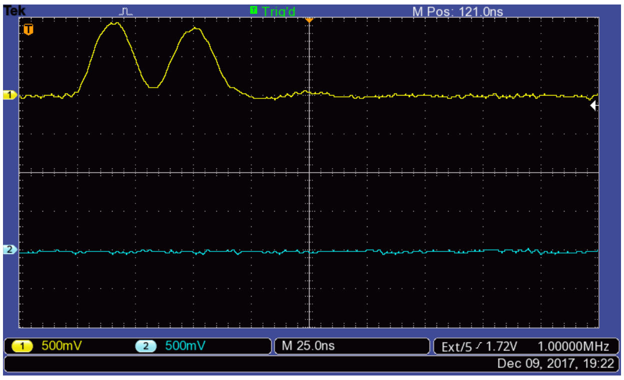
prekid u 3



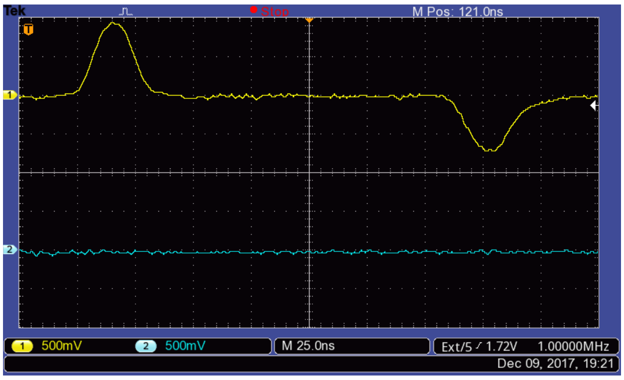
prekid u 2



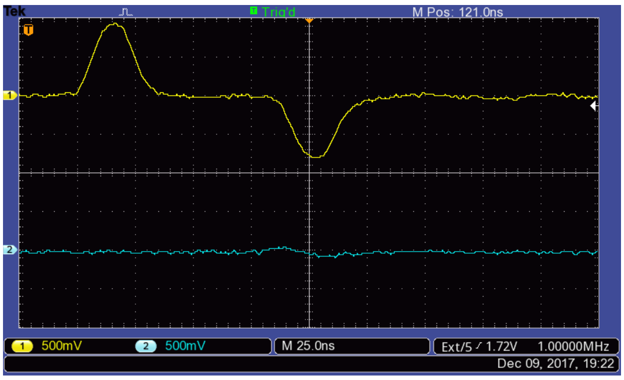
prekid u 1



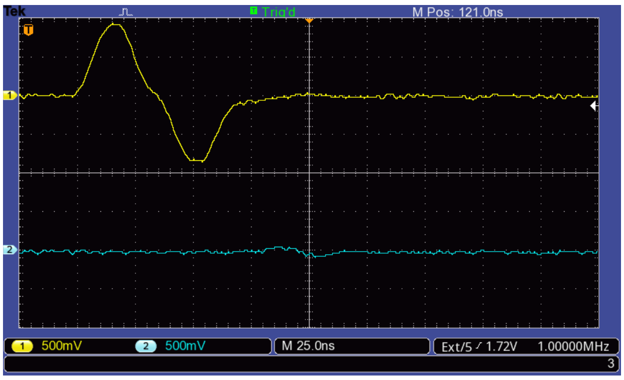
kratko u 3



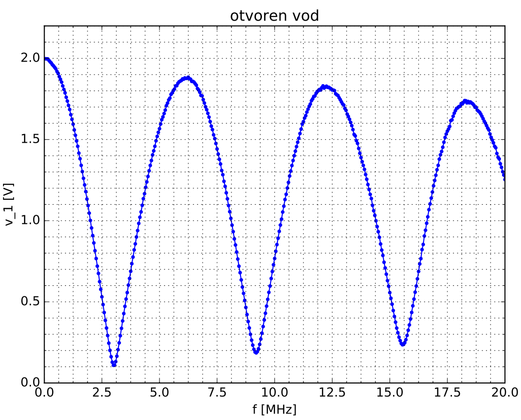
kratko u 2



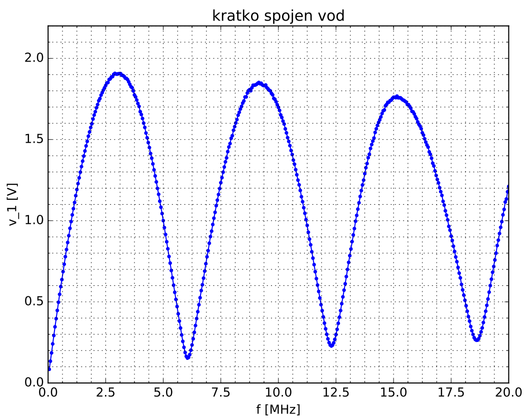
kratko u 1



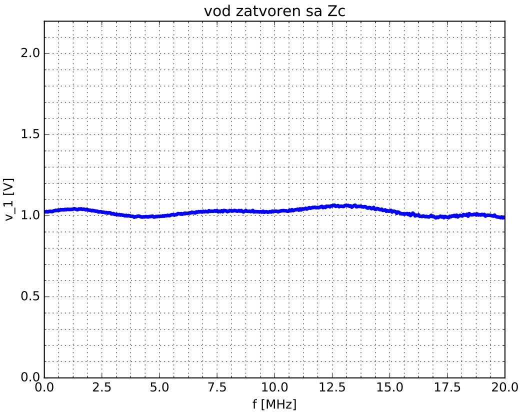
ulazni napon otvorenog voda



ulazni napon kratko spojenog voda

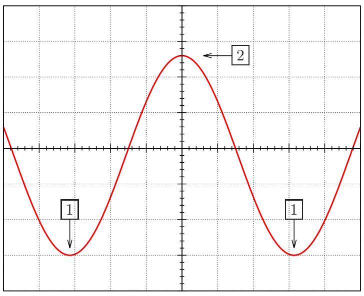


ulazni napon prilagođenog voda



NOV KVALITET?

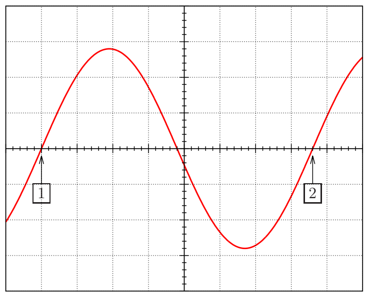
tradicionalno merenje amplitude



samo dva odbirka utiču na rezultat!

kako bi bilo da svi odbirci utiču na rezultat merenja amplitude?

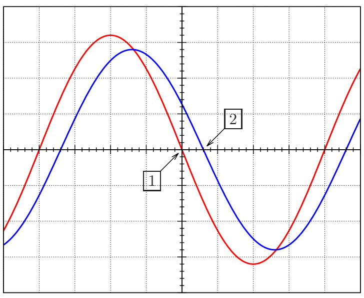
tradicionalno merenje periode



samo dva odbirka utiču na rezultat!

ne treba nam merenje! generator javlja periodu i frekvenciju, mnogo tačnije je zna nego što prosečan osciloskop može da izmeri!

tradicionalno merenje faze



opet samo dva odbirka utiču na rezultat!

kako bi bilo da svi odbirci utiču na rezultat merenja amplitude?

računarski pristup . . .

- svi odbirci, Fourier, ali samo prvi harmonik!
  1. preuzmemo odbirke
  2. vidimo koliko ih stane u periodu
  3. pripremimo težinske funkcije
  4. nađemo sinusnu i kosinusnu komponentu, amplitude
  5. odatle nađemo amplitudu i fazu
  6. a odatle imitansu ili transmitansu!
- ne može ručno, preskupo!

parče koda . . .

```
from pylab import *
from oscusb import *
import usbtmc
import time
from engineeringnotation import *
```

parče koda . . . generator

```
gen = usbtmc.Instrument(0x0957, 0x0407)
gen.write(’output off’)
gen.write(’function sinusoid’)
gen.write(’voltage:offset 0’)
gen.write(’output:load 9.9e37’)
#
def genamp(amplitude):
    gen.write(’voltage:amplitude ’ + str(amplitude))
#
def genfreq(frequency):
    gen.write(’frequency ’ + str(frequency))
#
def waigen():
    gr = ’0’
    while gr != ’1’:
        gr = gen.ask(’*opc?’)
```

parče koda . . . priprema

```
assert 2.0 <= f and 20e6 >= f

findex = where(wholef <= f)[0][-1]
ratio = wholef[findex] / f
ns = int(round(2500 * ratio))
wt = 2.0 * pi * arange(ns) / ns
s = 2 * sin(wt)
c = 2 * cos(wt)

genfreq(f)
genamp(2 * Vm)
waigen()
```

parče koda . . . sampling

```
o.write('horizontal:scale ' + str(timescales[findex]))
o.waituntilready()

o.autorange(1)
o.autorange(2)
o.autorangemath()

o.stop()
v = o.getsamples(1, start = 1, stop = ns)
vg = o.getsamples(2, start = 1, stop = ns)
o.run()
```

parče koda . . . obrada

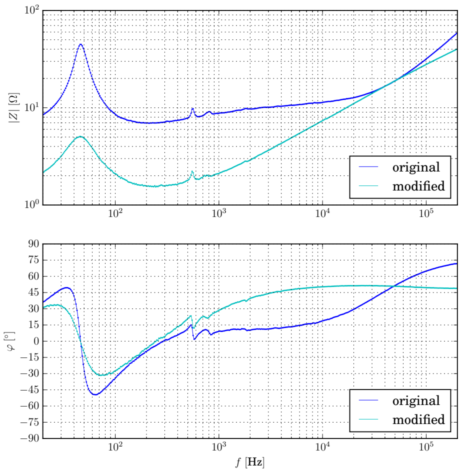
```
i = (vg - v) / R

Vs = mean(v * s)
Vc = mean(v * c)
Vm = sqrt(Vc**2 + Vs**2)
phiv = - arctan2(Vs, Vc)

Is = mean(i * s)
Ic = mean(i * c)
Im = sqrt(Ic**2 + Is**2)
phii = - arctan2(Is, Ic)

Z = Vm / Im
phideg = degrees(phiv - phii)
if phideg < -180:
    phideg += 360
elif phideg > 180:
    phideg -= 360
P = mean(i * v)
```

primena, zvučnik . . .



da obračunamo . . .

- 401 tačka, 401 slika
- oko 5000 odbiraka po slici
- oko 2 miliona merenja!
- oko 4 miliona množenja!
- i bar toliko sabiranja . . .
- još poneki sin, cos, arctan, ali to je sitno . . . sin, cos oko milion puta, arctan oko 800
- ovo bi nekada radili ručno?
- **to je nov kvalitet: metod koji ranije nije mogao biti korišćen!**

PRIKAZI RADOVA NA TEMU PRIMENE PYTHON-A U ATUTOMATIZACIJI MERENJA, MEĐUNARODNE KONFERENCIJE

VIRTUAL INSTRUMENTS FOR POWER ELECTRONICS BASED ON FREE SOFTWARE TOOLS

Introduction

- a story about virtual instruments . . .
- not that we have any other . . .
- surprise: not that we need any other!
- general-purpose instrument and a computer
- generally, acquisition module and a computer
- known
- ideological wars: proprietary versus free
- this paper: **free**
  1. possible to use free software, demonstrated
  2. easy to get required experience
  3. everything is under **your** control

Architecture of the Measurement System

- plant: lab exercise, three-phase rectifiers, MS students
- used in research previously ...
- instruments ... to be listed
- a computer ... not an impressive one
- requirements ... to be listed
- general purpose instruments, specific measurements
- ideal for virtual instrumentation!
- and not that we had any choice
- and not that we needed it!
- as we'll show in this paper ...

what do we have?

1. oscilloscope (Tektronix TDS 1000, TDS 210)
2. RS-232 communication port
3. PC, Ubuntu 12.10 LTS
4. free software (FSF)
5. 100 : 1 voltage probe
6. current probe (Agilent 1146A)

what do we want?

List of items:

1. mean value
2. root-mean-square (*RMS*) value
3. power
4. apparent power
5. power factor (*PF*)
6. displacement power factor (*DPF*)
7. total harmonic distortion (*THD*)
8. efficiency (η)
9. compute waveform spectra

Besides, automatic generation of diagrams, tables, and reports would be fine!

Communication with the Oscilloscope

- **really important!**
  differs VI software from pure data processing
- communication module, RS-232
- PySerial 2.7, http://pyserial.sourceforge.net/, [10]
- communication protocol (language), Tektronix, Programmer Manual TDS 200-Series Digital Real-Time Oscilloscope, [13]
- just sending and receiving strings ...
- Python, http://www.python.org/, [5]
- result:
  http://tnt.etf.rs/~ms1ee2/trofazni-ispravljac-init.zip
  `take.py`
  http://tnt.etf.rs/~ms1ee2/12-pulse-init-2.zip
  `uzmi.py`
- and we have samples! easy and efficient!

Data Processing

- just computation ...
- PyLab provides excellent platform:
  1. NumPy to provide array objects
  2. SciPy library for scientific computing
  3. Matplotlib to provide graphical representation of data
  4. ...
- just implement DSP algorithms ...
- ... if not already implemented
- really simple!

an important issue ...

- $f_0 = 50\,\text{Hz}$
- $T_0 = 20\,\text{ms}$
- $k_t = 2.5\,\text{ms/div}$
- $t_{frame} = 25\,\text{ms}$
- $n_S = 2500$
- synchronize: take 2000 samples (only)
- spectral leakage avoided!
- but we knew the frequency in advance!
- the application happened to be like that
- work in progress ...

computation ...

```
def rms(x):
    return sqrt(mean(x ** 2))

def apower(x, y):
    return mean(x * y)

def thd(x):
    X = abs(fft(x))
    X1rms2 = 2 * X[1] ** 2
    Xrms2 = sum(X ** 2)
    return 100 * sqrt(Xrms2 / X1rms2 - 1)

def dpf(x, y):
    X = fft(x)
    Y = fft(y)
    return cos(angle(X[1]) - angle(Y[1]))
```

achieved ...

