Appendix A

Lock-In Technique

The Lock-in amplifier is a laboratory instrument used to detect and acquire very small AC signals that may be down to a few nanovolts. It employs a technique known as phase-sensitive detection (PSD) to intercept the component of the real signal at a specific reference frequency and phase in the presence of an overwhelming noise background. In other words, the noise signals at frequencies other than the reference frequency are filtered. In addition, a versatile lock-in amplifier increased capability can function as an AC Signal Recovery Instrument, a Vector Meter, a Spectrum Analyzer, a Noise Measurement Unit, a Phase Meter and so on. While a lock-in amplifier is powerful measurement equipment, the underlying operation principle is quite simple as well. The following descriptions will elaborate on the basic functions of each component of a lock-in amplifier.

The schematic block diagram of a lock-in amplifier is as displayed in figure A. It chiefly consists of seven functional components: 1) a signal amplifier; 2) an bandpass filter; 3) a reference trigger; 4) a phase shifter; 5) a PSD; 6) a low pass filter; 7) a DC amplifier. The signal to be measured is fed into the input of signal amplifier as indicated in V_{in}. The output V_{out} of the DC amplifier is a DC voltage proportional to V_0, where V_0 is the amplitude of V_{in}. Detailed functions of these components of a lock-in amplifier are described as below.

The signal amplifier is utilized to amplify V_{in} and the included noise by an adjustable-gain, AC-coupled amplifier in order to match it more closely to the optimum input signal range of the PSD. They are usually fitted with high impedance inputs for voltage measurements or low impedance inputs for better noise matching to current sources.

The bandpass filter is to act as a LC filter to improve the performance of the PSD by reducing the bandwidth of the noise voltages from that of the full frequency range of the instrument.

The reference trigger is designed to trigger a synchronized signal with an external or internally reference input both in phase and frequency. Then, the generated reference signal is passed through a phase shifter, which is used to compensate for phase differences that may have been introduced between the signal and reference input before being applied to the PSD.

The PSD is simply a circuit which act as linear multiplier by taking in two voltages as inputs of V_{sa} and V_r and producing an output as the product of V_{sa} * V_r. Actually, this is just a rough description, since the PSD at least consists of three common methods to implement this work. They can be an Ananlog Multiplier, a Digital Switch or a Digital Multiplier.

The low pass filter is to act as a RC filter to remove the AC components from the desired DC output. Practical instrument employ a wide range of RC filter types. Typically, there are two successive filters so that the overall filter can roll off at either 6 dB or 12 dB per octave.

The DC amplifier is just a low-frequency amplifier. The use of this amplifier, in conjunction with the Signal amplifier, allows the unit to handle a range of signal inputs.

In general [1], a fed signal $V_{in} = V_0\cos(\omega_0 t)$ is amplified by Signal amplifier, where $\omega_0 = 2\pi f_0$ and f_0 are the angular and natural frequencies of the signal respectively. Furthermore, only a narrow band of frequencies around f_0 are passed through the Bandpass filter. Besides, the Signal amplifier has a voltage gain G_{ac}, which is determined by the sensitivity setting of the lock-in amplifier. The output of the Signal amplifier becomes $V_{ac}(t) = G_{ac}V_0\cos(\omega_0 t)$, which serves as one of the multiplicands. Also, the other is a voltage $V_r = E_0\cos(\omega_0 t + \varphi)$, which is furnished by the Reference trigger. The output of the PSD then becomes $V_{psd}(t) = G_{ac}V_0E_0\cos(\omega_0 t)\cos(\omega_0 t + \varphi)$. Through adjusting Phase shifter, φ can be set to zero, i.e. that V_r is in-phase with V_{ac}. By using the appropriate trigonometric identify, the V_{psd} can be rewritten by $V_{psd}(t) = 1/2 * G_{ac}V_0E_0[1 + \cos(2\omega_0 t)]$, where the amplitudes of the second harmonic and the DC voltage are both proportional to V_0. Next, by feeding the output signal of the PSD to a low pass filter, the AC information of V_{psd} can be attenuated. Finally, by amplifying the DC voltage of V_{psd} with a DC amplifier, the output voltage of V_{out} is then a DC voltage directly proportional to the amplitude of V_0 by $V_{psd} = 1/2 * G_{dc}\, G_{ac}V_0E_0$, where G_{dc} is the voltage gain of the DC amplifier and V_0 is the value of real signal.

