Chapter 4

AC Susceptometer
4. AC susceptometer

4.1. AC susceptibility

The general definition of the AC susceptibility of a material is given by

\[
\chi_{ac} = \frac{\partial M}{\partial H}
\]  \hspace{1cm} (4.1.1)

where \(M\) is the magnetization per unit volume and \(H\) is the applied ac field. Thus, AC susceptibility is actually the slope \((dm/dH)\) of the magnetization curve.

At high frequency of the magnetizing ac field the time period becomes comparable to the relaxation time of the system and the magnetization cannot keep pace with the applied field as a result of which a phase lag is produced.

Let the magnetic field be represented by

\[
H(t) = H_0 + H_1 e^{i\omega t}
\]  \hspace{1cm} (4.1.2)

where \(H_0\) is the dc field and \(H_1\) is the amplitude of the ac field. If the sample be placed at a constant temperature i.e. for isothermal process the magnetization can be written as

\[
M(t) = M_0 + \chi H_1 e^{i\omega t}
\]  \hspace{1cm} (4.1.3)

where

\[
\chi = \chi' + i\chi''
\]  \hspace{1cm} (4.1.4)

\[
\chi = |\chi|e^{i\phi}
\]  \hspace{1cm} (4.1.5)

and where

\[
\chi' = \frac{M_1 \cos \phi}{H_1} \quad \text{and} \quad \chi'' = \frac{M_1 \sin \phi}{H_1}
\]

Both real and imaginary parts of the susceptibility depend on the frequency as well as on the amplitude of the applied magnetic field.

\(\chi''\) measures the dissipation of the system. The energy dissipated per unit volume is given by
4. AC susceptometer

\[ A = \frac{\omega}{2\pi} \int H \, dm \]  \hspace{1cm} (4.1.6)

where the integral is taken over a full cycle. From the expression of \( M \), \( A \) can be written as

\[ A = \frac{\omega}{2} \chi'' H_1^2 \]  \hspace{1cm} (4.1.7)

At low frequencies \( \chi' \) can be approximated by \( \chi \) and \( \chi'' = 0 \). Thus, in this case \( M \) and \( H \) are in phase. However, in the case of hysteresis loops caused by dry-friction-like pinning, \( \chi'' \) will be present even near-dc frequencies.

4.2. Experimental system

The whole system is shown in fig 4.2.1. Basically it has the following parts:

1. Coil assembly
2. Bridge circuit
3. Sample movement assembly
4. Temperature controller
5. Null detection unit.

4.2.1. Coil assembly

The coil assembly consists of Main coil system, Zero-setting variable mutual inductance coil, compensating coil and resistor system.

4.2.1.1. Main coil system (MC)

The main coil system is mounted on a stone plate placed horizontally very close to a furnace. The furnace will be used for very high temperature operation. The main coil consists of a primary of about 3000 turns of SWG 25 insulated copper wire of three layers. There are two trim coils of about 50 turns of the same wire as the primary wound at each end of the solenoid. These trim coils reduce fringe effect at the ends and maintain uniform magnetic field inside the solenoid. Two secondaries each of about 3000 turns made of SWG 28 copper wire of multiple layers are wound inside the primary. The secondaries inside the primary ensures the increase of the filling factor of the sample keeping the secondaries as close to the sample as possible. The secondaries are 6 cm apart. They are wound in the same direction but connected in series opposition. The former of the coil system is made of wood to minimize thermal expansion and eddy currents. A Faraday shield, in the form of a single layer of tightly wound insulated copper wire SWG 22 with one end left floating and another end grounded, is placed below the primary winding in order to reduce the capacitance between the primary and the secondary. The field profile and the secondary positions are shown in figure 4.2.2 and figure 4.2.3 respectively. The measurements were done using a small copper probe coil.

Calibration of the main coil (magnetic field versus current) was done with the help of a small probe coil of precise geometry (7.5 mm diameter and 25 turns). Pickup voltages were measured with different primary currents and frequencies. The calibration constant was determined to be $\approx 0.009 m^{-1}$. The photograph of the main coil system is shown in figure 4.2.4.