1. mean value, `mean` ✓
2. root-mean-square value, `rms` ✓
3. power, `apower` ✓
4. apparent power, $S = V_{RMS} I_{RMS}$ ✓
5. power factor $PF = P/S$ ✓
6. displacement power factor, `dpf` ✓
7. total harmonic distortion, `thd` ✓
8. efficiency, P_{OUT}/P_{IN}, good for verification! ✓
9. compute waveform spectra, `fft` ✓

one frame (slide) of code, and we were done!

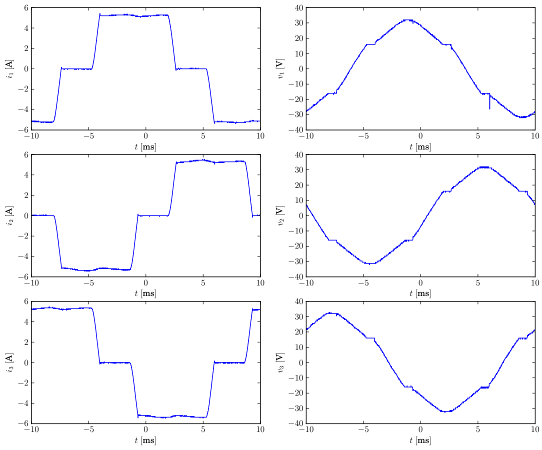
Organization of the Measurements

- measurement protocol, "do this, do that, . . . "
- not really educational . . . or is it?
- useful for standardized measurements . . .
- low qualification requirements . . .
- students might learn, if they wish . . .
- procedure:
  1. connect probes at appropriate places
  2. run appropriate programs to collect data
  3. after the data are collected, run appropriate program to process the data
- we do not have GUI at this time . . .
  (elementary literacy required!)
- neither we believe it's needed!

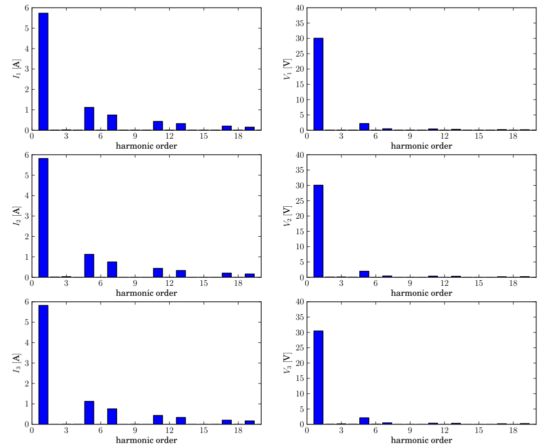
Automatic Report Generation

- really nice feature!
- . . . table data computed, PyLab, NumPy, Scipy
- . . . diagrams created, PyLab, Matplotlib
- all this done automatically; report?
- "copy/paste reports" . . .
  (the second author is an expert in that)
- Python!
  1. prepare LaTeX file automatically
  2. run `pdflatex` automatically
- and we have the report!
- no copy, no paste, abstraction level increased!
- no "lack of concentration" errors!

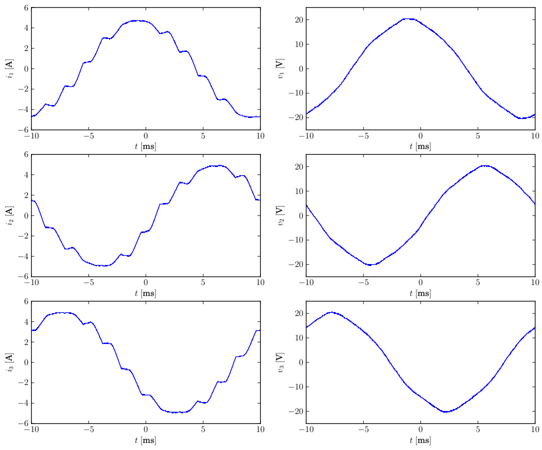
what it looks like?



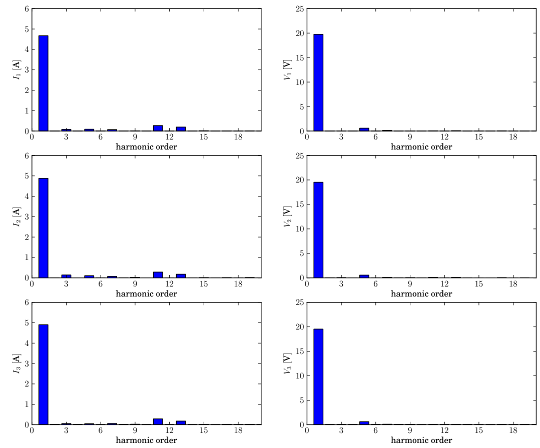
what it looks like?



what it looks like?



what it looks like?



reports?

- 49-page report:
  http://tnt.etf.rs/~ms1ee2/report.pdf
- 92-page report:
  http://tnt.etf.rs/~ms1ee2/report-12-pulse-2.pdf
- in Serbian, but not much of the language there . . .
- tables, diagrams . . .
- ideal for repeated measurements, certified laboratories
- **educational:** students **have** the code, GPLv3, free to reuse in their own systems!

software?

- available as free software, FSF, GPLv3
- six-pulse:
  http://tnt.etf.rs/~ms1ee2/trofazni-ispravljac-init.zip
- twelve-pulse:
  http://tnt.etf.rs/~ms1ee2/12-pulse-init-2.zip
- doing everything:
  1. data acquisition
  2. data processing
  3. automatic report generation
- we made it: take it, use it, it's free!

Conclusions

- virtual instruments for electric power and power electronics
- 50 Hz measurements
- everything achieved using free software exclusively!
  1. data acquisition
  2. data processing
  3. automatic report generation
- all the developed software is (GPLv3) free!
- implemented for lab exercises
- easy to modify for other applications
- educational points:
  1. virtual instruments (acquisition, processing, reporting)
  2. organization of a measurement system
  3. implementation in free software
- it **is** possible to create virtual instruments using only free software!

A SYSTEM FOR MEASURING MAINS VOLTAGE PARAMETERS AND LOGGING THE DATA

Introduction, #1

Requirements:

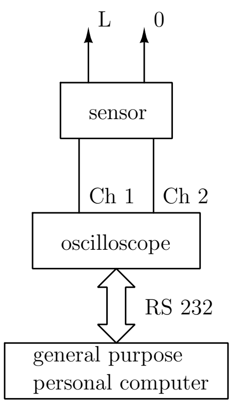
- measure THD and RMS of the mains voltage
- log the data over long time
- do not use specialized equipment
- use free software
- budget $\approx$ 0 US$

Introduction, #2

Choices:

- home made voltage sensor
- a digital oscilloscope we already had, Tektronix TDS 210
- RS-232 communication
- personal computer to operate the system and store the data, Ubuntu GNU/Linux
- software ingredients:
  1. Python Programming Language
  2. time — Time access and conversions
  3. pySerial
  4. NumPy
  5. matplotlib

The System

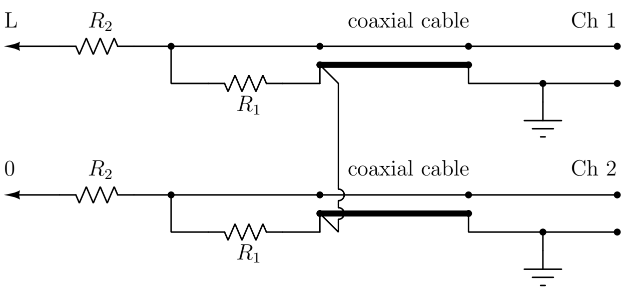


The Sensor, #1

Requirements:

- differential measurements
- symmetrical sensor, not sensitive on which wire is hot
- reduce interference as much as possible
- cheap!

The Sensor, #2



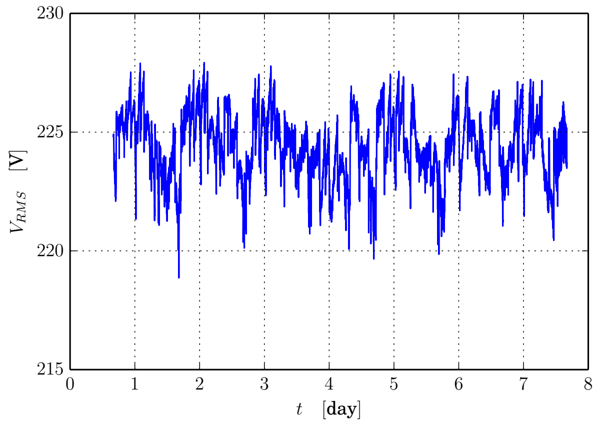
The Sensor, #3

The Only Equation

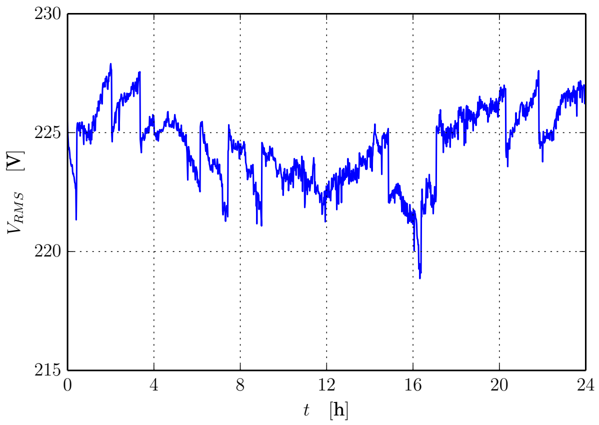
$$A = \frac{R_1}{R_1 + R_2}$$

... assuming that $R_1 = R_1$ and $R_2 = R_2$ ...

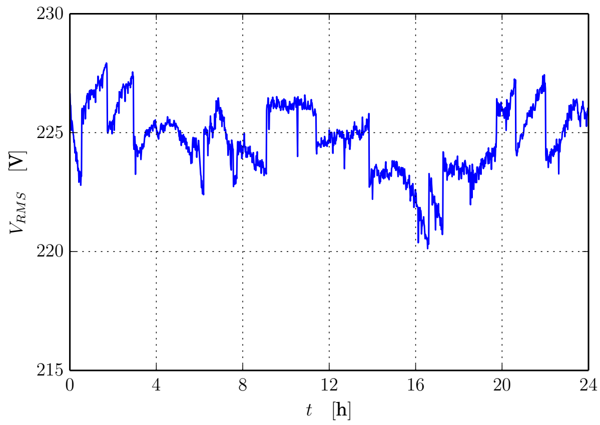
RMS of the voltage, a week



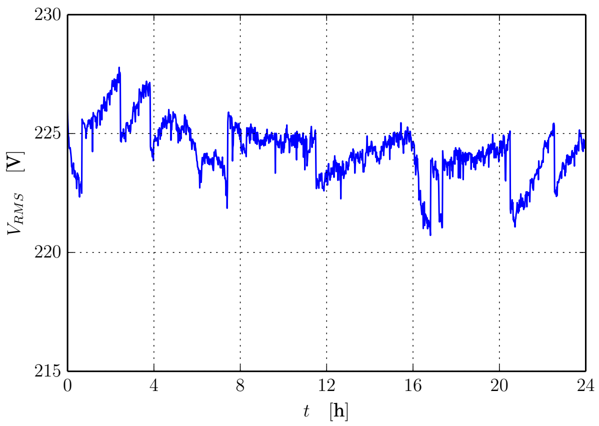
RMS of the voltage, a day



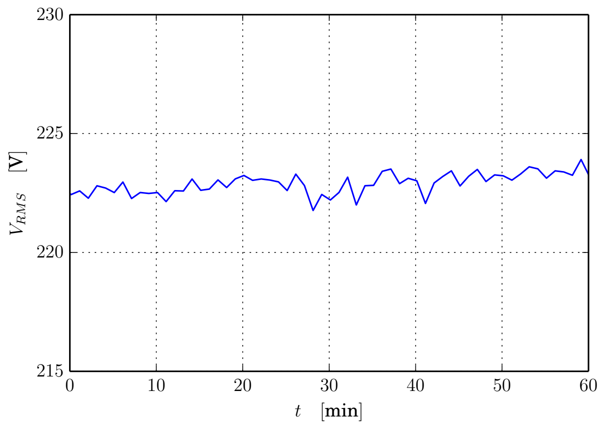
RMS of the voltage, another day



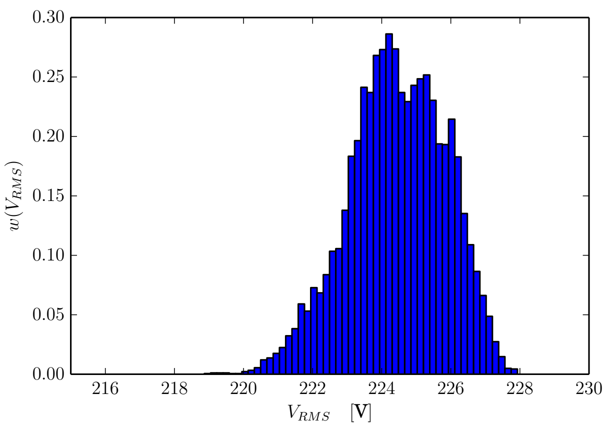
RMS of the voltage, yet another day



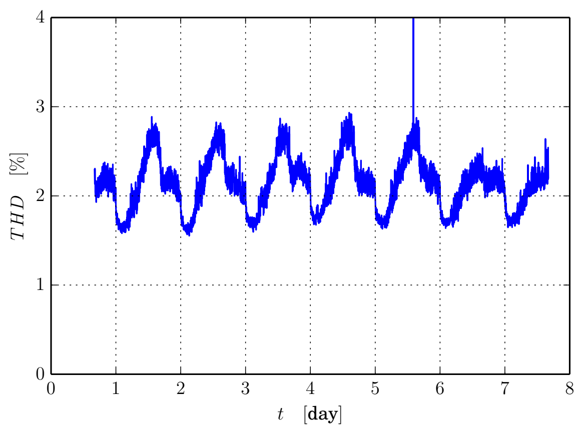
RMS of the voltage, an hour



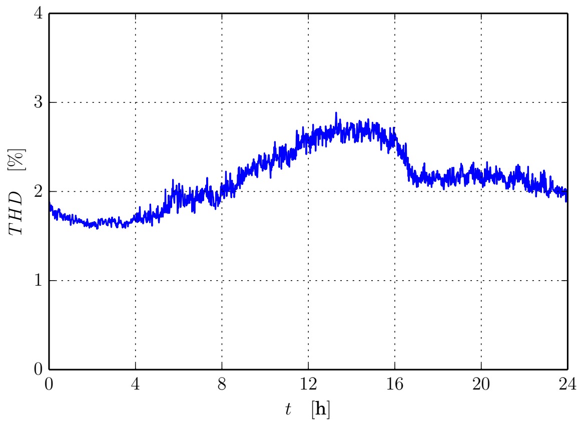
RMS of the voltage, histogram ...



