Improvement of the Second Sound Evaluation Program

Philipp Horn, Dresden University of Technology, Germany

September 4, 2013

Abstract

Superconducting cavities are very valuable devices to accelerate electrons. They will be used for accelerators like XFEL and ILC. The improvement of the quality and the field gradient would reduce energy loses and provide more acceleration for the particles. Therefore we need improved methods to inspect these cavities and optimize the actual ones. Second Sound is a established approach, because it achieves the same resolution as t-mapping in less time. My work at the FLA Group at Desy included extending some cavity inserts, taking part at vertical test with second sound measurements and improving the current evaluation program for second sound. This report explains all steps from the construction of the second sound measurement system till the analysis of the data including a comparison of the old and the new software code.

Contents

1 Introduction

Superconducting cavities are a useful and established tool for modern physics. They offer one of the most efficient ways to accelerate electrons. They gain energy via a electromagnetic standing wave. This is only possible because of the superconductivity, which minimizes the energy loss and the heat generation. Two main ways to use these particles for research is described in the next paragraphs. On the one hand they can be send through a undulator, a special assembly of magnets, which forces the electrons on a serpentine line. This leads to an emission of Synchrotron radiation. It is coherent, polarized and can for example be used in many fields of physics, biology and chemistry to look at the structure of objects or even on chemical reactions. At the moment there are thousands of requests of scientists to get beam time at the Positron-Elektron-Tandem-Ring-Anlage (PETRA) III or the Freie Elektron Laser in Hamburg (FLASH). Furthermore there is a new accelerator the European XFEL (X-Ray Free Electron Laser) in Desy under construction. It will be 3 km long with 5 end stations for photon science. On the other hand the electrons themselves can be used in colliders, like the Large Electron-Positron Collider (LEP) at Cern. The big advantage of leptons compared to protons is a much more precise measurement, because they are elemental particles and therefore the center of reaction is clearly defined as is the energy which is taking part in it. Their ability to emit synchrotron radiation is a big disadvantage in this field. It is more difficult to achieve higher center of mass energies in circular accelerators than in linear machines. Therefore the trend is going in the direction of linear accelerators which are used as colliders. The next international project is the International Linear Collider (ILC), which is now in the planing phase and will be 30 km long in the final state. The construction will probably start in 2015 and it will be finished in 2026. The possible construction site was recently announced by the Japaneses government.
To produce synchrotron radiation of a higher brilliance and to achieve more in collisions it is essential to accelerate electrons to higher energies. One way is the increase of acceleration length, but it would be more effective to improve the energy gain per accelerator length in addition to a high quality factor. XFEL already requested a energy gain of 23.6 MV/m and ILC needs 35 MV/m. There are some calculations of a theoretical maximum of 55 MV/m [1]. This means it is in the regard to superconducting radio frequency cavities as accelerating modules to raise the used acceleration gradient field and also the quality factor of the cavities itself. Systems to inspect and compare the effective with the ineffective ones are an essential step in this progress. There are some performance limitations due to production and the high electrical fields. If the field is high enough there will be a thermal breakdown eventually and the cavity will become normal conductive at one spot, usually impurities at the surface. It is important to delay this process as long as possible and therefore different techniques have been developed to observe the cavities or find impurities. At Desy there are at the moment OBACHT, which is a high resolution camera to see the inside of a cavity directly, the temperature measurement T-mapping and a system of detectors, which use the effect of second sound in Helium. This report will only be about the third possibility.

2 Background

2.1 Superconducting cavities

The superconducting TESLA-shaped cavities used in FLASH and in XFEL and ILC as well consist of 9 cells and accelerate the electron with a radio frequency (rf) standing wave. The frequency is around 1.3 GHz. In figure 1 the view of such a cavity is shown.

Figure 1: Superconducting cavity

The wave is injected into the cavity with the power coupler until there is a standing wave in the cavity The high order mode coupler (HOM) rejects the lower order modes, because they cause heat losses. To use the cavity within an accelerator, it is mounted into a cryostat module which has a specific temperature gradient where in the cavity itself is cooled to 2 K while in operation. Within a cryostat module generally more than one cavity is mounted as for example for European XFEL eight cavities are mounted in one module. Because of the necessary cryostat modules and the constant supply of expensive Helium the cooling system brings a lot of difficulties, but in exchange there is only a minor energy loss P_{diss} in regard to the stored energy U, which is expressed in the high quality factor of about $Q_c = 1 \cdot 10^{10}$, defined by

$$Q_c = \frac{\omega U}{P_{diss}} = \frac{\omega}{\Delta\omega}$$

. Applying the radio frequency ω the frequency bandwidth $\delta\omega$ is very small. With higher energy gain per meter the quality factor decreases. In figure 2 a typical Q_c vs. E_{acc} curve is shown.
There are different reasons for the decrease of the quality factor, like Multipacting, field emission of electrons, thermal instabilities and quenches [2]. I want to concentrate on the quench. Because of the higher electric field the cavity heats up and becomes normal conducting if the critical field or temperature is exceeded at some point. This resolves in a huge rise of temperature and evokes the process called second sound within the 2 K helium bath. On this spot there is usually a impurity in the surface located. A detection could help improve the cavity.

2.2 Second Sound

The effect of Second Sound is essential for the quench detection. Around and below two Kelvin Helium 4 can be described of two liquids: a normal (Helium I) and a superfluid

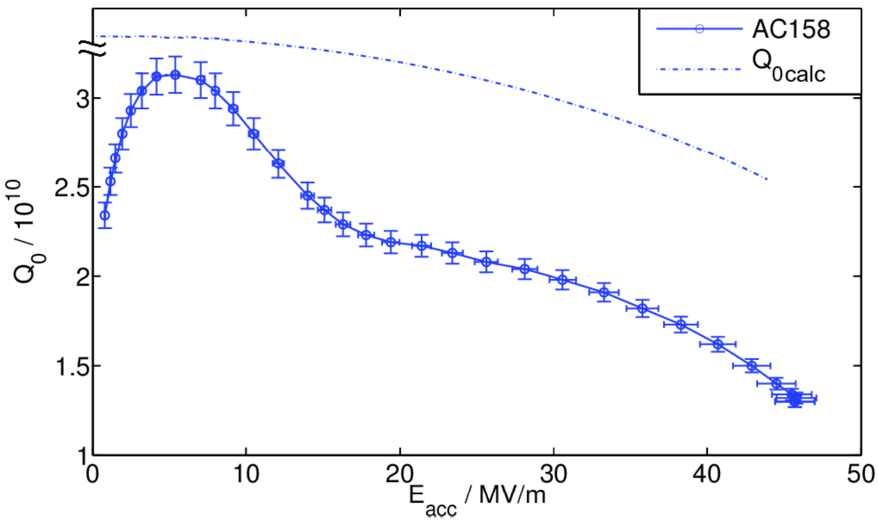


Figure 2: Q_c vs E_{acc} curve

(Helium II) one. The second liquid has very low friction. It can be compared with superconductivity. This means, that it can flow through thin tubes without pressure changes. The quench produces a heat impulse and the temperature is carried by the liquid in a wavelike motion. For further information see [3].

