RHK Technology
Application Note:

NC-AFM Amplitude Calibration with RHK R9 and a MATLAB Script

Matthias Temmen, Jannis Lübbe, Michael Reichling
Fachbereich Physik, Universität Osnabrück
49076 Osnabrück, Germany

Introduction

Spatial resolution, signal-to-noise ratio and the detection sensitivity for tip-sample interaction forces in non-contact atomic force microscopy (NC-AFM) have increased significantly over the past years. For several applications, it is crucial to precisely know the oscillation amplitude of the cantilever. For instance, force-distance curves derived from frequency shift data by analytic [1] or numeric [2] methods require the knowledge of the oscillation amplitude to provide accurate results. The method of calibrating the amplitude introduced here [3] is based on the method suggested by Simon et al. [4], however, we introduce a fully automated procedure to achieve highly reproducible calibration factors. We implemented a simple to use MATLAB script for remote control of the RHK R9. As the R9 offers a large variety of commands to control nearly all functions remotely, we need only 25 lines of MATLAB script for the routine automatically performing all steps for the amplitude calibration. The communication is very flexible through Ethernet (UDP and TCP protocol) so that multiple connections at a time are allowed.

2015.02.20

Calibration Method

In the optical beam-deflection detections scheme, the cantilever oscillation yields a nearly harmonically oscillating voltage V_Z at the output of the displacement detector preamplifier as illustrated in Figure 1. For the calibration, we relate the voltage amplitude V_A of the oscillating voltage $V_Z = V_A \sin(2\pi f_0 t + \varphi)$ to the amplitude A of the mechanical cantilever oscillation with frequency f_0 using a procedure where the tip–surface interaction is kept constant at a certain value while A is varied [4]. As often the cantilever is inclined by an angle θ towards the sample surface, we distinguish between the oscillation amplitude of the cantilever A and the component of the oscillation amplitude perpendicular to the sample surface A_Z. These two quantities are related by

$$A = A_Z \cos\theta$$

A deflection of the cantilever by an angle $\Delta\theta$ changes the angle of the reflected laser beam by $2\Delta\theta$, resulting in a displacement of the laser spot on the position sensitive device (PSD). The displacement on the PSD results in a variation of the photocurrent ΔI_Z from the PSD quadrants, that is converted to a voltage signal V_Z by the preamplifier. The calibration establishes the relation between the amplitude A of the oscillating cantilever and the voltage amplitude V_A

$$A = S \times V_A$$

where S is the sensitivity or calibration factor. Here, we obtain S in a non-contact measurement without any knowledge of the details of the detection system except the tilt angle θ. To accomplish this, the normalized frequency shift γ [5] derived from the frequency shift Δf

$$\gamma(z_{ts}, A) = \frac{k_0 A^{\frac{3}{2}}}{f_0} \Delta f(z_{ts}, f_0, k_0, A)$$

is kept constant during the entire calibration procedure to maintain a certain level of tip–surface interaction independent of the cantilever oscillation amplitude (z_{ts} tip-sample distance, k_0 cantilever stiffness). Note that this concept is valid only for amplitudes larger than a critical value, that is typically 1 nm [6].

During the calibration procedure, the topography feedback is turned on and A is stepwise increased or decreased by a variation of the amplitude feedback setpoint defining V_A while the normalized frequency shift γ is kept constant by appropriately tracking the frequency-shift setpoint Δf_{set}.

This causes the topography feedback to readjust the z-piezo position z_P (see Figure 1) that is recorded as a function of the set oscillation amplitude voltage V_A. A MATLAB script (The MathWorks, Inc., Natick, MA, USA) is used to control the setpoints of V_A and Δf and to record the corresponding z_P values. Initially, the tip is approached to the surface and stabilized at a distance in the long-range interaction regime by the choice of an appropriate frequency shift setpoint Δf_{set}. After waiting for a while to reduce piezo creep effects, the piezo position z_P is recorded as a function of the oscillation voltage amplitude V_A, while $V_A^{3/2}\Delta f$ is kept constant by adjusting the Δf setpoint accordingly. Typical parameters and results are shown in Figure 2.