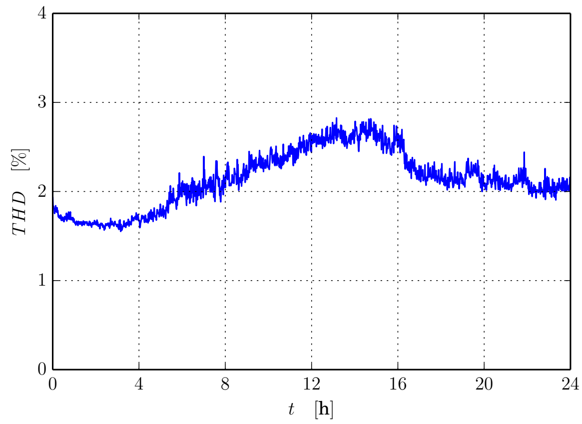
THD of the voltage, a week



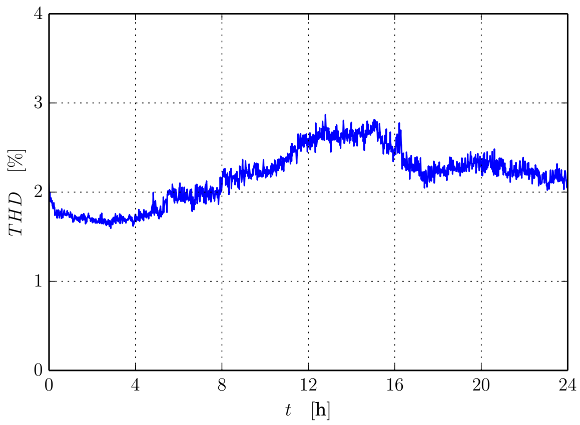
THD of the voltage, Monday



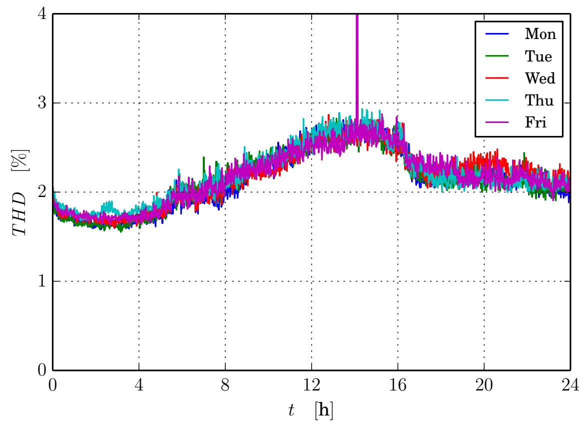
THD of the voltage, Tuesday



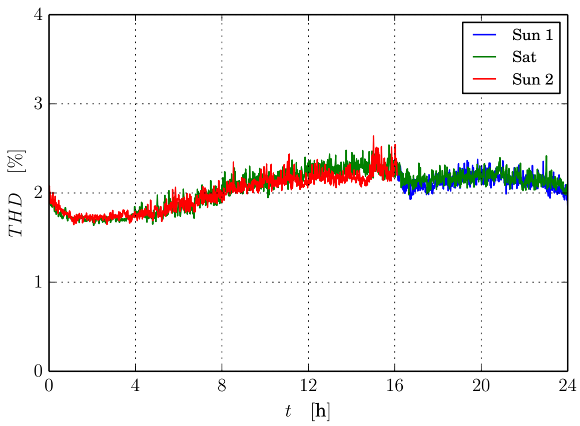
THD of the voltage, Wednesday



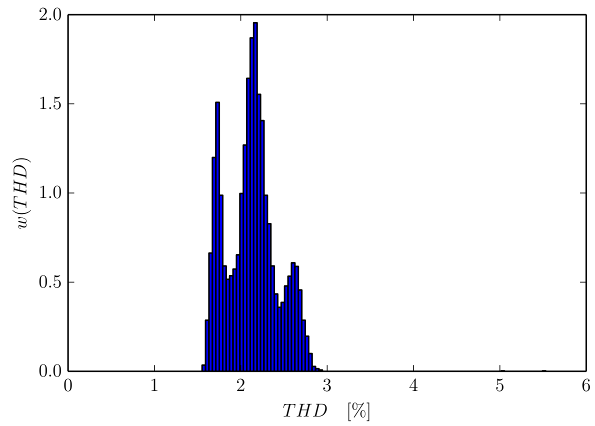
THD of the voltage, working days



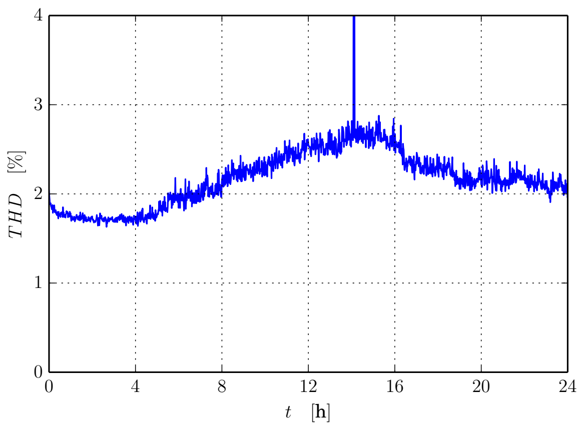
THD of the voltage, weekend



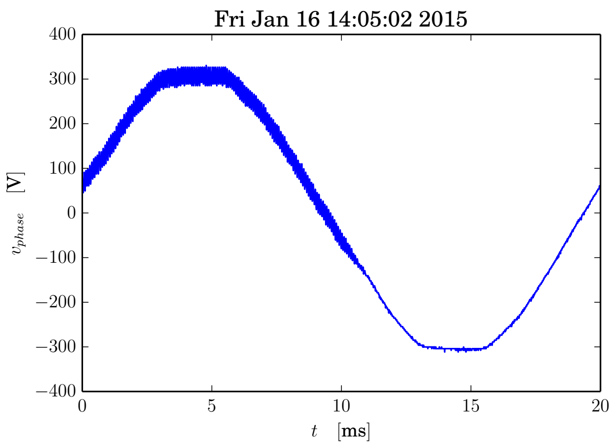
THD of the voltage, histogram



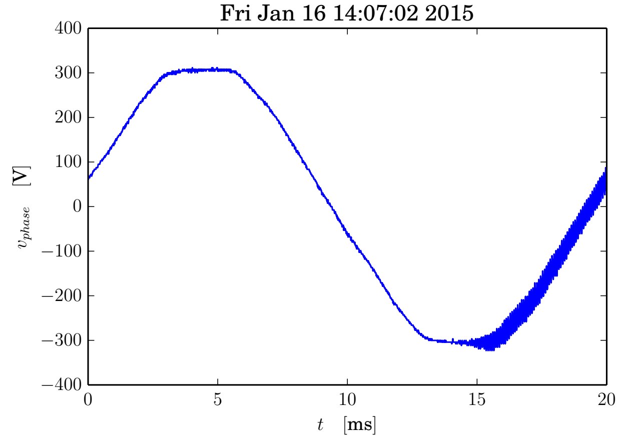
THD of the voltage, a very specific Friday



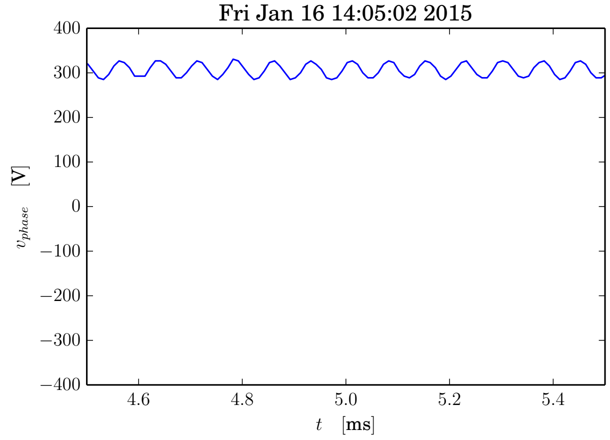
Waveforms, that Friday, case #1



Waveforms, that Friday, case #2



Waveform, case #1, magnified



What happened on Friday?

Interference:

- 20 V in amplitude
- frequency of 14 kHz
- origin?
- observed?
- "observed?"

Conclusions, system

- a system for phase voltage monitoring and recording of its parameters:
  1. RMS
  2. THD
- automated measurement system is programmed in Python
- relying entirely on free software modules for support
- upgrade of the system to previous publications
- time module of the Python Standard Library

Conclusions, measurements

- illustration of the system application, mains voltage monitored over a week
- sudden changes in RMS, in between smooth variation intervals
- some periodicity in RMS
- RMS histogram
- strong periodicity in THD, demonstrated
- THD histogram
- THD, isolated events, waveforms shown, cause?

Conclusions of the Conclusions

- simple
- cheap
- versatile
- free

A SIMPLE SYSTEM TO ESTIMATE ON-SITE SOLAR ENERGY HARVESTING

Introduction, #1

Requirements:

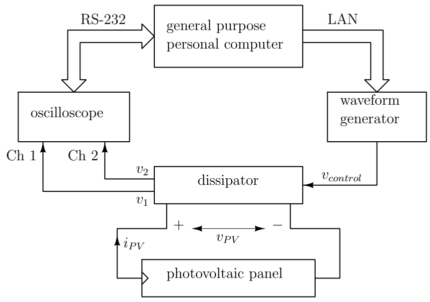
- estimate solar energy available on a site
- provide maximum power point tracking
- perform accounting of the harvested energy
- log the data
- minimize the use of specialized equipment
- do not care about the harvested energy

Introduction, #2

Choices:

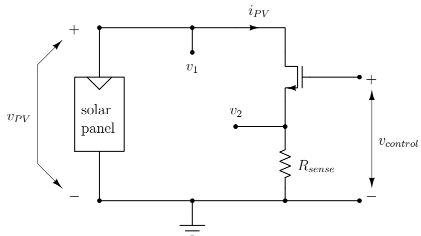
- home made controllable dissipator
- digital oscilloscope Tektronix TDS 210
- signal generator Agilent 33220A
- personal computer to operate the system and store the data, Ubuntu GNU/Linux
- software ingredients:
  1. Python Programming Language
  2. time — Time access and conversions
  3. pySerial
  4. NumPy
  5. matplotlib