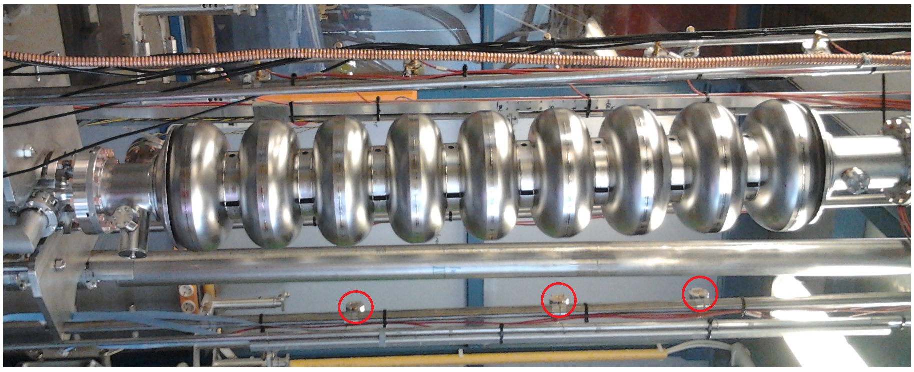


Figure 3: Insert with cavity and detectors (image rotated for 90 degrees)

During the quench detection with second sound there are 16 detectors positioned around the cavity, which can detect the density wave. In figure 3 you see a complete insert with cavity and second sound detectors, some marked by red circles.
Like shown in figure 4 the oscillating superleak transducer (OST) consist of a metal plate and a membrane. When set under voltage of 120 V they form a capacitor. Only the superfluid flows through the membrane and when the wave arrives at the detector the membrane oscillates because of the difference in densities. The change of capacity

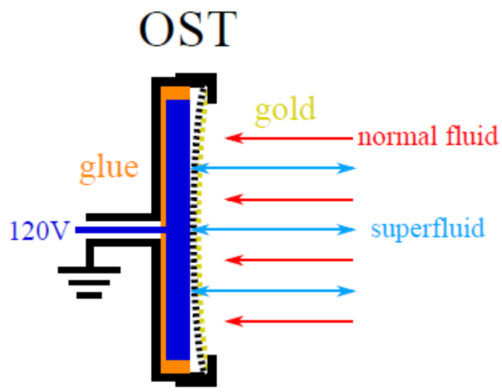


Figure 4: Setup of detector

can be measured by the variation of the voltage. The reflected signal of the cavity shows a abrupt change of intensity at the time of the quench and therefore yields the quench time. From the time difference between the quench, the arrival of the signal at the detector and the knowledge of a approximately constant velocity of 20 m/s we gain the distance between the quench spot and the OST. The velocity vs temperature curve is shown in figure 5.

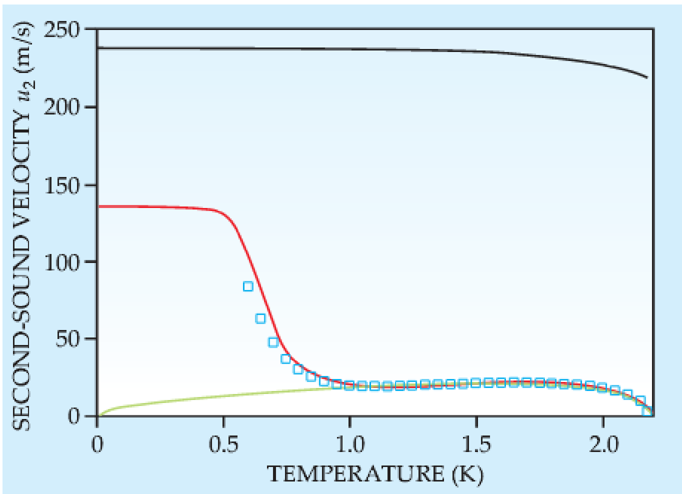


Figure 5: velocity-temperature-curve of second sound

If we use the fact, that the quench only occurs at the surface of the cavity as a restriction as well there are only 2 detectors needed to calculate the spot, but more detectors provide a more precise result.

3 Hardware

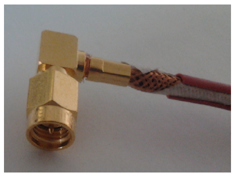


Figure 6:

Half of my work at Desy consisted of attaching detectors near the cavities, checking their functionality, measuring the coordinates of the OST for the evaluation later and perform an actual second sound measurement at a vertical test. First there were only eight detectors in use, but it is necessary to have a direct line of sight between them and the quench spot. Therefore a second set with another eight detectors was installed. The output data in the end of the vertical measurement from the 16 OSTs are divided in a time scale and the two detector sets. Each set consists of 8 OSTs, a cable for noise and a rf signal, from the reflected wave.

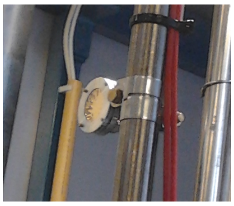


Figure 7:

First the OST plugs have to be soldered at the cables and the holders have to be mounted at one of the four poles around the cavity. In figure 6 a plug with cable is shown. A detector attached to the pole via a holder can be seen in figure 7. With this mechanism the OST can face away from the cavity to prevent damage during the mounting progress and with the aid of the groove you can return the detector to the same position afterwards. Each cell, except the middle one has two detectors near its equator. Their functionality is checked with compressed nitrogen. Therefore the gas is blown on the OST and the change of the electrical signal is recognized.
We use a cylindrical coordinate system with the point of origin in the first cell and the x-axis in perpendicular to the power coupler. The angle is defined anticlockwise. The method of the measurement of the distances and the angle with ruler and laser pointer still needs improvement, but is sufficient at the moment.
The vertical test stand is a 600 liter cryostat, which achieves lower temperatures than 4.2 Kelvin (Boiling point of liquid Helium) due to pumping over the fluid. It vaporises and drains energy of its surrounding. During the vertical test the rf acceleration field and the quality factor are measured indirectly via the power transfer within the cavity. The forward power, reflected power and the transmitted power are measured directly. This gives conclusion about the performance of the cavity. The acceleration field is increased step by step until the quench occurs. While the quench occurs the second sound measurement takes place. The reflected power signal from the cavity serves as rf signal.

