A Raspberry Pi module to scan over the resonance of piezo-tuning forks and do simple lock in applications.

Kai L. Kirn,[1] Amid Finkler,[1] Rainer Stöhr,[1] and Jörg Wrachtrup[1]
3th Physical Institute, University of Stuttgart

(Dated: 7 August 2018)

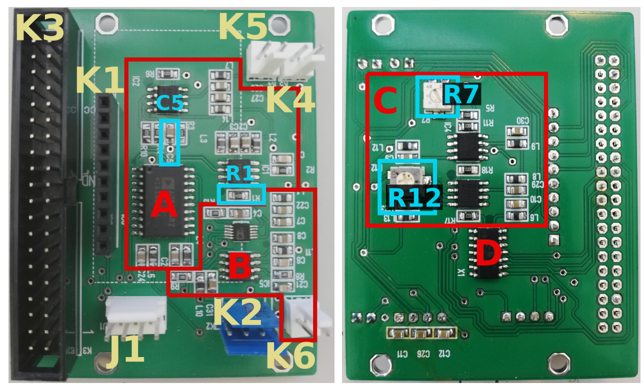
We present a device called Resosca (short for RESOnance SCAn) to scan electrically over the resonance of piezoelectric tuning forks. The presented device consist of a Raspberry Pi module for the creation and measurement of signals in the KHz regime and a software to control the module from a Raspberry Pi. As a sinus function generator the HC-SR08 module is used. It is based on the ADS9850, a CMOS, 125MHz, and Complete DDS synthesizer.
The used measurement principle is similar to the function of a lock in amplifier. Therefore, Resosca can furthermore be used for lock in amplification with a reference signal in the KHz regime.

Keywords: Raspberry Pi, lock in amplification, piezoelectric tuning fork

We introduce a device to scan over the resonance of piezo-tuning forks. It may be furthermore usable as a simple lock in amplifier in the KHz regime. The device consists of a Raspberry Pi and a home-brew PCB Raspberry Pi module. For the proper control of the module a Python software is introduced. The module has been tested for resonance scans on piezo tuning forks. The applicability and limitations for lock in amplification tasks has not been tested yet.
The whole project, including housing, PCB designs and the software can be found on GitHub[1].

I. HARDWARE

The module consists of 4 main parts (A, B, C, D), as shown in Fig. 1 and 2.
Beside the main parts there are the connectors K1 to K6 and J1 on the board. The connectors K5 to K6 have one pin on GND, each. To reduce noise twisted pair cables are recommended for these connectors.
J1 has two pins on GND. It is recommended to twist the cables of the two right and two left pins for noise reduction too. The module is designed to be driven by a Raspberry Pi over the connector K3. On the connector K1 a HC-SR08 module can be connected. It is a function generator based on the ADS9850, a CMOS, 125MHz, and complete DDS synthesizer. The HC-SR08 is capable to produce sine and square signals. Only the sine signal is used, though.
For the resonance scan with the later introduced software the HC-SR08 module is mandatory in the present configuration. For possible lock in amplifier tasks an external reference signal can be used too. However, its voltage should be between $-1\,\mathrm{V}$ and $1\,\mathrm{V}$ at all times. Unpredicted behaviour and non linearities have been observed for voltages $|U| > 1$.
The active pins of K4, K5 and K6 are connected to the signal-in-port of the lock in circuit (A), the sine output of the HC-SR08 and the input of the phase shifter (B), respectively.



(a) (b)

FIG. 1: In (a) the front side of the main-board of the module is shown. A is a lock in amplifier circuit. B is circuit to shift the phase of the reference signal. The connector K3 connects to the Raspberry Pi and the K1 connects to the HC-SR08 module. The Blue connector applies to the power supply. The white connectors connect to a front-board. (b) shows the back side of the main-board. C is a circuit to convert the $\pm 15\,\mathrm{V}$ output of the lock in circuit into a signal between $0\,\mathrm{V}$ - $5\,\mathrm{V}$. The potentiometers 1 and 2 are to fine tune the voltage conversion and D is an analogue to digital converter.