The System

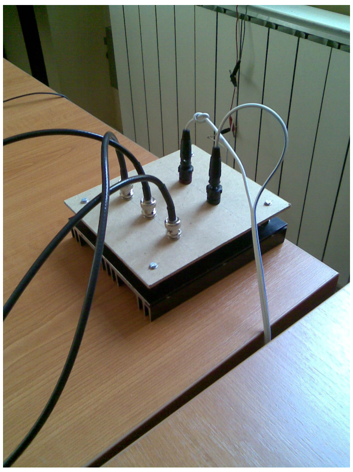


The System

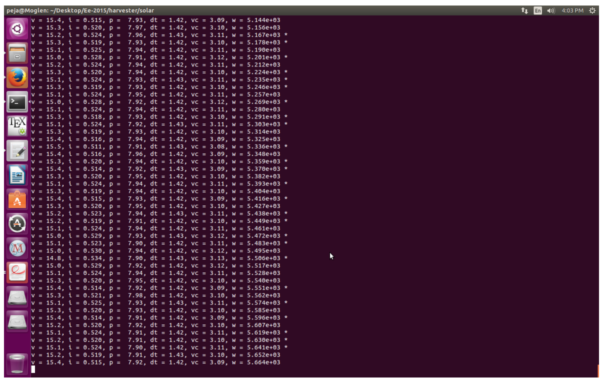
The Dissipator



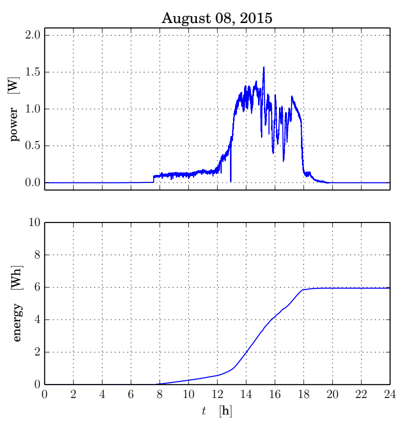
The Dissipator



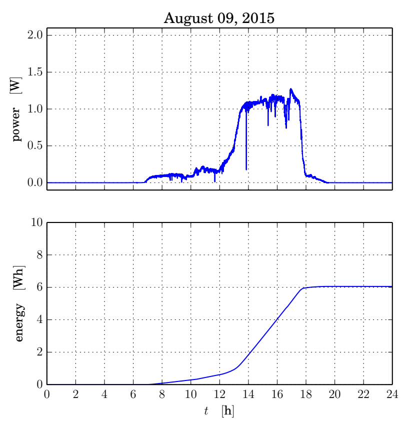
Screenshot



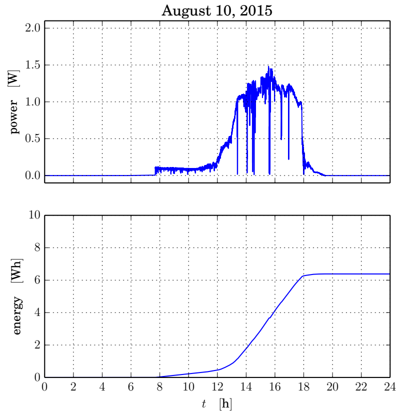
2015-08-08



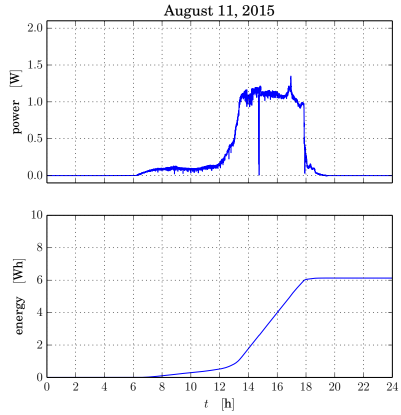
2015-08-09



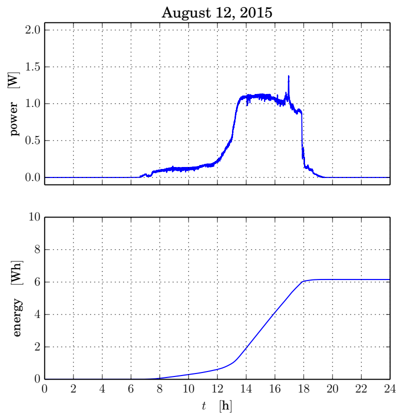
2015-08-10



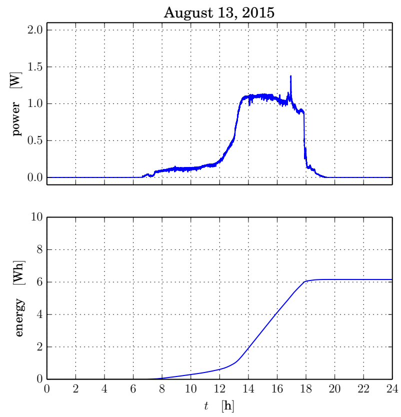
2015-08-11



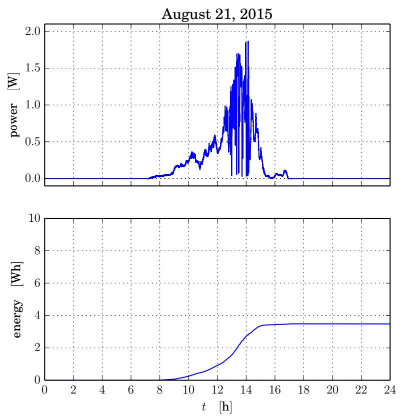
2015-08-12



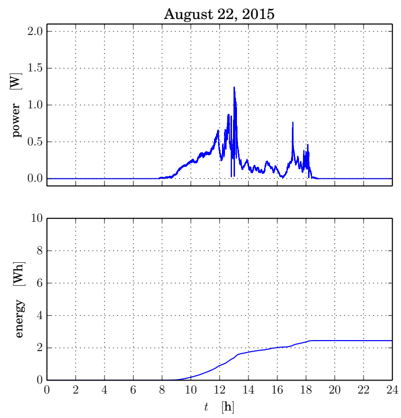
2015-08-13



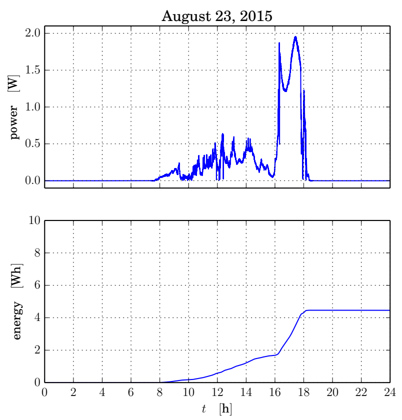
2015-08-21



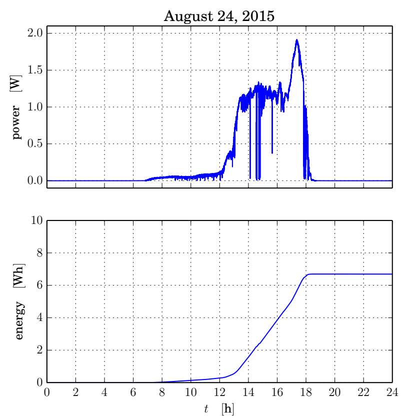
2015-08-22



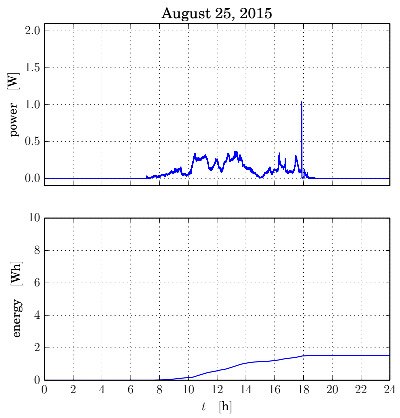
2015-08-23



2015-08-24



2015-08-25



Conclusions

- a system to estimate the energy that might be harvested at a specific location
- standard laboratory equipment, Tektronix TDS 210 and Agilent 33220A
- controllable dissipator to load the solar panel
- automated instrumentation, PC with Ubuntu GNU/Linux
- provide MPPT, control the dissipator
- measure power and do accounting for the energy
- store the data
- entirely based on free software components and tools
- Python and appropriate modules
- experimental results are presented
- **complete success!**
- about 6 Wh on a sunny day

MAXIMUM POWER POINT TRACKING CONTROL SYSTEM OF PHOTOVOLTAIC MODULE USING FREE SOFTWARE AND STANDARD LABORATORY EQUIPMENT

Introduction, #1

Requirements:

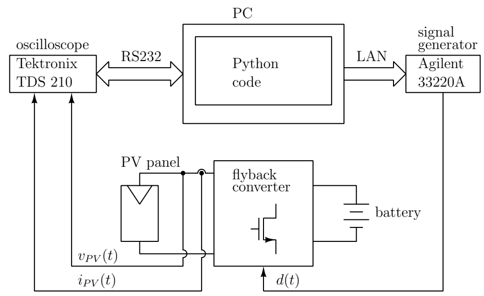
- supply a battery from a photovoltaic panel
- provide maximum power point tracking
- test various algorithms
- rapid prototyping
- limited resources and budget
- rely on instruments we already have
- rely on free software as much as possible
- converter with galvanic isolation

Introduction, #2

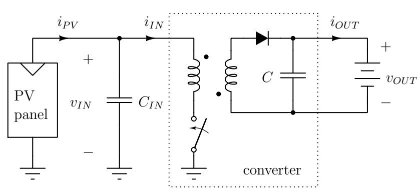
Choices:

- oscilloscope to measure photovoltaic panel voltage and current, Tektronix TDS 210
- RS-232 communication
- signal generator Agilent 3320A to provide $d(t)$
- communication using VXI-11 over TCP/IP interface
- personal computer to operate the system and store the data, Ubuntu GNU/Linux
- software ingredients:
  1. Python Programming Language
  2. time — Time access and conversions
  3. pySerial
  4. python-vxi11
  5. NumPy
  6. matplotlib

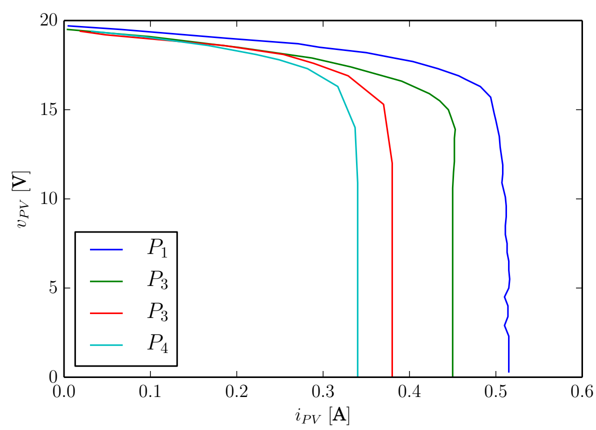
The System



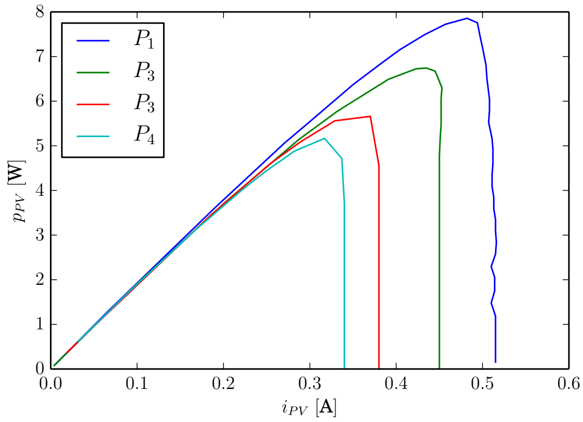
The Converter



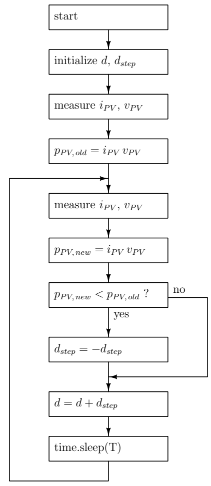
Photovoltaic Panel, $v_{PV}(i_{PV})$



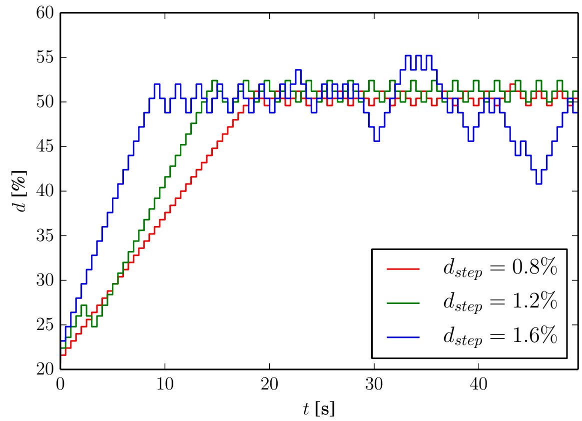
Photovoltaic Panel, $p_{PV}\,(i_{PV})$



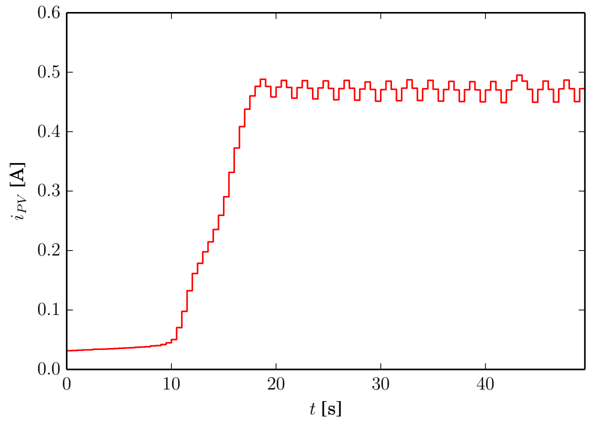
The Algorithm



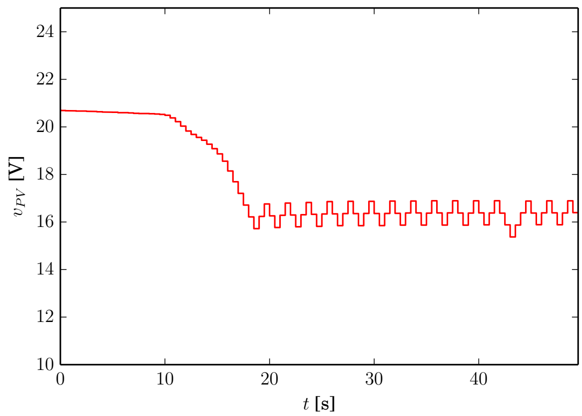
Startup, $d\,(t)$



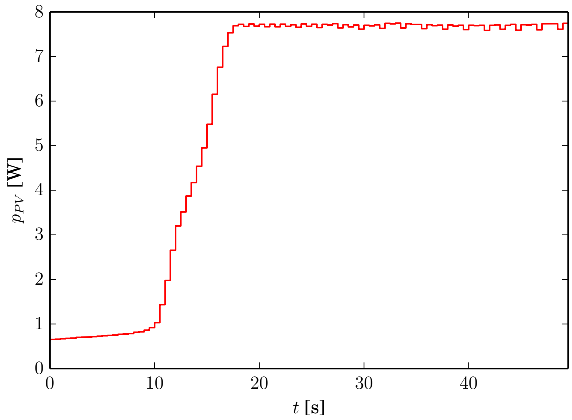
Startup, $i_{PV}\,(t)$



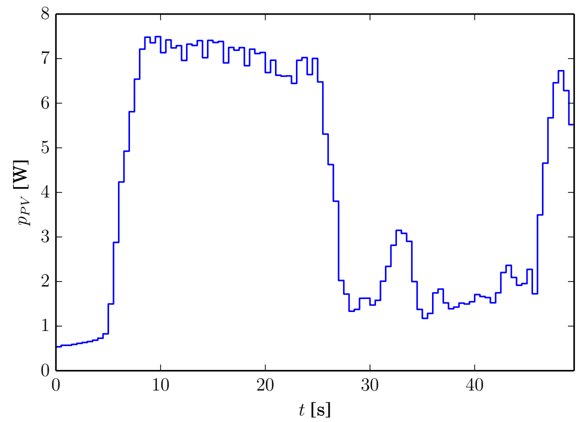
Startup, $v_{PV}\,(t)$



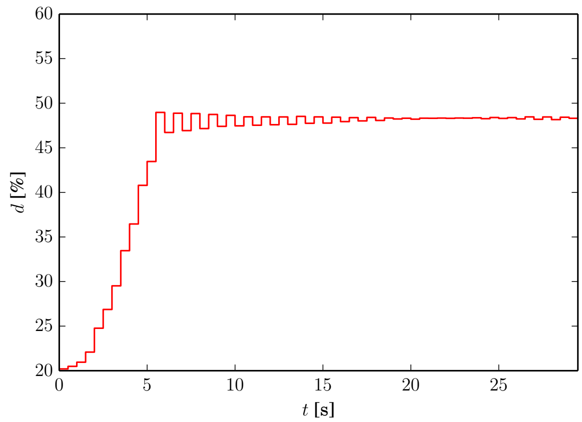
Startup, $p_{PV}\,(t)$



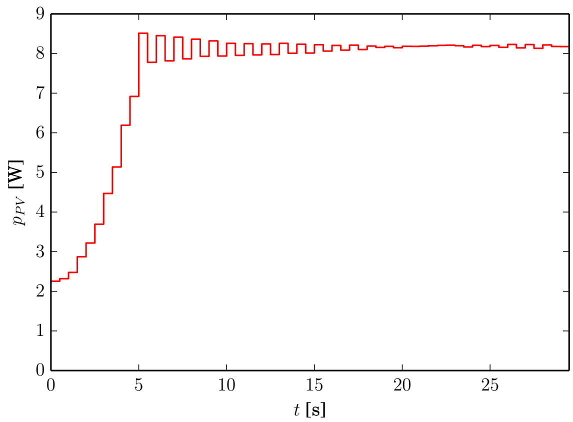
$i_{PV}\,(t)$, with unintentional shading