4 Software

4.1 Old program

An important part of my work at Desy was the improvement of the current evaluation program of the second sound data. The major new code segments are visible in the attachment. The main problem was the unfinished upgrade from 8 to 16 detectors. In the code all OSTs referred to the one rf signal, because they seemed similar enough. Unfortunately there was a difference of around 60 ms between the quench times. In figure 8 the consequence can be seen. Half of the signals are cut off. After a compromise solution of shifting the matrices the two sets are now treated completely separately and put together after the running time is already calculated.

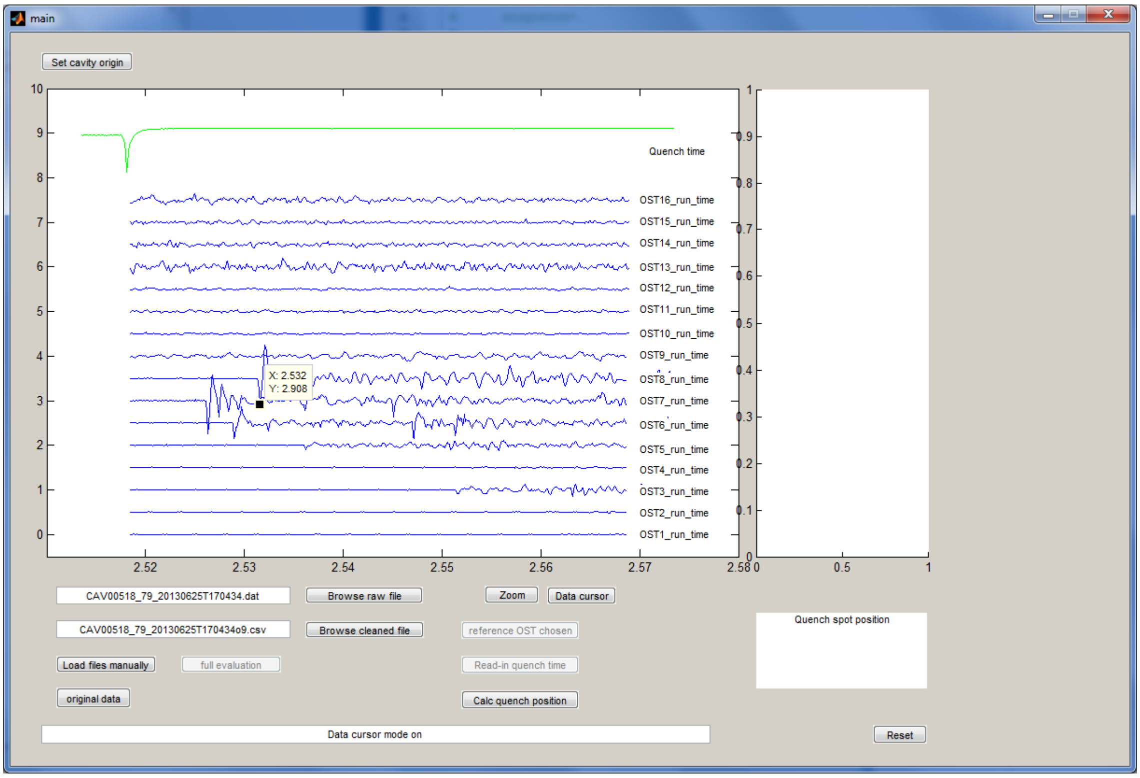


Figure 8: The old layout

In figure 8 you also see the layout of the old program. After browsing the raw file, a cleaned file was created automatically. Afterwards it was possible to load the file and the first quench time was calculated. The rf and the 16 OST signals were plotted and you had to pick fitting detectors and select the arrival times of the wave within the signals. Afterwards the calculation of the quench spot could be completed. The two buttons "reference OST chosen" and "Read-in quench time" are in use, if the program could not find the quench time by itself.

4.2 New Layout

In figure 9 is the new layout shown. The cleaned file was deleted. In this process the 9th cable, the noise was only deducted from the raw signal. A noise reduction is due to the great signal noise ratio not necessary. After pressing "Running times" the signals are being plotted, even if no quench time is detected. The big difference is the possibility

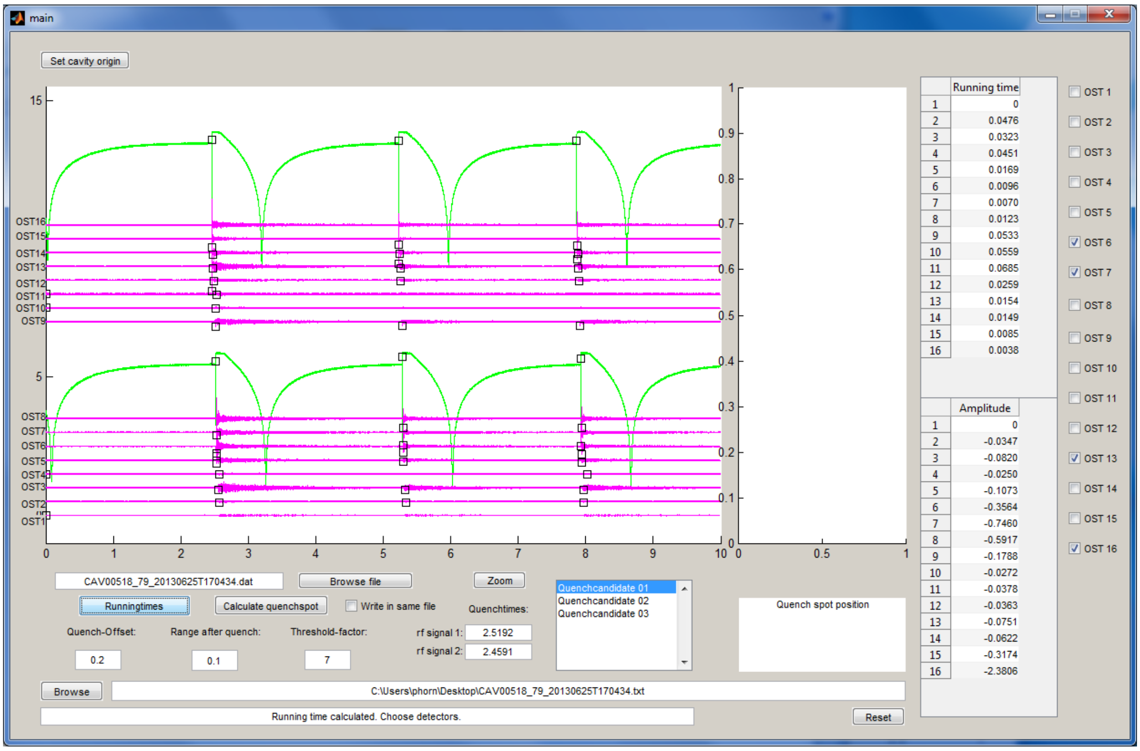


Figure 9: The new layout

to use at another quench besides the first one and the arrival time of the wave at the OSTs is calculated automatically. You pick the detectors via a check box and calculate the quench spot similar to the old program.
Another advantage is the increase of output. After choosing the quench you can see both quench times of the rf signals. In addition almost every detector is assigned to a running time and amplitude. With this information you can choose your detectors manually easier than before. The program even picks the two highest amplitudes, which often belong to the nearest detectors and picks another two OSTs on the same pole and suggest them to the user. In this case it is very unlikely to select a detector, which is not in direct line of sight to the quench spot. Of course this suggestion can be changed by the user.