The two left pins of J1 are connected to both inputs of the AD630 in the lock in circuit (A) - SEL A and SEL B. The two right pins are connected to the sources which are meant to be connected to SEL A and SEL B (GND and the output of the phase-shifter (B)). With this configuration, the input of the AD630 can be set by either connecting pin 1 to pin 3 and pin 2 to pin 4 or by connecting pin 1 to pin 4 and pin 2 to pin 3. This can be achieved by e.g. connecting J1 to a switch. The inversion of the input signals of the AD630 results in an inversion of the measured signal too.
A power supply may be connected to K2. In the orientation of Fig. 1 the pins are from left to right

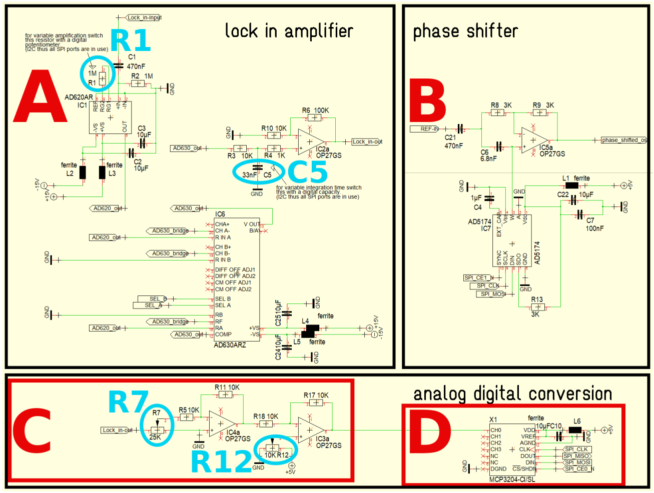


FIG. 2: The circuit plan of the Resosca main board shown in figure 1.
The power supplys of the op27's and the connectors K1 to K6 and J1 are not shown.

$-15\,\mathrm{V}$, $15\,\mathrm{V}$, $5\,\mathrm{V}$ and GND.

To perform a resonance scan of a piezo-tuning-fork the HC-SR08 module produces a sine-signal of a desired frequency. It is forwarded to the tuning fork and the input of the phase shifter B. The phase shifter applies a desired phase ϕ between $-176.5°$ and $-11.5°$ on the sine signal. Afterwards it is given as the reference signal to the lock-in-amplifier-circuit. Depending on the connection of the pins of J1 it is applied to one of the ports SEL A or SEL B. The other port will be on ground.
The response of the tuning fork comes over K4 to the lock_in_input in Fig. 2, A.
The lock in input signal is then lock in amplified with the reference signal from the phase shifter. The resulting output signal of the circuit A is fed to the circuit C. There the signal range is transformed from $-15\,\mathrm{V}$ – $15\,\mathrm{V}$ to $0\,\mathrm{V}$ – $5\,\mathrm{V}$ to be suitable for the analog-digital-converter (AD-converter) D. The signal shape is not affected by this process.

Following the parts A (lock in amplifier), B (phase-shifter), C (voltage transformer) and D (AD-converter) seen in the figures 1 and 2 are explained in detail.

A. LOCK IN AMPLIFIER

The heart of the lock in circuit is the AD630 Balanced Modulator/Demudulator.
Already in the documentation of the AD630 is a circuit for lock in amplification presented. After some additional research[2,3] the circuit (A) in Fig. 2 has been created.
The circuit consists of a pre-amplifier (AD620), the AD630 modulator/demudulator and a low pass filter (op27/IC2a).
By entering the circuit trough a $470\,\mathrm{nF}$ the signal is cleansed of any DC parts. The pre amplification depend on the value of $R1 = 1\,\mathrm{M\Omega}$. It has been chosen to be on a fixed value instead of a register of resistors, since the device is build for a certain task where the input signal intensity is in a constant range. According to the manual of the AD620 the amplification is calculated as

$$G = \frac{49.9\,\mathrm{k\Omega}}{R1} + 1 = \frac{49.9\,\mathrm{k\Omega}}{1\,\mathrm{M\Omega}} + 1 \approx 1 \qquad (1)$$

The pre-amplification is apparently not necessary for our task. It has been implemented for reason of completeness and for easy modification to other tasks, though.
The low-pass-filter before the output of the circuit (A) has also a fixed cut of frequency f_c. It is defined by the RC-element, formed by R3 and C5. The cut-of-frequency calculates to

$$f_c = \frac{1}{2\pi\ R3\ C5} = \frac{1}{2\pi\ 10\,\mathrm{k\Omega}\ 33\,\mathrm{nF}} \approx 480\,\mathrm{kHz} \qquad (2)$$

f_c can be adjusted by replacing the capacitor C5 with another capacitor.