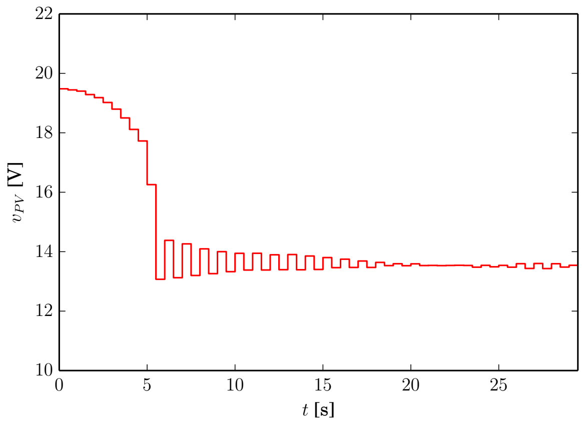
$d\,(t)$, adjustable d_{step}



$p_{PV}(t)$, adaptive algorithm



$v_{PV}(t)$, adaptive algorithm



Conclusions, #1

- experimental environment for rapid prototyping of MPPT algorithms
- standard laboratory equippment, oscilloscope and a signal generator
- networked instruments, connection via RS-232 and TCP-IP
- standard PC to provide networking and data processing
- free software, Python and some modules:
  1. pySeiral
  2. python-vxi11
  3. NumPy
  4. time - Time access and conversions
  5. matplotlib

Conclusions, #2

- two algorithms sucesfully implemented
- both "perturb and observe hill climbing algorithm"
- Diploma Thesis of the first author, Vladan Lazarević, who is in Madrid now, doing his Ph.D.
- **complete success!**

SOFTWARE SUPPORTED DC VOLTAGE CALIBRATOR

Introduction

What is this all about?

- we needed a DC voltage calibrator . . .
- which is expensive . . .
- . . . and will never pay out for . . .
- and we needed it urgently . . .
- . . . meaning that it would arrive too late.
- we had:
  1. controllable signal generator
  2. controllable (*LXI*) voltmeter
  3. **free** software
- and we made it!
- and we would like to share the experience with you
- for free, since we got the underlying software for free

Introduction

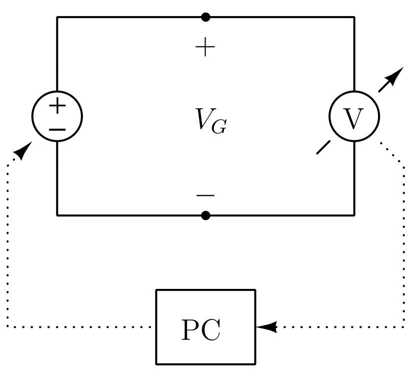
What did we actually have?

- Agilent 33220A signal generator (TCP/IP supported, SCPI commands)
- Agilent 34411A digital multimeter (TCP/IP supported, *LXI* label, SCPI commands)
- PC with Ubuntu 16.04, Python 2.7.12, Python VXI-11 module, PyLab (NumPy, SciPy, matplotlib, IPython)
- later on we included Tkinter to make GUI (not in the paper)

What did we obtain?

- voltage comparator accurate to $\approx 0.5\,\mathrm{mV}$ in $-10\,\mathrm{V} \leq v_{OUT} \leq 10\,\mathrm{V}$ range, which is about 50 ppm
- out of components we already had . . .
- only by adding some (free) software!

an idea . . . (final) . . .



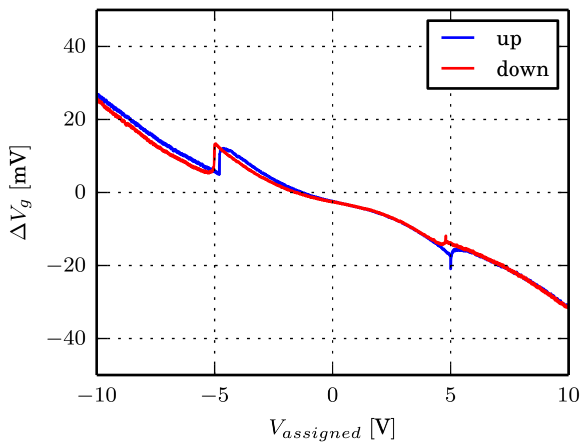
measure the voltage, compensate for the error **somehow** . . .

. . . easy to build . . .

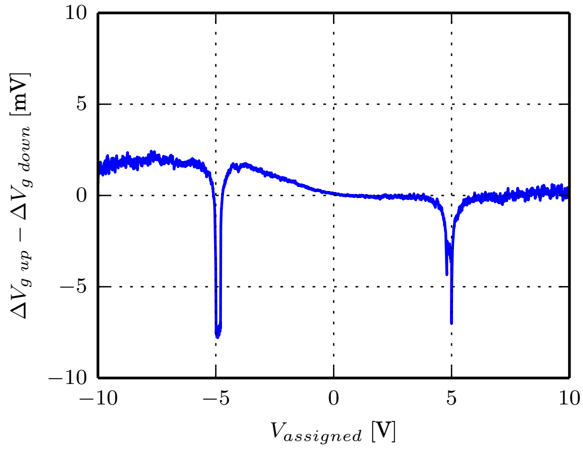
let's check the generator . . .

- transfer curve $V_g\,(V_{assigned})$
- forward and backward!
- long term stability
- for a set of different output voltages
- required an auxiliary automated system to measure . . .
- . . . to make conclusions and decisions . . .
- included 24 hour measurements!
- cannot be done without an automated system!
- . . . an auxiliary automated system

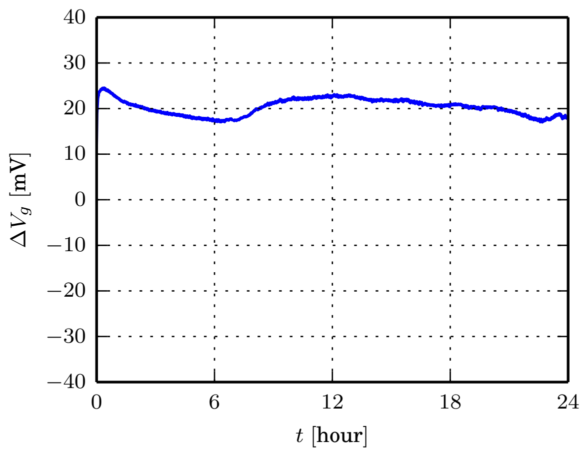
. . . forward and backward . . .



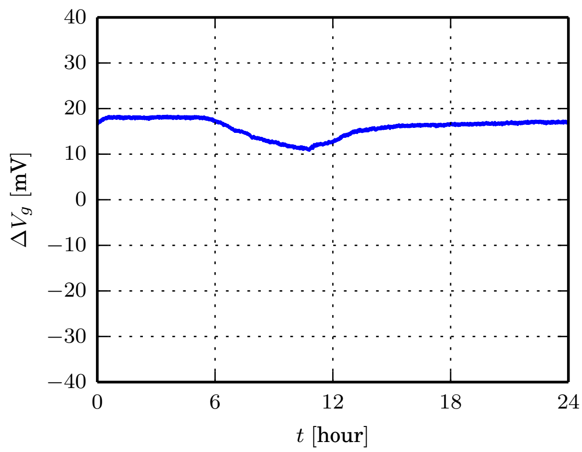
. . . and there is hysteresis . . .



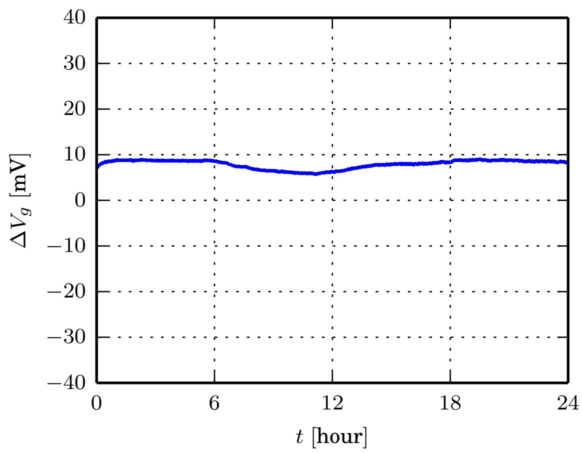
. . . and long term instability . . . (−9 V)



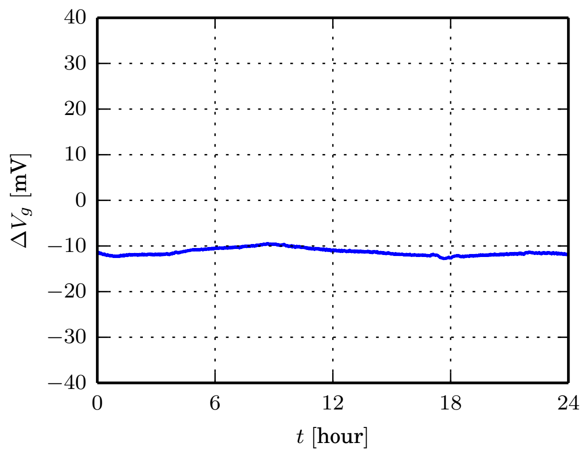
. . . and long term instability . . . (−7.5 V)



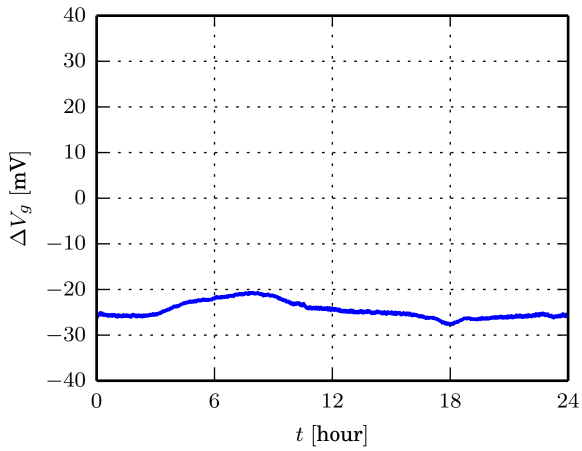
. . . and long term instability . . . (−3.5 V)



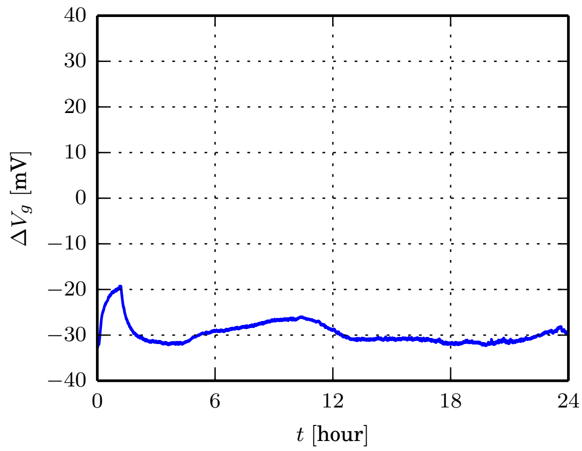
. . . and long term instability . . . (3.5 V)



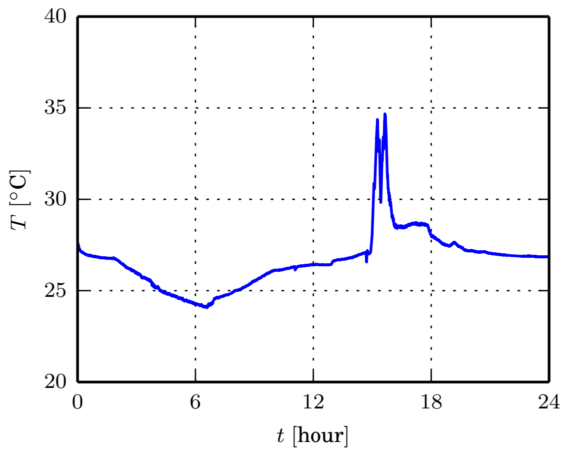
… and long term instability … (7.5 V)



… and long term instability … (9 V)