4.3 Edit fields

The program offers the user three edit fields in the lower left corner of the layout. The first one concerns the quench time detection. This value was set as 0.1 in the original program and determine the biggest difference between the maximum amplitude and the amplitude at the moment of the quench. When this value is too small a quench could remain undetected like shown in figure 10. A bigger Quench-Offset solves this problem, like you can see in figure 11. When you choose a even higher offset, too many quench

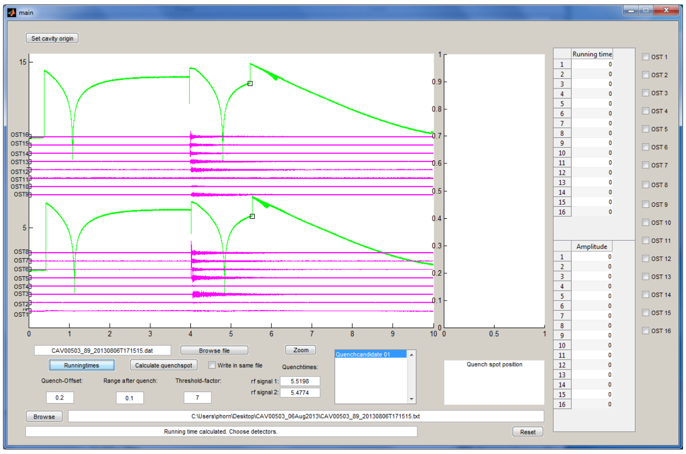


Figure 10: Quench detection with offset=0.2

candidates could be detected.

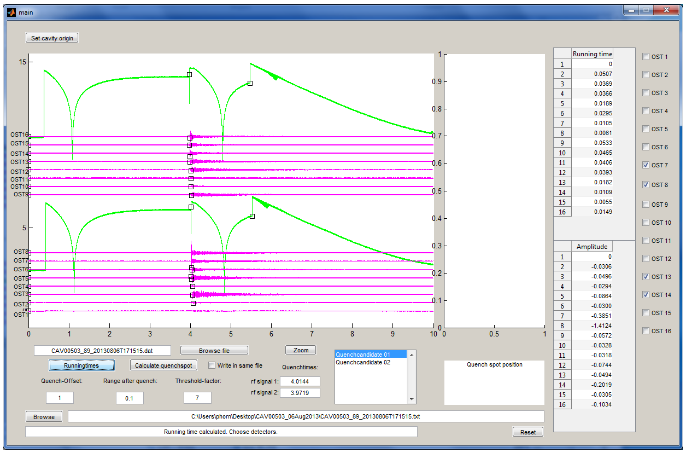


Figure 11: Quench detection with offset=1

The "Range after quench" determines the maximal running time the program is looking for the arrival time of the wave at the detectors. If it is too small the program could miss important signals. A value too high could led to picking a signal from the next quench. In addition a quench is not displayed, if the quench time plus the range exceeds the 10 second border.
The most important function is the "Threshold-factor". The picking of the arrival time of each detector depends on this value. Afterwards the program shifts this point to the nearest minimum, which provides a very accurate arrival time, like shown in the enlarged figure 12. The higher the value, the higher the amplitude has to be to be picked. So the number of pickable detectors also depend on this factor.

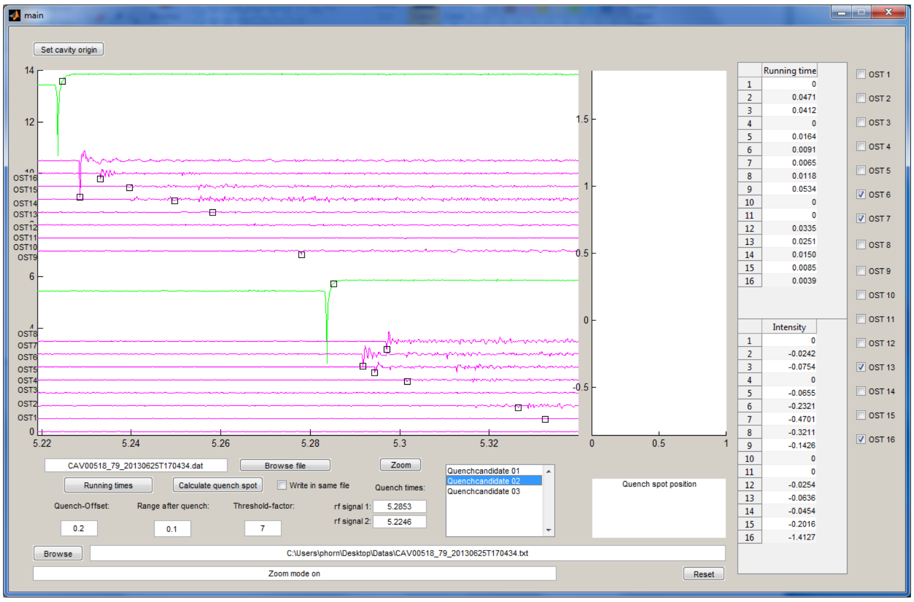
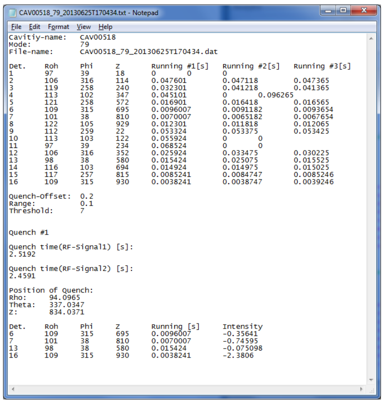


Figure 12: Zoom of Picking of arrival times

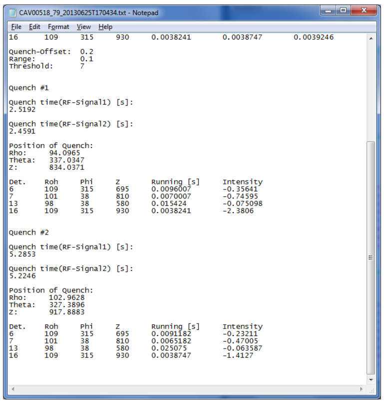
These edit fields can be changed by the user at any time and the standard settings, which are used at the start of the program, are accessible in "main.m" under "main-OpeningFcn".