B. PHASE SHIFTER

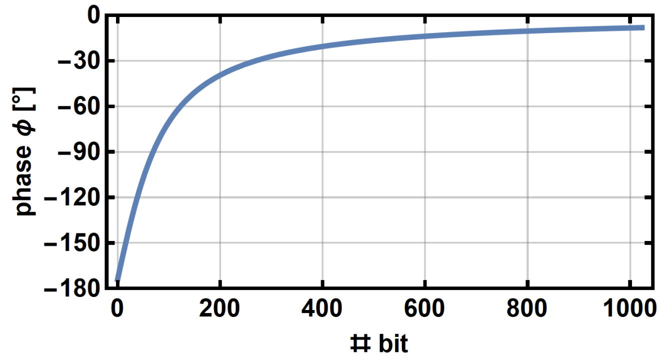


FIG. 3: The theoretical behaviour of the phase-shifter. On the Y-axis is the phase and on the X-axis the possition of the 10 bit register of the AD5174 programmable resistor.

The phase-shifter is realized as a high-pass equivalent of an all-pass-filter with an AD5174 programmable 10 bit resistor. It apply a phase on the reference signal. The possible range of the phase depends highly on the phase ϕ of the reference signal, the capacitance of C6 and the resistance of IC7. The phase shift follows the rule[4,5]

$$\phi = -\pi + 2\arctan\left(\omega R_{IC7} C_{C6}\right) \qquad (3)$$

Whereas, ϕ is the phase, applied to the input signal, ω is the frequency of the input signal, R_{IC7} is the resistance of the digital resistor IC7, and C_{C6} is the the capacity of the capacitor C6. Since C6 is a constant, the phase-shift depends on the frequency of the input signal and the resistance of IC7. To maximise the range in which the phase can varied for the ideal frequency $f_{pt} = 2^{15}$Hz $=$ 32 768 Hz of the piezo-tuning-forks in our focus, C6 is set to 33 nF.

C. VOLTAGE TRANSFORMER

The voltage transformer C in Fig. 1 and 2 transforms the voltage range of the output of the lock in amplifier to a range suitable for the AD-converter. The output range of the lock in amplifier is −15 V to 15 V. While the input range of the AD-converter is 0 V to 5 V.
The transformation takes place in two steps. In the first step the signal is attenuated to a −2.5 V to 2.5 V range. The attenuation can be tuned by the potentiometer R7. In the second step the signal is risen by 2.5 V to end with the desired range of 0 V to 5 V. The shift of the signal can be tuned by the potentiometer R12.

D. ANALOG-DIGITAL-CONVERSION

D is a MCP3204 12 Bit, 4 channel analog-digital-converter. It is the final peace of the circuit. It translates the signal in a form, understandable for the Raspberry Pi and the software, described in the following section.

II. SOFTWARE

For the proper use of the device a python software with a PyQT GUI and PyQTGraph graphs has been developed. It has two modes. A resonance scan mode for the scan over the resonance of tuning forks and a continuous scan mode for normal lock in applications. The different panels of the software are shown in Fig. 4. To achieve live images of the measurements, the acquisition of the data is set into a different thread than the GUI.
The software is split into six files.

- main_thread.py houses all the in and out of the GUI and by running it the software is started.
- acquisitionThread.py is home to the two acquisition threads of the two measurement methods described in the following sections. It also contains the fit routine and all the communication with the hardware.
- digipot.py is the driver for the AD5174 digital potentiometer.
- HC_SR08.py is the driver for the HC-SR08 dds module.