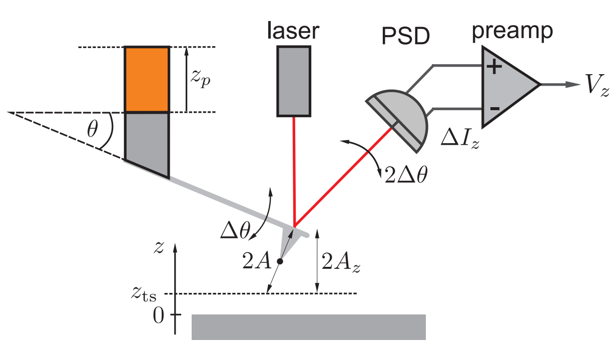

Figure 1 Schematic representation of the signal path in an NC-AFM system based on the optical beam-deflection scheme [3]. The relation between the amplitude A of the oscillating cantilever and its component A_Z perpendicular to the surface is illustrated as well as the periodic cantilever deflection $\Delta\theta$, the PSD output current difference ΔI_Z, and the preamplifier output voltage $V_Z = V_A \sin(2\pi f_0 t + \varphi)$. The quantity z_{ts} represents the tip–sample distance in the lower turning point of the cantilever oscillation while zp denotes the position of the piezo carrying the cantilever.

Calibration Factor

The calibration factor S is obtained from the slope in the plot z_P versus V_A while considering the correction for the tilt angle θ.

$$S = \frac{\Delta z_P}{\Delta V_A} \cos\theta$$

To compensate for thermal drift as well as residual piezo creep, this procedure is performed with increasing and decreaing oscillation amplitude. Respective measurements are denoted as “up” and “down” in the plot of Figure 2 for a measurement strongly influenced by thermal drift. From the measurements shown Figure 2, a calibration factor of

$$S = (166.0 \pm 23.2)\ \text{nm/V}$$

is derived by calculating the mean slope from the up and down measurements. If thermal drift is negligible, the curves for up and down measurements coincide within the experimental error and directly yield the result.

RHK Technology

Application Note:

NC-AFM Amplitude Calibration with RHK R9 and a MATLAB Script

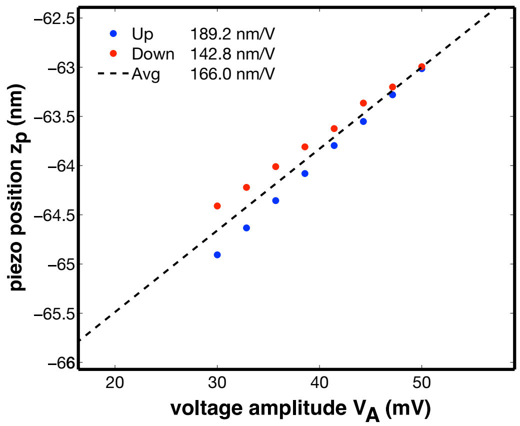

Figure 2 Typical result for an amplitude calibration [3] taken within 37 s for a commercial cantilever of NCH type [7]. Blue and red dots represent measured positions for up and down movement of the cantilever. The curves differ in their slope due to thermal drift that is compensated by extracting the calibration factor *S* from the slope of the mean straight line.

MATLAB Script

The procedure is fully automated by the MATLAB script, however, the script requires some configuration before the first start. As a prerequisite, there must be a "Network Stream Hardware Item" connected to the "Z PI Output" and an "External Command Receiver" in the R9 software dashboard. The settings of the "Network Stream Hardware Item" have to be adjusted so that they fit the present network architecture. The values for the UDP port of the stream item, the TCP port of the "External Command Receiver" and the IP address of the PC where the R9 software is running are then entered in the "Network Properties" section of the MATLAB script.

Before running the script, the tip is positioned in the attractive interaction regime of the sample surface by running the standard approach procedure. The amplitude setpoint should be chosen so that the expected oscillation amplitude is about 20 nm. The Δf setpoint should then be raised to a rather high value (typically -20 Hz). Finally, the MATLAB script can be started. Performing the routine typically takes 37 seconds and the calibration factor *S* is shown in the popup window.

text	V_A	γ	S	Δf
script	amplitude	norm_df	ampcal	df

Table Correspondence of MATLAB variable names and symbols used in this application note.