4.4 Output

Immediately after pressing "Running times" the program suggests a possible destination of a text file next to "Browse" to save the data, which can be changed by the user. After the calculation of the quench spot the program creates the file shown in figure 13. It



```
Cavitiy-name:   CAV00518
Mode:           79
File-name:      CAV00518_79_20130625T170434.dat

Det.    Roh     Phi     Z       Running #1[s]   Running #2[s]   Running #3[s]
1       97      39      18      0       0       0
2       106     316     114     0.047601        0.047118        0.047365
3       119     258     240     0.032301        0.041218        0.041365
4       113     102     347     0.045101        0       0.096265
5       121     258     572     0.016901        0.016418        0.016565
6       109     315     695     0.0096007       0.0091182       0.0093654
7       101     38      810     0.0070007       0.0065182       0.0067654
8       122     105     929     0.012301        0.011818        0.012065
9       112     259     22      0.053324        0.053375        0.053425
10      113     103     122     0.055924        0       0
11      97      39      234     0.068524        0       0
12      106     316     352     0.025924        0.033475        0.030225
13      98      38      580     0.015424        0.025075        0.015525
14      116     103     694     0.014924        0.014975        0.015025
15      117     257     815     0.0085241       0.0084747       0.0085246
16      109     315     930     0.0038241       0.0038747       0.0039246

Quench-Offset:  0.2
Range:          0.1
Threshold:      7


Quench #1

Quench time(RF-Signal1) [s]:
2.5192

Quench time(RF-Signal2) [s]:
2.4591

Position of Quench:
Rho:      94.0965
Theta:    337.0347
Z:        834.0371

Det.    Roh     Phi     Z       Running [s]     Intensity
6       109     315     695     0.0096007       -0.35641
7       101     38      810     0.0070007       -0.74595
13      98      38      580     0.015424        -0.075098
16      109     315     930     0.0038241       -2.3806
```



```
16      109     315     930     0.0038241       0.0038747       0.0039246

Quench-Offset:  0.2
Range:          0.1
Threshold:      7


Quench #1

Quench time(RF-Signal1) [s]:
2.5192

Quench time(RF-Signal2) [s]:
2.4591

Position of Quench:
Rho:      94.0965
Theta:    337.0347
Z:        834.0371

Det.    Roh     Phi     Z       Running [s]     Intensity
6       109     315     695     0.0096007       -0.35641
7       101     38      810     0.0070007       -0.74595
13      98      38      580     0.015424        -0.075098
16      109     315     930     0.0038241       -2.3806


Quench #2

Quench time(RF-Signal1) [s]:
5.2853

Quench time(RF-Signal2) [s]:
5.2246

Position of Quench:
Rho:      102.9628
Theta:    327.3896
Z:        917.8883

Det.    Roh     Phi     Z       Running [s]     Intensity
6       109     315     695     0.0091182       -0.23211
7       101     38      810     0.0065182       -0.47005
13      98      38      580     0.025075        -0.063587
16      109     315     930     0.0038747       -1.4127
```

Figure 13: Output file (one quench)   Figure 14: Output file (two quenches)

consists of the name of the cavity, file and the mode. Below all detectors, their positions and the running times for every quench are visible. The values of the three edit field are also mentioned. Afterwards the quench times, the used detectors, their positions and the associated runnings times and amplitudes follow. As long as the "Write in same file" is not checked, this file is overwritten with every new calculation. Otherwise the next one is added to the file like shown in figure 14.
In figure 15 an example of a calculated quench spot can be seen. The cylindrical coordinates are shown below the plot of the cavity. In this plot the quench is represented as a red dot and there are 16 red square shaped detectors around it. The ones used by the program to calculate the quench spot have a little black square inside the red one as marker. During the calculations and in the plot the origin of the coordinate system is located at the flange of the cavity near the power coupler, but the output file and the display of the quench spot are oriented to the origin at the equator of the first cell. This is the common cavity coordinate system [4]. The difference between flange and the first cell is 163.3 mm.

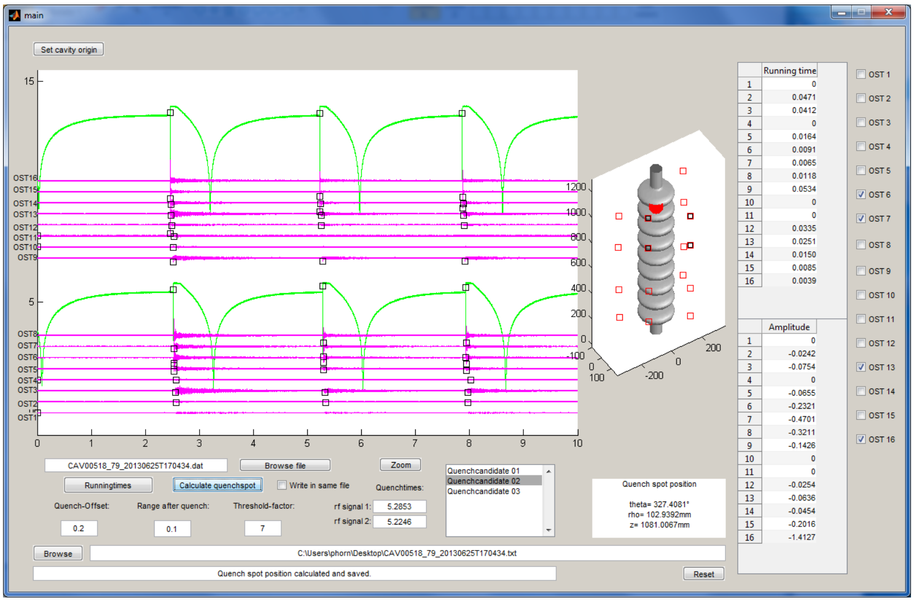


Figure 15: Quench spot calculation

5 Conclusion

Second sound is only at the beginning of long and promising way of research. There are already plans about connecting the t-mapping and the second sound system to get a more precise result. There is still a lot of room for improvement, for instance the measurement of the OST coordinates and the algorithm for the quench spot detection. Almost all changes and new parts of the code are in the appendix. A lot of procedures could be done faster and with a lesser amount of files. The program is now easier to use, to change and the user needs less knowledge of the positions of the detectors. With the possibility to evaluate more quenches, more data of a measurement can be of use and in the end there can be more statistics. Also the output has been improved. Everything is visible at the first sight and a save file is created automatically. Even with all these new features and a better commented structure the size of the code shrank from 250 kb to 150 kb.