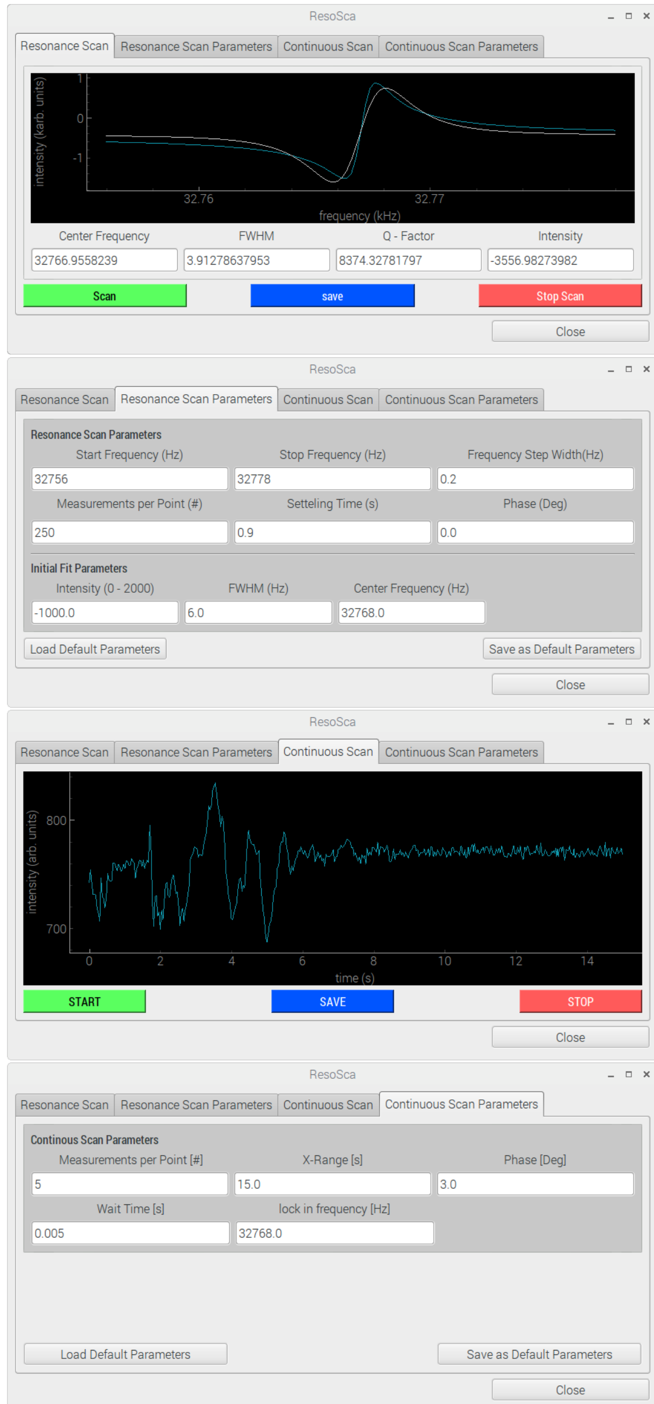


FIG. 4: Te ResoSca python software with a PyQT gui and PyQTGraph graphs. The top two windows are the resonance scan application. In the upper window the scan can be started, stopped, saved and live observed. At teh end of the scan a differential lorentzian is fitted to the data automaticaly. In the lower window the values for the scan can be set.
The lower two windows show the continous scan application. In this mode normal lock in tasks can be conducted. The output signal of the circuit is shown in real time for a fix reference frequency. The data, shown in the graph can be saved on the fly or after the measurement has been stoped. The lower window of the two shows parameters, that can be set for the measurement.

- mcp3204.py is the driver for the MCP3204 AD-converter.
- if present, the files parameters.csv and parameters_cont.csv (cont for continuous scan) hold the saved default parameters.
- The folder gui contains a .ui file (QT-designer file), a bash script to convert it to a python file and the corresponding python file, containing the GUI.

The bash script, which converts the ui file to a py file is run at the start of the main_threat.py file. Therefore the ui file in be altered e.g. in the QT designer and is loaded in the updated state wit the start of the program.

A. RESONANCE SCAN

1. Principle of measurement

The resonance scan mode is designed to scan over the resonance of piezo-tuning forks. Therefore sine signals with rising frequencies are send to the tuning fork and the response is lock in amplified with the same sine as reference. The tuning fork is directly electrical excitation at one tine. The response is measured at the other tine. The closer the frequency of the excitation comes to the resonance, the better the excitation will be transported from one tine to the other. As a result the intensity of the response signal will raise in proportion to the distance to the resonance in the frequency domain.
With this approach no additional excitation signal is needed. Since, in contrast to normal lock in usage, the frequency of the reference signal is used for the excitation of the sample too.

2. GUI explanation

In the resonance scan window the data is plotted life, while the scan is done. after the scan, a differential Lorentzian is fitted on the data. The parameters of the fit are shown under the graph. The scan can be stopped at any time with the red stop button. The fit will be done anyway.
After scanning, the data can be saved with the blue save button. The software creates three files, one for the data, one for the fit points and one for the fit parameters. Each file has a time stamp with minute precision.
In the resonance scan parameters window all the relevant parameters for the scan can be set. Start and stop frequency define the scan range. The frequency step width defines the frequency resolution of the scan. The minimal step width is limited by the HC-SR08 frequency generator module to 0.026 Hz. The measurement per point perk defines over how many measurements per frequency it will be integrated over.
To give the tuning fork a time to swing in between the measurements a settling time can and should be set. As a rule of thumb is, that the settling time has to be larger for larger Q-factors, e.g. 1 s for a Q-factor of 10 000.
As last parameter for the scan a phase can be applied to the reference signal only. The possible apply able phases are between $-174°$ and $-8°$. Even though the applied phase is negative, positive phases can be entered. The system always will calculate the absolute of the entered value. The phase range is hardware limited by the phase shifter, described in section I B. For any entered value above or under the maximum ore minimum value the maximum or minimum value will be set for the phase, respectively. The phase should in the actual state be set close to 0° or 180° to end with a differential Lorentzian signal. For a phase of roughly 90° the signal is a normal Lorentzian.
In the section initial fit parameters the initial parameters for the fit can be set. The exact formula for the fit is