… caused by temperature variations …



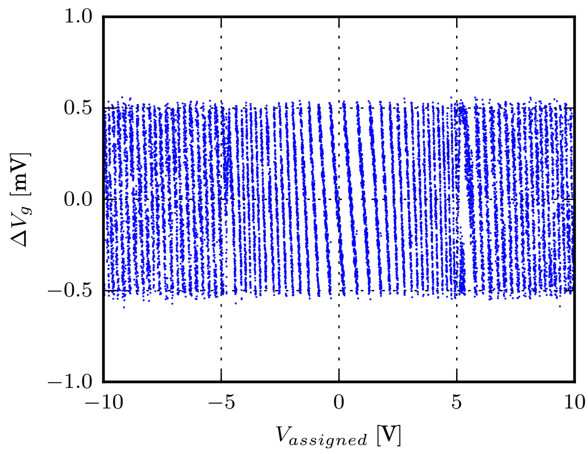
Important local conclusions: closed loop

- error within ±40 mV
- which is ±4000 ppm
- which is too much!
- unstable within 10 mV band, ±500 ppm!
- hysteresis!
- no way to compensate to ±50 ppm by a lookup table …
- neither with an interpolating polynomial …
- the only solution is a **closed loop** compensation!
- which means:
  1. measure
  2. adjust the assigned value for one quantum
     (there are other options, though, to be published)
  3. assign the new value
  4. repeat in a loop
- **no way to fix the problem in an open loop!**

… a part of the code … run in a thread!

```
vreal = vread()
eps = (vreal - vold) * 1e3
if eps < - 0.5 and vassigned < 10.0:
    vassigned += 1e-3
    setgen(vassigned)
elif eps > 0.5 and vassigned > - 10.0:
    vassigned -= 1e-3
    setgen(vassigned)
else:
    pass
```

Verification of the Calibrator



a word about the previous diagram …

- takes time to load …
- … since it has been taken in **20001 points** and plotted in 2001 points!
- could you do that without automation?
- another Luddite movement?
- the error within ±56 ppm for $-9.95\,V \le V_g \le 9.95\,V$!
- actually, the error is between +0.5311 mV and −0.5522 mV
- remember: we started with ±4000 ppm, **improved 70 times!**
- theoretical minimum is ±50 ppm (i.e. 0.5 mV), we are about 10% worse than that
- saturation effects, at ends the error grows to about ±32 mV, the range has to be limited

GUI, and how to put us down … down?

- GUI?
- do we need a GUI?
- marketing …
- operators …
- so we made the GUI!
- a topic for the papers to come …
- nice and easy …
- Tkinter
- under Ubuntu
- and under Linux Mint
- took one morning to create from zero …
- … and could be applied in other projects …
- … topic for the papers to come!

Conclusions

- DC voltage calibrator
- requires signal generator and a voltmeter
- both instruments of the "*LXI*" type ...
- and some free software ...
- error reduced about 70 times, almost to orders of magnitude!
- new development: includes GUI!
- not that we believe it is necessary
- though it looks nice, and will be published in future ...
- and used in other projects, in different areas ...
- thanks for complaining!
- code provided in the paper
- take it, use it, enjoy it, it's free!

ELECTRICAL MEASUREMENTS REVISITED — EXPERIENCES FROM MODERNIZING THE COURSE

Introduction

- to make an honest presentation, or a politically correct one?
- what the presentations are for?
- to say something which is **not** in the paper!
- otherwise the paper would be enough!
- so, an honest one!
- Electrical Measurements, sophomore class ...
- shared between at least three departments ...
- different teachers, different students, different **interests**, different views, lots of compromising ...
- result: patchwork shared among teachers that disagree
- the only common point: a lab with obsolete equipment
- an administrative opportunity no one thought about: to split the class into groups, customize as needed
- let's use it!

Introduction

- I'm in charge for the Department of Electronics
- motivated students, capable, with interest in hardware, making, creating, measuring, verifying ...
- I believe in the use of computers!
- and I believe in free software ...
- which I use in my everyday practice for everything ...
- including automated measurement systems ...
- so I could teach my students ("the kids") like they are my own kids, the best I can, topics I believe in
- to be capable, responsible, and independent
- so, let's start

Basic Principles

1. **real world principle**: to teach measurements on the equipment used in everyday practice, in the way they are performed in everyday practice
2. **up to date principle**: to use computers in the way they are used (or should be used) in everyday practice; forget about "a discipline that requires tedious work"; don't calculate by hand in the 21-st century, just to train muscles, not brain
3. **integrating principle**: students tend to treat courses as separate entities, pass the exam and forget approach; measurements are useful, measurements are needed; integrate the knowledge with other courses! support each other; illustrate the theory in practice!
4. **support excellence principle**: don't focus to average, or even worse, below average students; provide enough material to motivate the best students to go further, to improve! require only fundamental skills and topics to pass; encourage students to learn more if they are able and motivated!

Equipment

six benches, twelve students simultaneously in the lab, equippment:

1. Tektronix TDS 1002 + communication port
2. Agilent 33220A signal generator
3. Agilent E3630A triple output DC power source
4. an obsolete computer, Linux Mint MATE
5. two Fluke 111 multimeters
6. one DT-838 or RTO-1035N multimeter
7. protoboard
8. a set of elementary electronic components

mostly thanks to Tempus JEP 17028-02 project

Standard Workbench

Lab Exercises

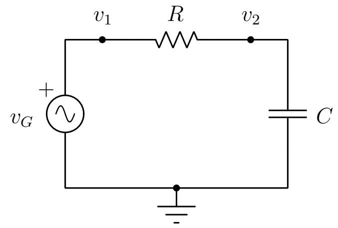
- 12 students simultaneously in the lab
- whenever possible, doing the same exercise
- two supervisors, competent
- already acquired knowledge: measurement of current, voltage, and resistance
- prerequisites are a sort of problem ...
- but with good students, not a real problem
- goal to illustrate concepts learned in Fundamentals of Electrical Engineering and Electric Circuit Theory
- good personal cooperation
- in parallel with Software Tools in Electronics course ...
- which I teach
- so the teachers agree!
- idea: to gain benefit for all courses involved!

Lab 1: Multimeter, DC Power Source, Protoboard

- not a big deal of measurement?
- well, just the opposite!
- practice, learn and remember, we won't get back!
- three voltmeters measuring the same voltage?
- yup! however, the readings are different!
- amazingly, the students were amazed!
- correlate the measurements, linear least squares, practice, Python Linux Mint MATE, look and feel similar to what students were familiar with
- measure resistance, protoboard, measure current and voltage, fith a line through the readings, linear least squares once again
- learn and **remember**! will be used later on!

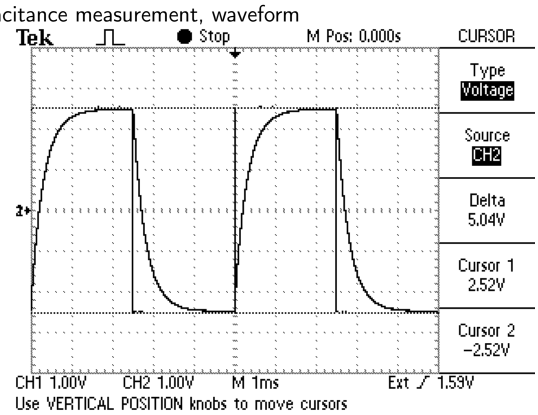
Lab 2: Oscilloscope and Signal Generator

- signal generator settings, manual: waveform type, voltage levels, period, frequency, duty ratio
- oscilloscope settings, manual; coupling, voltage scale, time scale, trigger
- synchronization methods
- assemble circuit, measure capacitance by measuring rise time



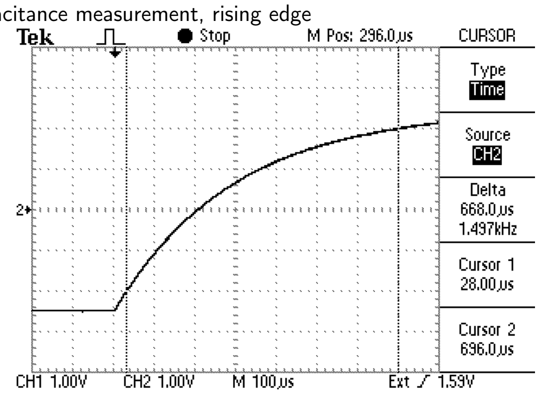
Lab 2: Oscilloscope and Signal Generator

capacitance measurement, waveform



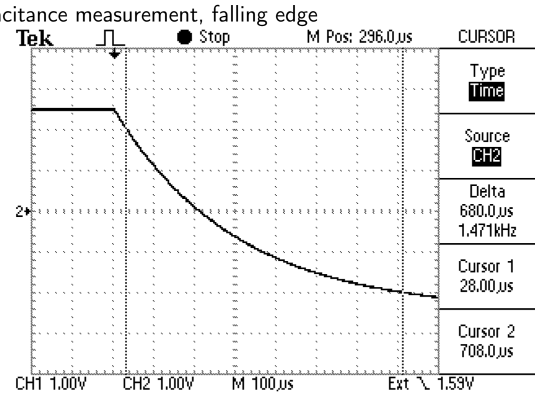
Lab 2: Oscilloscope and Signal Generator

capacitance measurement, rising edge



Lab 2: Oscilloscope and Signal Generator

capacitance measurement, falling edge

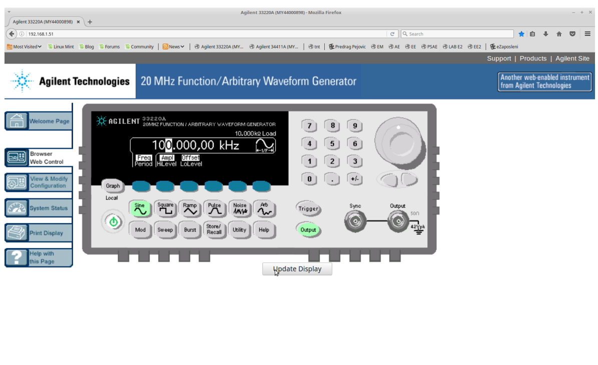


Lab 3: Control of Instruments Using a Computer, Automated Measurements, and Statistical Processing of Measurement Results

- really new!
- some experience in GNU/Linux assumed, very basic
- Python used to control instruments, not a prerequisite
- `ping`, verify connection, set IP address
- Agilent 33220A internal web server, web control of the instrument
- LXI compliant instruments, SCPI commands
- queries for the oscilloscope direct measurement
- statistical processing of measurement data, voltage average, 10 measurements manually, 100 and 1000 measurements automatically
- voltage divider, transfer curve, linear least squares
- the oscilloscope input impedance emerged as an auxiliary teaching topic

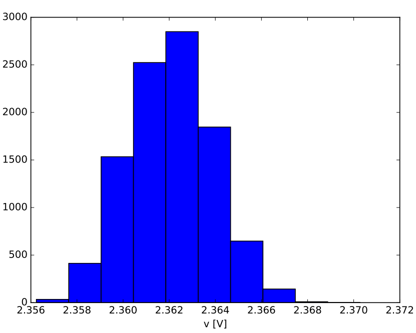
Lab 3: Control of Instruments Using a Computer, Automated Measurements, and Statistical Processing of Measurement Results

signal generator, browser control



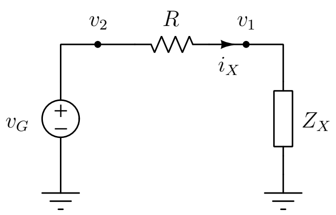
Lab 3: Control of Instruments Using a Computer, Automated Measurements, and Statistical Processing of Measurement Results

repeated voltage measurements, histogram



Lab 4: Impedance Measurement Using Oscilloscope

- relatively few new elements
- integrate previous exercises
- synchronized with Electric Circuit Theory course
- the circuit:



$$i_X = \frac{v_2 - v_1}{R}$$

Lab 4: Impedance Measurement Using Oscilloscope

just some formulas ...

- $R_X = R \, \frac{V_1}{V_2 - V_1}$
- $X_X = R \, \frac{V_1}{\sqrt{V_2^2 - V_1^2}}$
- $C_X = \frac{1}{2\,\pi\, f\, R} \, \frac{\sqrt{V_2^2 - V_1^2}}{V_1}$
- V_k, $k \in \{1, 2\}$ RMS or Pk-2-Pk values? Different statistics! Students are able to see the difference!

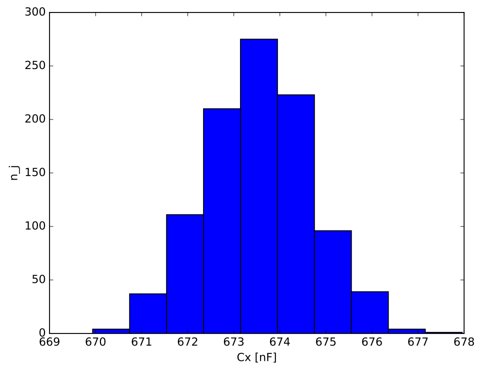
Lab 4: Impedance Measurement Using Oscilloscope

just some more formulas ...

- capacitor, relate voltage and charge
- $q_{C0}[k] = \Delta t \sum_{0}^{k} i_C[k]$
- $q_0 = \frac{1}{n} \sum_{0}^{n-1} q_{C0}[k]$
- $q_C[k] = q_{C0}[k] - q_0$
- linear least squares assuming $q_C = C\, v_C$, determine C
- similar to determining R using voltage divider
- statistics?
- $q_C\,(v_C)$ curve?
- similar for inductors; ellipse?
- very few new elements
- however, turned out to be difficult!