All in all I am very glad I was given the opportunity to participate at this program. Both the work at the hardware and the programing was interesting and I hope I at least helped a little bit with the improvement of the program. I want to thank my supervisor Ricarda Laasch for the support and guidance. It was always possible to ask her, if there were any problems. In addition I want to thank Yegor Tamashevich, Marc Wenskat, Aliaksandr Navitski and Professor Elsen for their advices and ideas.

6 Attachment

```matlab
function out=loadfile (handles)

%change out to 1, if error occures
out=0;

%get variables from edit
quench_offset=str2double(get(handles.edt_queoff,'String'));
range=str2double(get(handles.edt_range,'String'));
thresh_fac=str2double(get(handles.edt_thresh,'String'));

%load data from file
file=[getappdata(0,'path') get(handles.edt_filerf,'String')];
if strcmp(get(handles.edt_filerf,'String'), 'file')
    set(handles.edt_status, 'String', 'Please choose file.')
    out=1;
    return
end
set(handles.edt_save, 'String', [file(1:end-4) '.txt'])
data=dlmread(file, ' ', 2, 0);

%shift average of signals to zero (except time)
for i=2:21
    a=sum(data(:,i))/length(data(:,i));
    data(:,i)=data(:,i)-a;
end

%separate data in time, two rf signals, two datasets
t=data(:,1);
rf1=data(:,2);
s1=data(:,3:10);
rf2=data(:,12);
s2=data(:,13:20);

%plot rf signals
hold(handles.axes1, 'on')
plot(handles.axes1, t, rf1+5, 'g');
plot(handles.axes1, t, rf2+13, 'g');

%plot signals
for i=1:8
    plot(handles.axes1, t, s1(:,i)+(i-1)*0.5, 'm');
    plot(handles.axes1, t, s2(:,i)+7+(i-1)*0.5, 'm');
end

%calculate quenchtimes
qt1 = quenchtime(t,rf1,quench_offset);
qt2 = quenchtime(t,rf2,quench_offset);

%disp error, if no quench detected
if isempty(qt1) || isempty(qt2)
    set(handles.edt_status, 'String', 'No Quench detected. Please change quench offset.')
    out=1;
    return
end

% get time and amplitude of arrival of signal for each detector and quenchtime
[time1,ampl1,out1]=ost_signal(t,s1,qt1,range,thresh_fac);
[time2,ampl2,out2]=ost_signal(t,s2,qt2,range,thresh_fac);
```

```
%delete last quench, if range too high
if out1==1 || out2==1
    set(handles.edt_status, 'String', 'Last quench close to end of data. Change range.')
    number=length(qt1);
    qt1(number)=[];qt2(number)=[];
    time1(:,number)=[];time2(:,number)=[];
    ampl1(:,number)=[];ampl2(:,number)=[];
end

%calculate runningtime
running1=time1-[qt1;qt1;qt1;qt1;qt1;qt1;qt1;qt1];
running2=time2-[qt2;qt2;qt2;qt2;qt2;qt2;qt2;qt2];

%set unfound values to zero
running1(running1<0)=0;
running2(running2<0)=0;

%combine signals or disp error, if not same number of quenches
if size(running1,2) ~= size(running2,2)
    set(handles.edt_status, 'String', 'Different number of quenches found. Please change
quench offset.')
    out=1;
    return
end
running=[running1;running2];
ampl=[ampl1;ampl2];
num_quench=length(qt1);

%check for best detectors
check=zeros(16,num_quench);
for i=1:num_quench
    m=0;num_max=2;
    while m==0
        [check(:,i),m]=choose_det(ampl(:,i),num_max);
        num_max=num_max+1;
    end
end

%disp results
for i=1:num_quench
    if i<10
        set(handles.listbox2, 'String',[get(handles.listbox2, 'String'); 'Quenchcandidate 0'
num2str(i)])
    else
        set(handles.listbox2, 'String',[get(handles.listbox2, 'String'); 'Quenchcandidate '
num2str(i)])
    end
end
set(handles.tbl_run,'Data',running(:,1))
set(handles.tbl_amp,'Data',ampl(:,1))
set(handles.edt_quenchtime1,'String', num2str(qt1(1)))
set(handles.edt_quenchtime2,'String', num2str(qt2(1)))
for i=1:16
   set(eval(['handles.ost' num2str(i)]),'Value',check(i,1))
end

%save results
setappdata(0,'running', running)
```

```
setappdata(0,'amplitude', ampl)
setappdata(0,'quenchtime', [qt1;qt2])
setappdata(0,'check',check)

%plot positions of arrival time
plot(handles.axes1, qt1, rf1(int64(qt1*10000))+5, 'sk');
plot(handles.axes1, qt2, rf2(int64(qt2*10000))+13, 'sk');
for i=1:8
    plot(handles.axes1, time1(i,:), ampl1(i,:)+(i-1)*0.5, 'sk');
    plot(handles.axes1, time2(i,:), ampl2(i,:)+7+(i-1)*0.5, 'sk');
end

hold(handles.axes1, 'off')
```

Published with MATLAB® R2012b

```
 function qt = quenchtime(t,rf,quench_offset)
%ANALYSIS Summary of this function goes here
%   Detailed explanation goes here
qt=[];
i=1;

rfmax=max(rf);
rfmin=min(rf);
while i<=99500

    %use only if almost vertical rise and close to maximum
    if ((sum(rf(i+199:-1:i+150))/50-0.1) > (sum(rf(i+149:-1:i+100))/50)) &&
rf(i+250)+quench_offset>rfmax
        qtc=(i-1)*0.0001+0.015;           %roughly quenchtime

        %rf values of beginning and end of jump in signal and half maximum
        amin=rf(round(qtc*10000-50));
        amax=rf(round(qtc*10000+50));
        astart=abs(amax-amin)/2;

        %time and rf signal around the quenchtime
        tf=t(round(qtc*10000-50):round(qtc*10000+50));
        rff=rf(round(qtc*10000-50):round(qtc*10000+50));

        %fitting of errorfunction
        s=fitoptions('Method','NonlinearLeastSquares','Lower',[astart-2,0,qtc-0.01,rfmin-
1],...

'Upper',[astart+2,1000000,qtc+0.01,rfmax+1],'Startpoint',[astart,10,qtc,(rfmin+rfmax)/2]);
        f=fittype('a*erf(b*(x-c))+d');
        [data]=fit(tf,rff,f,s);
        c=data.c;                         %x-value of fit-function

        qt=[qt c];                    %take data
        i=i+990;                      %jump over quench
    end

i=i+10;
end
```