There are two helper functions inside the MATLAB script provided by RHK. The "get_R9_packet" telling the R9 to send data and parse the data afterwards and "jUDP", that is used to manage the network communication. The source code and the helper scripts can be downloaded from http://www.rhk-tech.com

References

1. Sader, J. E. & Jarvis, S. P. Accurate formulas for interaction force and energy in frequency modulation force spectroscopy. Appl. Phys. Lett. 84, 1801–1803 (2004).

2. Giessibl, F. J., A direct method to calculate tip–sample forces from frequency shifts in frequency-modulation atomic force microscopy, Applied Physics Letters 78, 123-125 (2001)

3. Luebbe, J. et al. Thermal noise limit for ultra-high vacuum noncontact atomic force microscopy. Beilstein Journal of Nanotechnology 4, 32–44 (2013).

4. Simon, G. H., Heyde, M. and Rust, H.-P. Recipes for cantilever parameter determination in dynamic force spectroscopy: spring constant and amplitude. Nanotechnology 18, 255503 (2007).

5. Giessibl, F. and Bielefeldt, H. Physical interpretation of frequency-modulation atomic force microscopy. Phys. Rev. B 61, 9968–9971 (2000).1476-78 (1999).

6. Giessibl, F. Forces and frequency shifts in atomic-resolution dynamic-force microscopy. Phys. Rev. B 56, 16010–16015 (1997).