[1] John H. Scofield, American Journal of Physics **62**(2) 129-133(Feb. 1994)

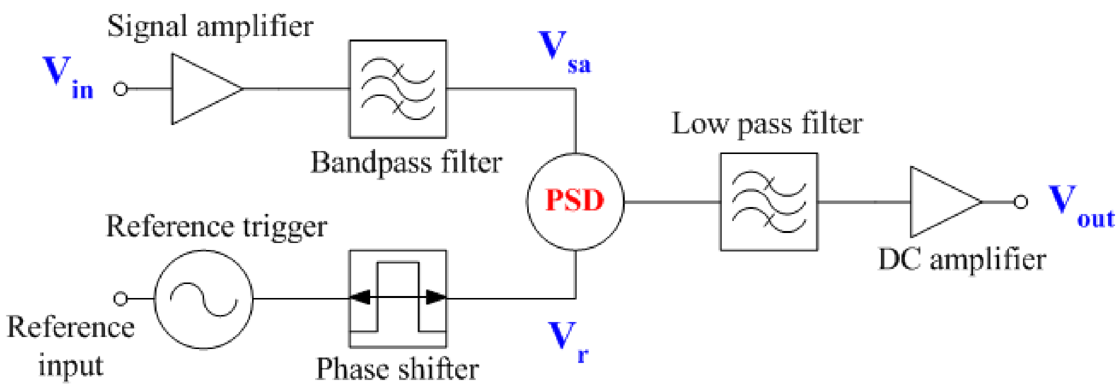

Fig. A Functional block diagram of a lock-in amplifier.

Appendix B

Matlab codes for calculating the Fast Fourier Transform

This appendix has the Matlab code to calculate the fast Fourier transform of the magnetoresistance in order to extract the electron densities of the two wells.

```
FFT
% 1.Plot SdH oscillation and Hall resistance as a function of applied magnetic field
% 2.FFT method to find the peak position of the carry density for each 2DEG using
SdH oscillations
% Huang-Ming Lee, Mar.1st, 2005

% Load experiment datum and plot SdH oscillation and Hall resistance
clf;
clear all;
fdn = input(' Name of the folder?','s');
fnn = input('Name of the data you are going to compute?','s');
fn = [fnn,'.fma']
cmd = ['load c:\ntt_data\',fdn,'\',fn]; % Default directory of your data
eval(cmd);
cmd = ['data =',fnn,';']
eval(cmd);
cmd = ['clear',' fdn',' fnn',' fn'];
eval(cmd);

data = flipud(data);
data(1,:) = []; % Delete first row of data
mfd = data(:,1);
Rxx = data(:,2)./10*1E7;
Rxx_s = input('Enter the lock-in sensitivity of Rxx: ');
Rxx = Rxx*Rxx_s;
figure(1); clf;
plot(mfd, Rxx);
title('SdH oscillation');

Rxy = data(:,3)./10*1E7;
```

```
Rxy_s = input('Enter the lock-in sensitivity of Rxy: ');
Rxy = Rxy*Rxy_s;
figure(2); clf;
plot(mfd, Rxy);
title('Hall resistance');

% Polt 1/B as a function of Rxx
invB = 1./mfd
data(:,4) = invB
figure(3); clf;
plot(invB, Rxx);
axis([0,10,-inf,inf])
title('1/B vs Rxx');

% Interpolation for equal interval of invB with corresponding Rxx valuse
data(1:49,:) =[];
x = data(:,4);
y =    data(:,2);
xi = 1:0.0001:10;
yi = interp1(x,y,xi);
xi(40002:90001) = []; yi(40002:90001) = [];
intpdata(:,1) = xi; intpdata(:,2) = yi;
figure(4); clf;
plot(xi,yi);

% FFT to get the Fourier Coefficients in the Complex Plane using hanning
% window method
pi = 3.1415926535;
N = 40001;
n = 1:N;
hw = 0.5-0.5*cos(2*pi*n/(N-1)); % Hanning window
yihw = yi(:).*hw(:);
interval = 0.0001

fftdata = fft(yihw);
freq = (0:length(fftdata)-1)/(length(fftdata)*interval);
powerspect = abs(fftdata)/(sqrt(length(fftdata)));;
len = length(fftdata);
```

```
t = 0:interval:(len-1)*interval;
%subplot(2,1,1), plot(t, yi);
%xlabel('Time'), grid on;
%title('Time domain signal');
%subplot(2,1,2),
figure(5); clf;
plot(freq(1:len/2), powerspect(1:len/2));
axis([0 10 0 10]);
xlabel('Frequency (Hz)'), grid on;
title('Power spectral density');
```