Published with MATLAB® R2012b

```
function [time,ampl,out] = ost_signal(t,signal,quench,range,thresh_fac)

%change out to 1, if range too high
out=0;

time=zeros(8,length(quench));
ampl=zeros(8,length(quench));

for i=1:8

    %calculate average amplitude of signal
    thresh=-(sum(abs(signal(:,i)))/length(signal(:,i)))*thresh_fac;

    for n=1:length(quench)

        %select datarange around quench
        lower=int64(round(quench(n)*10000));
        upper=int64(round(quench(n)*10000+range*10000));
        if upper >= 100000
            out=1;
            break
        end
        t_part=t(lower:upper);
        signal_part=signal(lower:upper,:);

        %stops if amplitude gets too high(wavearrival)
        j=1;
        while signal_part(j,i)>thresh

            %stop if end of vector is reached
            if length(signal_part) <= j, j=-1; break, end
            j=j+1;
        end

        if j ~=-1

            %shift point to minimum
            while signal_part(j,i)>signal_part(j+1,i)
                j=j+1;
                if length(signal_part) <= j, break, end
            end

            %take datapoint
            time(i,n)=t_part(j);
            ampl(i,n)=signal_part(j,i);
        end
    end
end
```

Published with MATLAB® R2012b

```
function [check,m]=choose_det(ampl,num_max)

%variables for checking, if both maxima are on the same pole
m=1;
a=zeros(1,4);

%get two highest signals
[~,amplsort]=sort(ampl);
check=zeros(16,1);
for j=[1,num_max]
    amplrow=ampl;
    switch amplsort(j)

        %choose the right pole
        case {1,7,11,13}
            a(1)=a(1)+1;

            %delete every other entry
            amplrow(4)=0;amplrow(8)=0;amplrow(10)=0;amplrow(14)=0;
            amplrow(3)=0;amplrow(5)=0;amplrow(9)=0;amplrow(15)=0;
            amplrow(2)=0;amplrow(6)=0;amplrow(12)=0;amplrow(16)=0;

            %get candidates of detectors
            [cand,candind]=sort(amplrow);

            %only add detector, if runningtime exsists
            for n=1:2
                if cand(n)~=0,check(candind(n))=1;end
            end
        case {4,8,10,14}
            a(2)=a(2)+1;

            %delete every other entry
            amplrow(1)=0;amplrow(7)=0;amplrow(11)=0;amplrow(13)=0;
            amplrow(3)=0;amplrow(5)=0;amplrow(9)=0;amplrow(15)=0;
            amplrow(2)=0;amplrow(6)=0;amplrow(12)=0;amplrow(16)=0;

            %get candidates of detectors
            [cand,candind]=sort(amplrow);

            %only add detector, if runningtime exists
            for n=1:2
                if cand(n)~=0,check(candind(n))=1;end
            end
        case {3,5,9,15}
            a(3)=a(3)+1;

            %delete every other entry
            amplrow(1)=0;amplrow(7)=0;amplrow(11)=0;amplrow(13)=0;
            amplrow(4)=0;amplrow(8)=0;amplrow(10)=0;amplrow(14)=0;
            amplrow(2)=0;amplrow(6)=0;amplrow(12)=0;amplrow(16)=0;

            %get candidates of detectors
            [cand,candind]=sort(amplrow);

            %only add detector, if runningtime exists
            for n=1:2
                if cand(n)~=0,check(candind(n))=1;end
```

```
            end
        case {2,6,12,16}
            a(4)=a(4)+1;

            %delete every other entry
            amplrow(1)=0;amplrow(7)=0;amplrow(11)=0;amplrow(13)=0;
            amplrow(4)=0;amplrow(8)=0;amplrow(10)=0;amplrow(14)=0;
            amplrow(3)=0;amplrow(5)=0;amplrow(9)=0;amplrow(15)=0;

            %get candidates of detectors
            [cand,candind]=sort(amplrow);

            %only add detector, if runningtime exists
            for n=1:2
                if cand(n)~=0,check(candind(n))=1;end
            end
    end

    %perform the program again, if the two detectors with highest amplitude are on same pole
    if max(a)==2, m=0;end
end
```

Published with MATLAB® R2012b

```
function out=findquench(handles)
% find spot on cavity surface

%change out to 1, if error occures
out=0;

%load place of detectors and cavity and set current axes
set(gcf, 'CurrentAxes', handles.axes2)
setdetectors;
teslashape;

%load variables
running=getappdata(0,'running');
check=getappdata(0,'check');
num=get(handles.listbox2,'Value');

%load detector positions
detpos=getappdata(0,'detectorposition');

% create coordinate parameters
z0=163.3; % equator height of cell #5
r0=103.3; % equatorial radius of cavity

%transform parameters in cart coord
detposcart=detpos;
for i=1:16

[detposcart(i,1),detposcart(i,2),detposcart(i,3)]=pol2cart((detpos(i,2)/360)*2*pi,detpos(i,1)+
r0,detpos(i,3)+z0);
end

% velocity of 2nd sound in He in m/s
v=19.9;

%used detectors and fitting runningtimes
det_nr=find(check(:,num)==1);
run=running(check(:,num)==1,num);

%error, if detector with runningtime=0 is choosen
if isempty(run(run==0)) ~= 1
    set(handles.edt_status, 'String', 'Choose a detector with runningtime~=0')
    out=1;
    set(gcf, 'CurrentAxes', handles.axes1)
    cla(handles.axes2)
    return
end


% get width of propagation
r=run*v*1000;
setappdata(0,'r',r');

%get detector positions
p=zeros(3,length(det_nr));
for i=1:length(det_nr)
    p(:,i)=detposcart(det_nr(i),:)';
    plot3(p(1,i),p(2,i),p(3,i),'sy','MarkerEdgeColor','k','MarkerSize',5)
end
```

```
%calculate distance-function depending on number of selected detectors
sigma=0.1;
switch length(det_nr)
    case 2
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2;
    case 3
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2;
    case 4
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2+((norm(x-p(:,4))-r(4))/sigma)^2;
    case 5
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2+((norm(x-p(:,4))-r(4))/sigma)^2+((norm(x-p(:,5))-r(5))/sigma)^2;
    case 6
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2+((norm(x-p(:,4))-r(4))/sigma)^2+((norm(x-p(:,5))-
r(5))/sigma)^2+((norm(x-p(:,6))-r(6))/sigma)^2;
    case 7
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2+((norm(x-p(:,4))-r(4))/sigma)^2+((norm(x-p(:,5))-
r(5))/sigma)^2+((norm(x-p(:,6))-r(6))/sigma)^2+((norm(x-p(:,7))-r(7))/sigma)^2;
    case 8
        f=@(x) ((norm(x-p(:,1))-r(1))/sigma)^2+((norm(x-p(:,2))-r(2))/sigma)^2+((norm(x-
p(:,3))-r(3))/sigma)^2+((norm(x-p(:,4))-r(4))/sigma)^2+((norm(x-p(:,5))-
r(5))/sigma)^2+((norm(x-p(:,6))-r(6))/sigma)^2+((norm(x-p(:,7))-r(7))/sigma)^2+((norm(x-
p(:,8))-r(8))/sigma)^2;
end

x0=[0 0 0]';
[x,~]=fminsearch(f,x0);
x0=x;

%calculate and save g function
R=61.3;r=42;
g=@(t,p) [(R+r*cos(p))*cos(t);(R+r*cos(p))*sin(t);r*sin(p)+z0];
setappdata(0,'g',g)

%calculate minimum consiering the shape of the cavity
[x,~,~,~]=fmincon(f,x0,[],[],[],[],[],[],@neben,optimset('Algorithm','active-set'));

%plot location of quench
plot3(x(1),x(2),x(3),'-
r.','MarkerSize',48,'LineWidth',8,'MarkerEdgeColor','r','MarkerFaceColor',[.49 1 .63])

%calculate and show location of quench
[theta,rho,z]=cart2pol(x(1),x(2),x(3));
thetar=mod(theta+2*pi,2*pi)/(2*pi)*360;

view(thetar+90,45)
setappdata(0,'rho',rho);
setappdata(0,'theta',thetar);
setappdata(0,'z',z-z0);
set(handles.quench_pos, 'String', sprintf(['Quench spot #' num2str(num) ' position\n\ntheta= '
num2str(thetar) '°\nrho= ' num2str(rho) 'mm\nz= ' num2str(z-z0) 'mm']))

function[c,ceq]=neben(x)
c=[];
```

```
ceq=((tesladata(x(3))*cos(atan2(x(2),x(1))))^2+(tesladata(x(3))*sin(atan2(x(2),x(1))))^2+x(3)^
2)^(.5)-norm(x);
axis equal;
```