$$L(x) = I\frac{-2\gamma^2(x-x_0)}{((x-x_0)^2+\gamma^2)^2} + y_0 \qquad (4)$$

with the intensity I, the x-shift x_0, the y-shift y_0 and the half width at half maximum (HWHM) γ. The initial parameter for y_0 if calculated automatically as the average of the first 10 % of the signal.
All parameters can be saved to and loaded from a file with the contributing buttons.

B. CONTINUOUS SCAN

1. Principle of measurement

The continuous scan is the normal lock in mode. The frequency of the out signal has the same fix frequency and amplitude as the reference. The input signal is then lock in amplified with the reference signal. This mode can also be used with a external reference source.

2. GUI explanation

The continuous scan window shows the input signal over a certain time. The measurement can be started via the green start button and stopped via the red stop button. The data on the screen can be saved via the blue save button. Data which has been wandered of the screen already is lost and can't be restored!
In the continuous scan parameters window the parameters for the scan can be set.
Measurements per point gives the number of measurements over which the software integrates for each measurement point. The x-range defines the x-axis in seconds. Since only the data on the screen can be saved, the x-range also defines the time range over which data

can be saved. The phase defines a phase which is applied on the reference signal only. The same limitations and rules as shown in the previous section II A 1 for the phase, apply here too.
As in the resonance scan, a wait time can be set too. Even though it might not necessary for the task it is highly recommended to set one > 0. Otherwise lagging can occur which might have its origin in the event loop of the QT GUI.
The frequency of the reference and output sine can be set via the lock in frequency field, if the internal function generator is used. For the specification of the function generator, please review the documentation of the HC-SR08[6]. Values over 40 kHz haven't been tested.
All parameters can be saved to and loaded from a file with the contributing buttons.

III. ACKNOWLEDGEMENTS

Special thanks goes to Norman Karle and Dr. Axel Griesmaier for their help and support on the PCB design and manufacturing. Further special thanks goes to Matthias Niethammer for his crucial help on programming the software.

IV. OUTLOOK

- Variable pre-amplification by replacing R1 with a potentiometer or a stack of resistors.
- Variable integration time of the lock in amplifier by replacing C5 wit a variable capacitor or a stack of capacitors.
- Put the lock in circuit on a separate PCB so that it can be used as an independent element. For analog, only lock in tasks the remaining circuit it not necessary.
- replace the potentiometers R7 and R12 with a fix resistor and a small potentiometer for fine tuning. Or with two potentiometers, one for rough and one for fine tuning. Right now they are way to sensitive.
- Change the housing design so, that the power cables don't have to go all through the housing. From above sight with back side on top, power plug to the right, main board to the left and power regulator mount to the ground on the right.

[1] "ResoSca on GitHub," `https://github.com/KaKir/ResoSca`, accessed: 2018-08-7.
[2] S. K. Sengupta, J. M. Farnham, and J. E. Whitten, "A simple low-cost lock-in amplifier for the laboratory," Journal of chemical education **82**, 1399 (2005).
[3] Y. L. Liu and R. Zhang, "Ad630 lock-in amplifier circuit for weak signal," in *Advanced Materials Research*, Vol. 482 (Trans Tech Publ, 2012) pp. 975–980.
[4] U. Tietze, C. Schenk, and E. Gamm, *Electronic circuits: handbook for design and application* (Springer, 2015).
[5] Source ([4]) contains only the description of the low pass equivalent of an all-pass-filter. Though the difference between the two equivalents are only that the phase shift goes from $-180\,^\circ$ to $0\,^\circ$ or from $0\,^\circ$ to $-180\,^\circ$ in the case of the high-pass or the low-pass equivalent, respectively.
[6] "HC-SR08 documentation," `http://www.electronicoscaldas.com/datasheet/HC-SR08.pdf`, accessed: 2018-08-7.