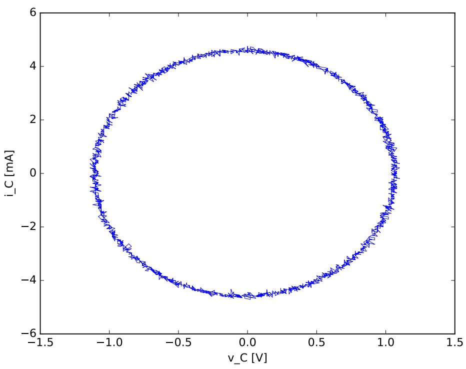
Lab 4: Impedance Measurement Using Oscilloscope

repeated capacitance measurement, RMS, histogram



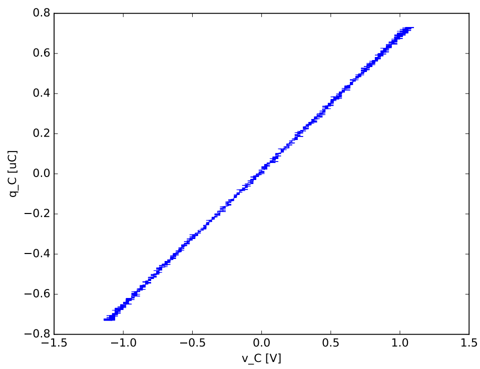
Lab 4: Impedance Measurement Using Oscilloscope

i_C versus v_C



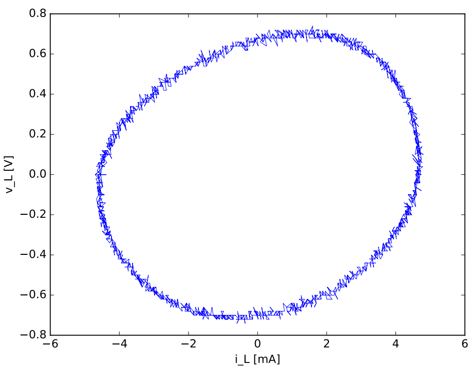
Lab 4: Impedance Measurement Using Oscilloscope

q_C versus v_C



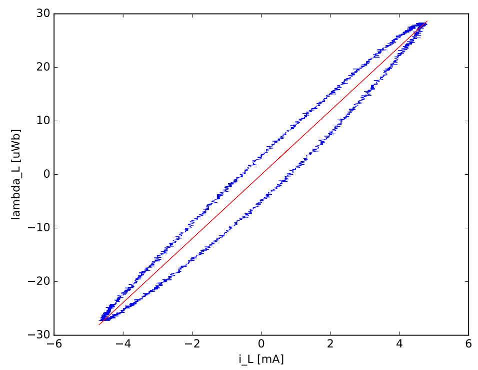
Lab 4: Impedance Measurement Using Oscilloscope

v_L versus i_L



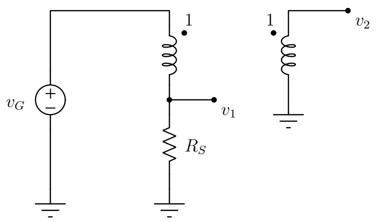
Lab 4: Impedance Measurement Using Oscilloscope

λ_L versus i_L

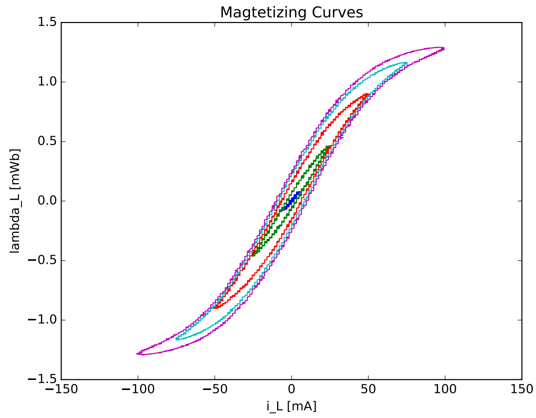


Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

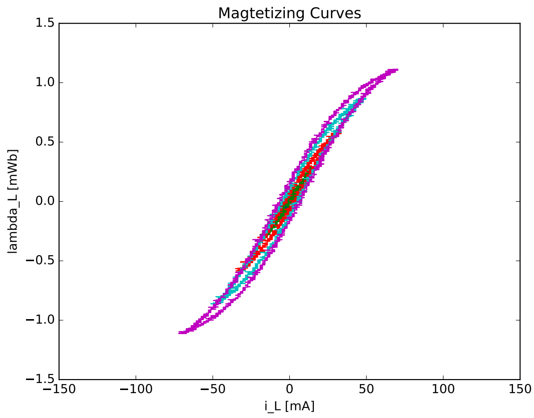
- theoretically demanding!
- how to measure magnetizing curve?
- careful in connecting!
- hysteresis curve
- mutual inductance
- dependence of mutual inductance on bias current



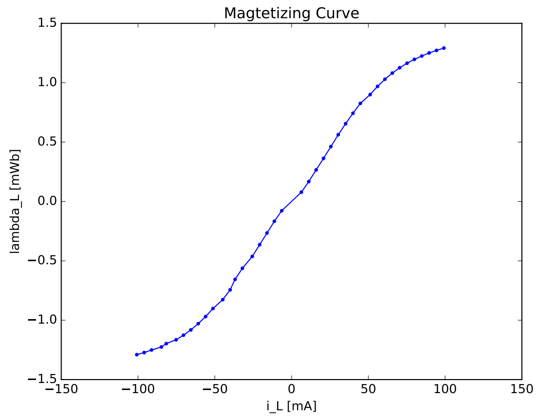
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



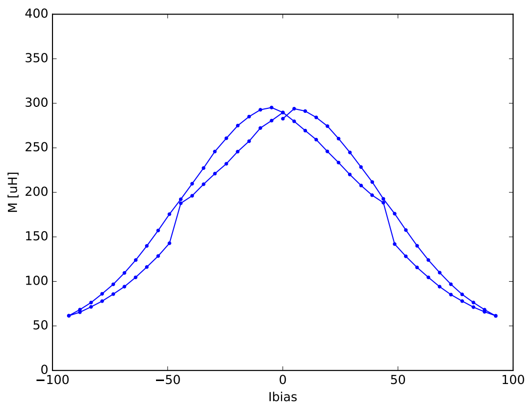
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



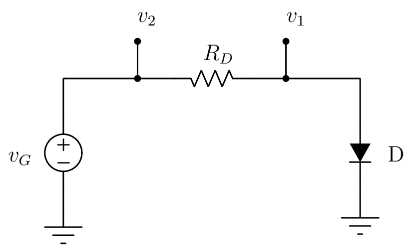
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



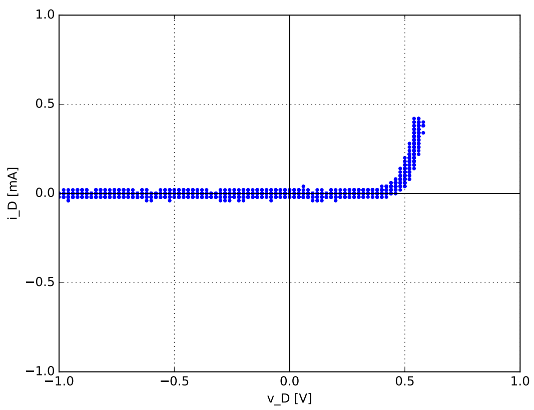
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



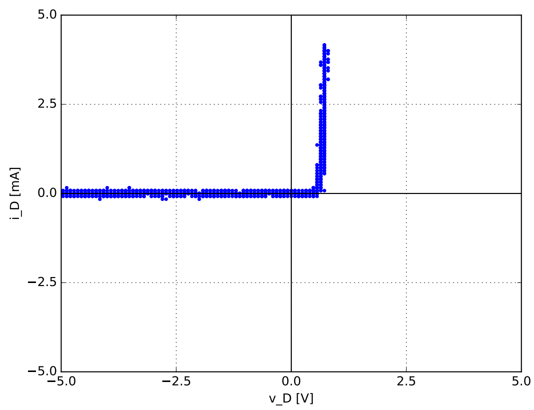
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



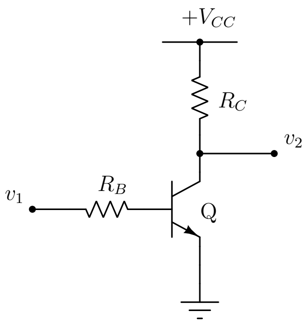
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

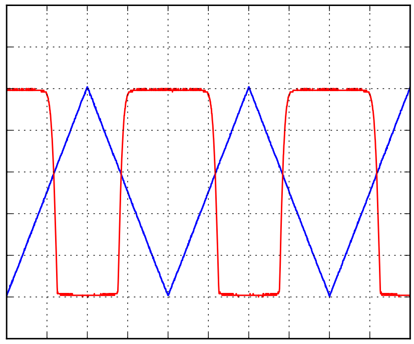


Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



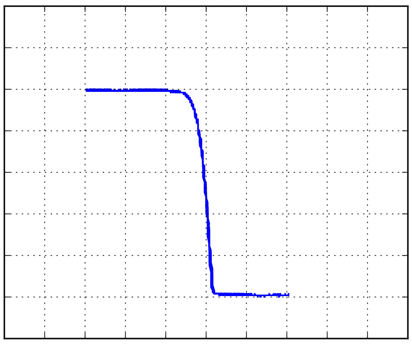
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_1 and v_2



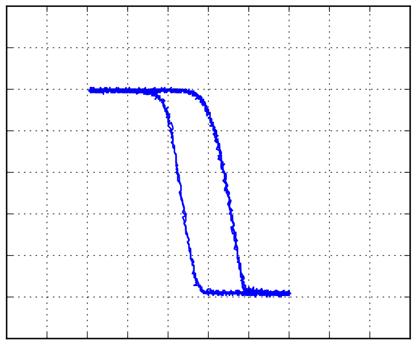
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_2 versus v_1, $f = 100\,\text{Hz}$



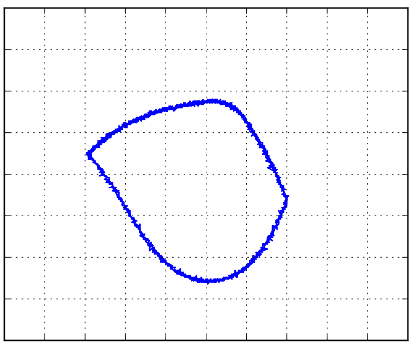
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_2 versus v_1, $f = 10\,\text{kHz}$



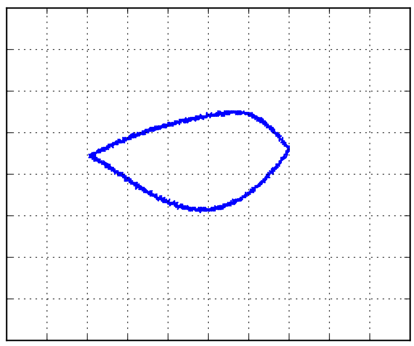
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_2 versus v_1, $f = 100\,\text{kHz}$

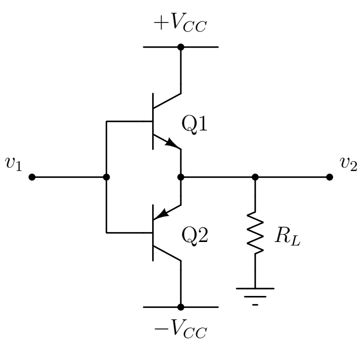


Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_2 versus v_1, $f = 200\,\text{kHz}$

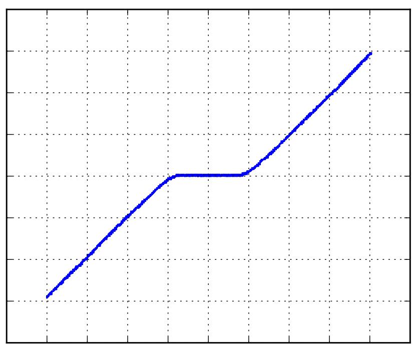


Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope



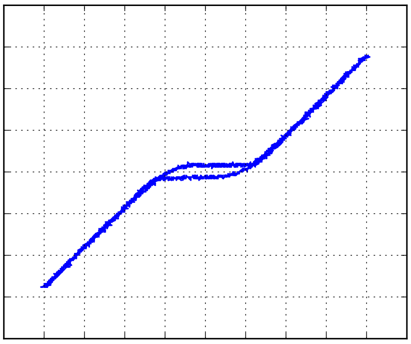
Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

v_2 versus v_1, $f = 1\,\text{kHz}$



Lab 5: Analysis of Nonlinear Elements and Systems Using Oscilloscope

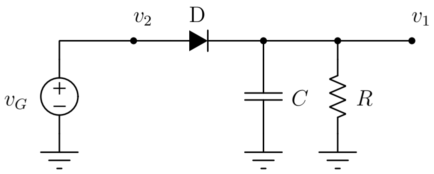
v_2 versus v_1, $f = 10\,\text{kHz}$



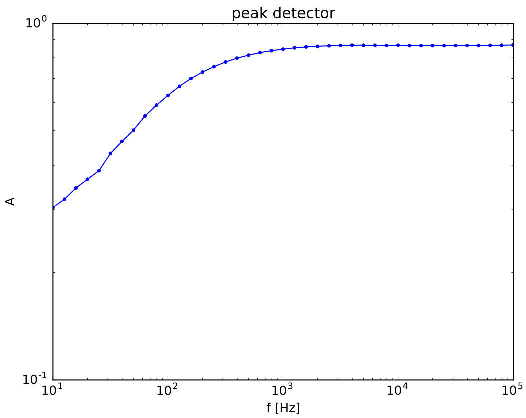
Lab 6: Measuring Parameters of AC Waveforms

- classical, common topic
- a little bit obsolete: I was not able to find to buy non-true-RMS instruments!
- still some educational value: diodes, real diodes, rectification, half-wave, full-wave
- both in voltmeters and ampere meters
- diode forward voltage drop, limitations at low voltages
- peak voltage detectors and their frequency response
- natural extension: envelope detector, link to Telecommunications course