Published with MATLAB® R2012b

```
function writefile(handles)

%get variables
running=getappdata(0,'running');
ampl=getappdata(0,'amplitude');
qt=getappdata(0,'quenchtime');

%get variables from edit
quench_offset=str2double(get(handles.edt_queoff,'String'));
range=str2double(get(handles.edt_range,'String'));
thresh_fac=str2double(get(handles.edt_thresh,'String'));

%get used quench and detectors and detector positions
check=getappdata(0,'check');
num=get(handles.listbox2,'Value');
used=find(check(:,num)==1);
detpos=getappdata(0,'detectorposition');

%get file and destination
file=get(handles.edt_filerf,'String');
dest=get(handles.edt_save, 'String');

%get quench position
rho=getappdata(0,'rho');
theta=getappdata(0,'theta');
z=getappdata(0,'z');

%delete, if checkbox is not activated
if get(handles.savefile,'Value')==0
    if exist(dest,'file')==2,delete(dest),end
end

%prepare head of running times
head='Det.    Roh     Phi     Z       ';
for i=1:size(qt,2)
    if i<10
        head=[head 'Running #' num2str(i) '[s]   '];
    else
        head=[head 'Running #' num2str(i) '[s]  '];
    end
end

%write head(cav, mode, filename), all running times and edit fields, if not already there
if exist(dest,'file')~=2
    dlmwrite(dest,['Cavitiy-name:   ' file(1:8)],'newline','pc','delimiter','')
    dlmwrite(dest,['Mode:           ' file(10:11)],'-append','newline','pc','delimiter','')
    dlmwrite(dest,['File-name:      ' file],'-append','newline','pc','delimiter','')
    dlmwrite(dest,' ','-append','newline','pc','delimiter','')
    dlmwrite(dest,head,'-append','newline','pc','delimiter','')
    dlmwrite(dest,[linspace(1,16,16)' detpos running],'-
append','newline','pc','delimiter','\t')
    dlmwrite(dest,' ','-append','newline','pc','delimiter','')
    dlmwrite(dest,['Quench-Offset:  ' num2str(quench_offset)],'-
append','newline','pc','delimiter','')
    dlmwrite(dest,['Range:          ' num2str(range)],'-append','newline','pc','delimiter','')
    dlmwrite(dest,['Threshold:      ' num2str(thresh_fac)],'-
append','newline','pc','delimiter','')
    dlmwrite(dest,' ','-append','newline','pc','delimiter','')
```

```
    dlmwrite(dest,' ','-append','newline','pc','delimiter','')
end

%write quenchtime and position
dlmwrite(dest,['Quench #' num2str(num)],'-append','newline','pc','delimiter','')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')
dlmwrite(dest,'Quench time(RF-Signal1) [s]:','-append','newline','pc','delimiter','')
dlmwrite(dest,qt(1,num),'-append','newline','pc','delimiter','')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')
dlmwrite(dest,'Quench time(RF-Signal2) [s]:','-append','newline','pc','delimiter','')
dlmwrite(dest,qt(2,num),'-append','newline','pc','delimiter','')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')
dlmwrite(dest,'Position of Quench:','-append','newline','pc','delimiter','')
dlmwrite(dest,['Rho:     ' num2str(rho)],'-append','newline','pc','delimiter','')
dlmwrite(dest,['Theta:   ' num2str(theta)],'-append','newline','pc','delimiter','')
dlmwrite(dest,['Z:       ' num2str(z)],'-append','newline','pc','delimiter','')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')

%write used detecors, suiting running time and amplitude
dlmwrite(dest,'Det.    Roh     Phi     Z       Running [s]     Intensity',...
    '-append','newline','pc','delimiter','')
dlmwrite(dest,[used detpos(used,:) running(used,num) ampl(used,num)],...
    '-append','newline','pc','delimiter','\t')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')
dlmwrite(dest,' ','-append','newline','pc','delimiter','')
```

Published with MATLAB® R2012b

References

[1] R. Brinkmann et al. (eds.), TESLA Technical Design Report, DESY 2001/011 (March 2001). http://tesla.desy.de

[2] F. Schlander, Dissertation, University of Hamburg (2012).

[3] R. J. Donnelly, Phys. Today, 34-39 (Oct. 2009).

[4] S. Aderhold et al, ILC-HiGrade-Report-2011-003 (March 2012).

[5] F. Schlander, E. Elsen, ILC-HiGrade-Report-2010-007 (Oct. 2010)