7. The cantilever was supplied by Nanoworld Services GmbH. f0 = 311,148 kHz, Q0 = 30810.

RHK Technology Inc.
1050 East Maple Road
Troy, Michigan 48083 USA
T: 248.577.5426 F: 248.577.5433
RHK-Tech.com

```matlab
%AMPLITUDECALIBRATION_R9     Calibration of the cantilever oscillation amplitude
%   - Only works in constant amplitude mode and with topography feedback
%     turned on.
%   - Change the network properties in this file to your needs.
%   - Connect a Network Stream Hardware Item to the Z PI Output. Change the
%     IP address to the PC where the Matlab script is running (can be
%     127.0.0.1) and choose a free UDP port.
%   - Add an 'External Command Receiver' to the IHDL workspace. Set a free
%     port in the item - this is the TCP_port_R9.
%   - Ensure your topography feedback loop is set to a gentle value.
%   - Set an amplitude which is expectedly larger than 20 nm. (Repeat the
%     calibration if it was to small)
%   - Set a delta f setpoint which is relatively high (e.g. -20 Hz). If
%     this is not possible, choose the highest stable setpoint.
%   - Repeat the calibration if the two curves do not align (this can be an
%     issue of large drift or piezo creep)
%   - The returned value is the calibration factor in units of nanometers per volt.
%
% Version 1.0, 2014/07/22
% Author: Matthias Temmen, NanoScience research group
% Universitaet Osnabrueck, Germany
% http://reichling.physik.uos.de
function ampcal=AmplitudeCalibration_R9()
%% Initialization and clean-up
clear all
%Network properties (CHANGE TO YOUR NEEDS)
IP_Address_R9_PC        = '127.0.0.1';      %IP-Address of the PC running the R9 software
TCP_port_R9             = 12600;            %Port of "External Command Receiver"
Network_Stream_UDP_Port = 12501;            %Port of the "Network Stream Item"
%Find any open TCP or UDP ports and close them
in = instrfind;
if ~isempty(in)
    fclose(in);
end
%Establish a new TCP connection to the R9 unit to make Forth calls and set
%parameters, open and flush it
global con;
con = tcpip(IP_Address_R9_PC,TCP_port_R9);
fopen(con);
if (con.BytesAvailable > 0)
    fread(con,con.BytesAvailable);
end
%Read the current delta f and amplitude setpoint with the force of forth
fprintf(con,'CallForthCommand, TopoPI pi-sp ctrl-get fe.');pause(0.1);data = fread(con,con.BytesAvailable); chars = char(data); charcell = cellstr(chars); value = strcat(charcell{:,1});
dfmax       = str2double(value)
fprintf(con,'CallForthCommand, DissPI pi-sp ctrl-get fe.');pause(0.1);data = fread(con,con.BytesAvailable); chars = char(data); charcell = cellstr(chars); value = strcat(charcell{:,1});
ampmin      = str2double(value)
%Change the delta f and amplitude setpoints to x percent (maximum adjustment)
%of the current values returned
percent     = 20;
%Number of measurement points for piezo position vs. amplitude plot
points      = 8;
%Generate the amplitude array
amplitude   = linspace(ampmin,ampmin+percent./100.*ampmin,points);
%Calculate the normalized frequency shift
norm_df     = dfmax .* min(amplitude).^(3./2);
df          = norm_df./(amplitude.^(3./2));
%Initialize the two data arrays
data1       = NaN(1,points);
data2       = NaN(1,points);
%% Data acquisition
%Walk through the array and gather data, first upstairs
for i=1:points
    while isnan(data1(i)) == 1
        %Use the external script for getting data
        data = get_r9_packet(Network_Stream_UDP_Port);
```

```
        data1(i) = mean(data.data);
        if isnan(data1(i)) == 1
            disp(['NaN detected by no. ' num2str(i) ' upstairs']);
        end
    end
    disp(['Measurement of datapoint no. ' num2str(i) ' upstairs     ' num2str(data1(i))]);
    %Set amplitude and df with function "dfampset"
    if i<points
        dfampset(df(i),df(i+1),amplitude(i),amplitude(i+1));
    end
    pause(0.5);
end

% Then downstairs
for j=1:points
i = points+1-j;
    while isnan(data2(i)) == 1
        data = get_r9_packet(Network_Stream_UDP_Port);
        data2(i) = mean(data.data);
        if isnan(data2(i)) == 1
            disp(['NaN detected by no. ' num2str(i) ' downstairs']);
        end
    end
    disp(['Measurement of datapoint no. ' num2str(i) ' downstairs   ' num2str(data2(i))]);
    if i>1
        dfampset(df(i),df(i-1),amplitude(i),amplitude(i-1));
        pause(0.5);
    end

end

%% Data analysis

%R9 sends values in meters, so we better scale it to nanometers
z1 = data1.*1e9;
z2 = data2.*1e9;

%Open figure and plot the two arrays
figure;
plot(amplitude,z1,'.b');
hold on;
plot(amplitude,z2,'.r');
xlabel('voltage amplitude (mV)');
ylabel('piezo position (nm)');

%Fit the mean of the two arrays as a linear function
xplot = linspace(min(amplitude).*0.5,max(amplitude).*1.2,100);
pfit1 = polyfit(amplitude,z1,1);
pfit2 = polyfit(amplitude,z2,1);
yplot = polyval((pfit1+pfit2)./2,xplot);
plot(xplot,yplot,'k--')
legend(sprintf('Up %2.4f nm/V',pfit1(1)),sprintf('Down %2.4f nm/V',pfit2(1)),sprintf('Avg %2.4f nm/V',(pfit1(1)+pfit2(1))./2),'Location','Northwest');

%Close the connection and return the fitted slope as the amplitude
%calibration factor in units of nm/V
disp('Close connection...');
fclose(con);
disp('Finished!');
ampcal = (pfit1(1)+pfit2(1))./2



function dfampset(dfold,dfnew,ampold,ampnew)
global con;
steps   = 4;
dfstep  = (dfnew-dfold)./steps;
ampstep = (ampnew-ampold)./steps;
for j=1:steps
    fprintf(con,'SetHWSubParameter, Z Pi Controller, Set Point, Value, %e',(dfold+dfstep.*j));
    fprintf(con,'SetHWSubParameter, PLL, DissPI Setpoint, Value, %e',(ampold+ampstep.*j));
    pause(0.5)
end
```

RHK Technology

Application Note:

NC-AFM Amplitude Calibration with RHK R9 and a MATLAB Script

RHK Technology Inc.
1050 East Maple Road
Troy, Michigan 48083 USA
T: 248.577.5426 F: 248.577.5433
RHK-Tech.com