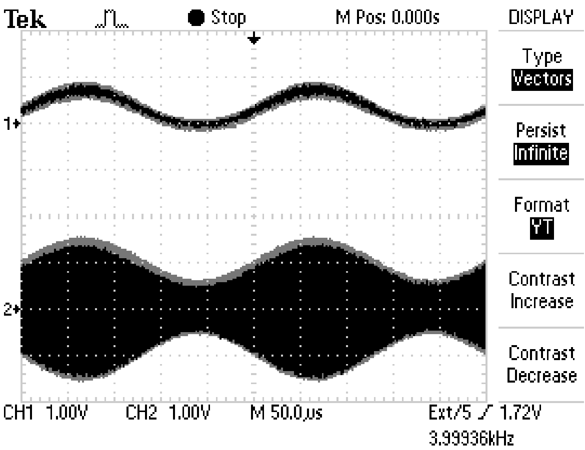
Lab 6: Measuring Parameters of AC Waveforms



Lab 6: Measuring Parameters of AC Waveforms



Lab 6: Measuring Parameters of AC Waveforms

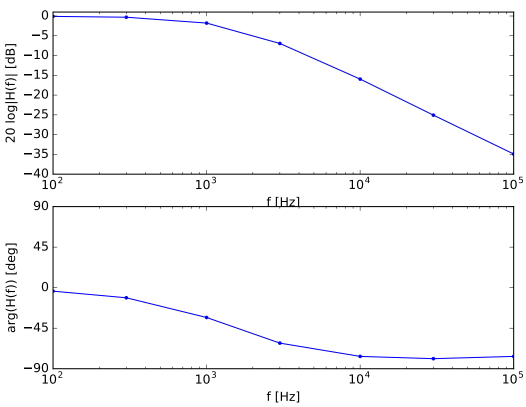


Lab 7: Frequency Response and AC bridges

- two topics: how are they related?
- they share the same course, which lacks spacetime
- frequency response: RC low-pass, RC high-pass, RLC band-pass
- bridge to Electric Circuit Theory
- aim: measure phase, emphasis on its **sign**
- manual measurements, especially phase
- frequency response of the oscilloscope input AC filter
- bridges are the classical topic
- De Sauty bridge, potentiometer to balance
- Maxwell bridge
- Wien bridge, to assemble and to measure frequency of the phase resonance

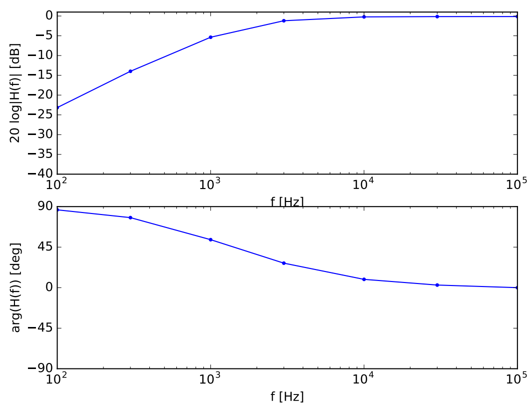
Lab 7: Frequency Response and AC bridges

frequency response of the low-pass filter



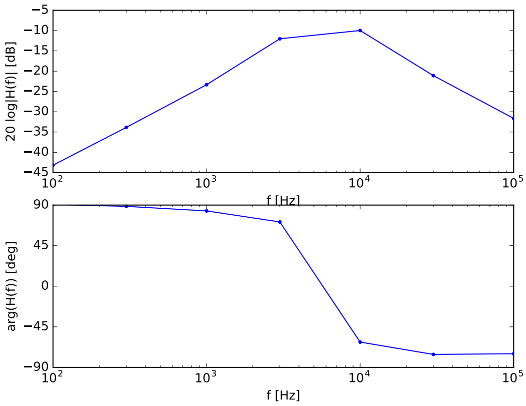
Lab 7: Frequency Response and AC bridges

frequency response of the high-pass filter



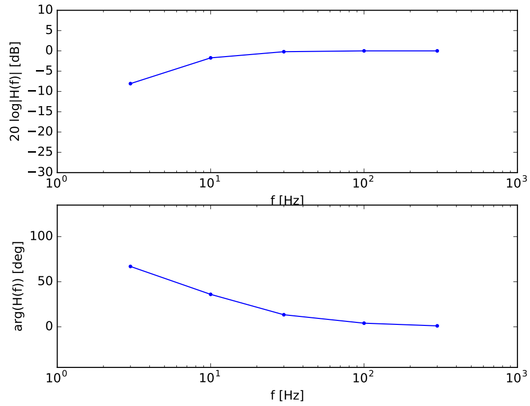
Lab 7: Frequency Response and AC bridges

frequency response of the band-pass filter



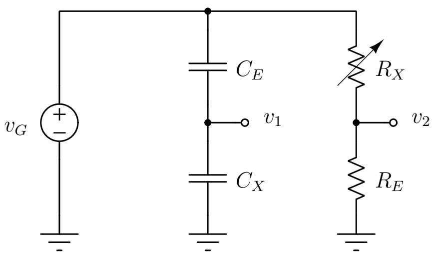
Lab 7: Frequency Response and AC bridges

frequency response of the oscilloscope input filter



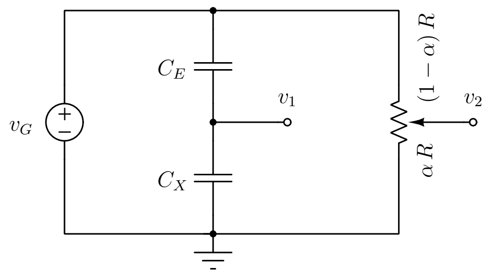
Lab 7: Frequency Response and AC bridges

De Sauty bridge, assemble and measure, protoboard



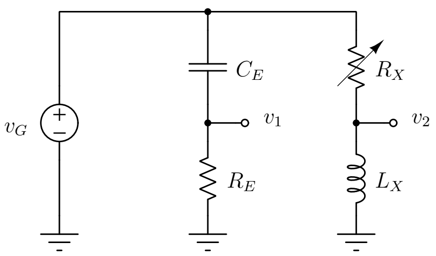
Lab 7: Frequency Response and AC bridges

De Sauty bridge with potentiometer, angle versus capacitance, no computing, assemble and measure, protoboard



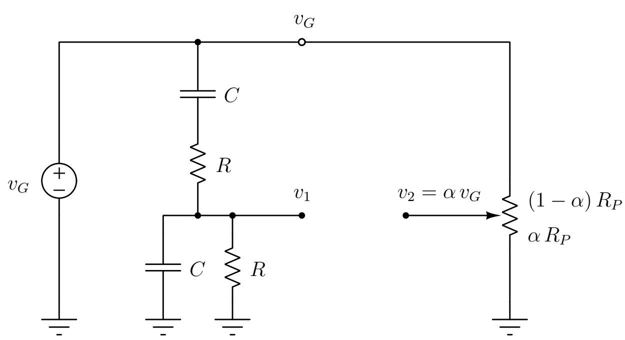
Lab 7: Frequency Response and AC bridges

Maxwell bridge, assemble and measure, protoboard



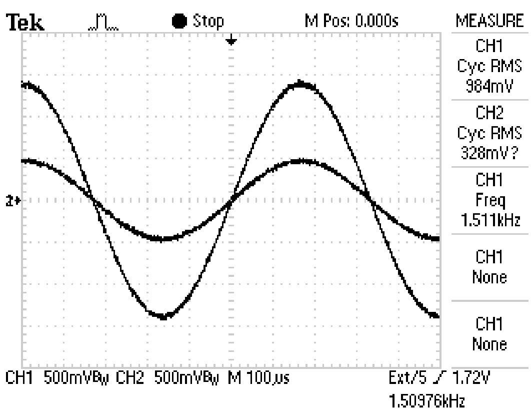
Lab 7: Frequency Response and AC bridges

Wien bridge, assemble and measure, protoboard



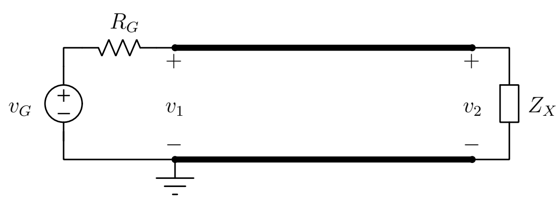
Lab 7: Frequency Response and AC bridges

Wien bridge, phase resonance



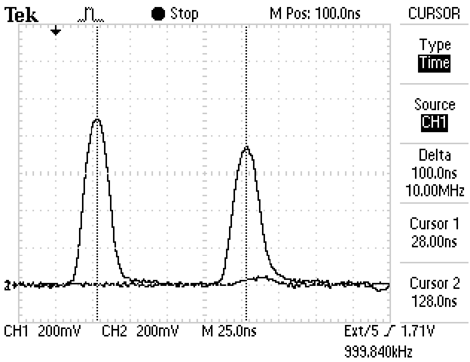
Lab 8: Measurements on Circuits with Distributed Parameters

- something really new
- to support Electric Circuit Theory course
- transmission lines are just a bunch of boring equations?
- never ever truly understood, at the end of the course, lack of time, boring, . . .
- let's see the lines in real life
- a cable from an old computer network, recycled



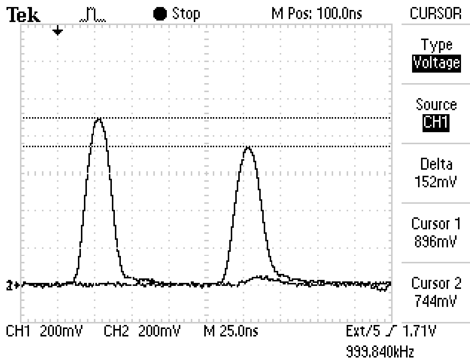
Lab 8: Measurements on Circuits with Distributed Parameters

propagation, delay



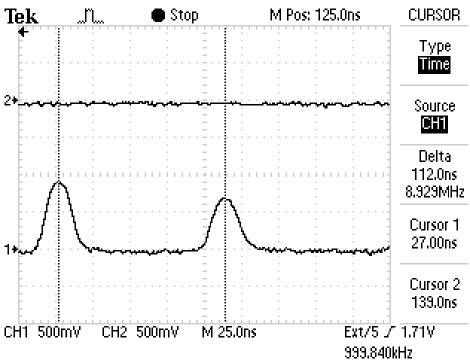
Lab 8: Measurements on Circuits with Distributed Parameters

propagation, attenuation



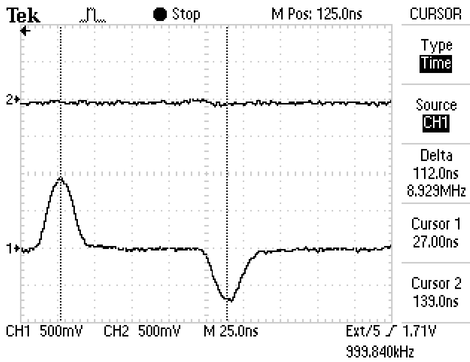
Lab 8: Measurements on Circuits with Distributed Parameters

reflection, open



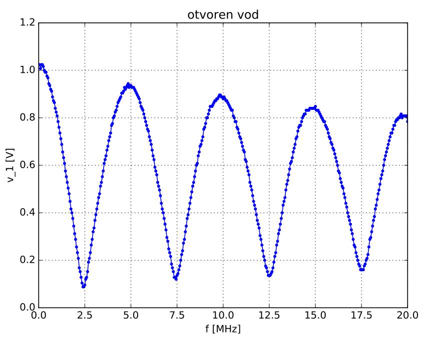
Lab 8: Measurements on Circuits with Distributed Parameters

reflection, short



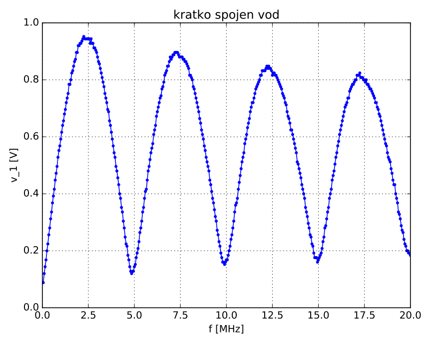
Lab 8: Measurements on Circuits with Distributed Parameters

voltage at the input versus frequency, open



Lab 8: Measurements on Circuits with Distributed Parameters

voltage at the input versus frequency, short



Experiences and Plans to Improve the Course

- something quite unexpected and new
- students worked hard during the lab sessions
- we had hard time to kick them out of the lab
- they **enjoyed** the lab!
- maybe a little bit too much
- questionnaire at the and, last minute idea
- 8 or 10 exercises?
- 80% voted for 10!
- really affirmative response

Grading

- 20% lab performance, during the exercises
- 20% lab performance, lab exam, the students are assigned to measure something, really close to the measurements they already did; however, this time they are alone, no pairs
- 60% written test
- really good results!
- almost all of the students already completed the exam with very good grades!

Conclusions

- a reformed course in Electrical Measurements presented
- customized to Electronics majors
- based on four principles
- computers heavily involved
- only free software!
- no donations, no dependence, everything open
- eight newly designed lab exercises
- understanding focused, not manual work
- everything available at http://tnt.etf.bg.ac.rs/~oe2em/
- success!
- students enjoyed the course!
- but even more surprising: teaching assistants and teacher enjoyed the course!

ZAKLJUČCI

Zaključci

- prikazana primena Python-a u merenjima
- automatizacija merenja, oslobađanje ljudi dosadnog posla
- nov kvalitet: merenja koja nisu mogla biti izvedena ranije
- veoma jednostavno!
- slobodno!
- moćno!
- korisno!
- napravljeni instrumenti koje nismo imali
- realizovana merenja koja nismo mogli izvoditi ranije
- ilustrovano na nizu primera
- primenjeno u praksi i u nastavi

Take it, use it, enjoy it: it's free!

Zaključci

Zašto je Python moćan u automatizaciji merenja?

- gluing language
- moduli:
  1. `numpy`
  2. `scipy`
  3. `matplotlib`
  4. `serial`
  5. `vxi11`
  6. `usbtmc`
  7. `time`
  8. `sys`
  9. `os`
- otvoren, lako se povezuje sa ostalim otvorenim softverom
- OOP, klasa `oscusb`