4.2.1.2. Zero-setting mutual variable inductance coil (VM)

This pair of coils is of the type used in the undergraduate / postgraduate laboratory. The coils are mounted vertically on a wooden board. The system is such that one coil can turn inside other. The inner coil is considered here as primary and this coil is fixed. The outer coil is taken as secondary which can rotate relative to the inner primary. The resistance of the primary is 9.6Ω and resistance of the secondary is 23.4Ω. The lower resistance is considered to be the primary
Figure 4.2.2.: Field profile of the primary of the main coil system as a function of distance along the axis. Data are taken with ‘home-made’ LIA. The y-axis shows the LIA output with the input taken from the probe coil. The exciting signal is supplied to the primary of the coil system.
Figure 4.2.3.: Secondary response profile of the main coil. Data are plotted with error bars. A shorted coil is used as a small test sample and the 'home-made' LIA is used to measure compound secondary output as a function of position.

Figure 4.2.4.: Photograph of the main coil system
4. AC susceptometer

so as to minimize the drive voltage. By turning the outer coil with respect to the inner one, one can change the mutual inductance by changing the effective flux linkage. So, manual compensation can be done with the help of this mutual inductance coil.

4.2.1.3. Compensating coil system (CC)

This is a fixed mutual inductance. Here two coils are mounted on a wooden flat board. The inner coil is the primary coil of resistance 11Ω and the outer coil is the secondary of resistance 16.5Ω. The compensating coil produces a 90° phase difference between the current in the primary and the voltage induced in the secondary. The signal from this secondary is attenuated by the bridge circuit to balance out the 90° component.

4.2.1.4. Total coil configuration

The block diagram of the coil assembly is shown in figure 4.2.5. The primaries of the MC, VM and CC are connected in series and this coil combination is connected to ground via a known resistor. With different resistance values of the series resistance one can fix the desired current through the primaries as also the signal across the resistor which may be used as an input to the bridge. A signal generator is connected across the primaries. The secondaries of MC and VM are connected in series and the sum of the signals (compound secondary) is fed to the bridge-balancing circuit. The signal from the CC is also fed to another input of the bridge-balancing circuit. This particular signal balances the part of the secondary pickup which is 90° out-of-phase with the current. The output taken from the known resistor is in phase with the current i.e. with the magnetic
field. This output is also fed to the bridge-balancing circuit and to the LIA as a reference input.

4.2.2. Bridge circuit

The change of secondary signal obtained by placing a sample alternately in the positions of the two secondaries is measured with the mutual inductance bridge. The nulling is done using a bridge circuit, the circuit diagram of which is shown in figure 4.2.6. It has six units. These are the compound secondary unit, the in-phase reference unit, the compensating coil secondary unit, attenuator and sign changer unit for the in-phase term, attenuator and sign changer unit for the in-quadrature term and an adder. Apart from the adder the other five units are basically unity-gain high input impedance differential instrument amplifiers. Each of the units are identical. The op-amps used here are TL084 and the resistances are all of values $3.3K\Omega$. All the ground connections are done through buffers to reduce ground return currents as all currents are returned to the power supply via a unit gain grounded amplifier. The inputs of the differential amplifiers are protected from PCB leakage currents with guard rings. The guard rings are placed at the same voltage as the input, but connected to the output of the unit-gain buffer. This allows leakage current to flow into the low impedance output without disturbing the input.

The positive power supply to the TL084 is also covered with a guard ring connected to the output of the unity-gain amplifier having ground input. Thus power supply voltages cannot leak out and disturb the other pins.

Two unbalanced bridges are connected from the outputs of the “in-phase reference unit” and “compensating coil input unit” respectively as shown in figure 4.2.6. The input of the “in-phase reference unit” is derived from the voltage drop across the known resistance in series with the primary coil assembly and that of the “compensating coil secondary unit” from the output of the “compensating coil secondary”. These bridges (figure 4.2.7) use the principle of the Wheatstone bridge of which the first and second arms are two $2.2k\Omega$ resistors and the third and fourth arms together is $5K\Omega$ potentiometer. The differential voltage between the junction between the two $2.2K\Omega$ resistors and the variable contact of the $5K\Omega$ potentiometer can thus be attenuated and, if necessary phase reversed, simply by turning the potentiometer knob. This is the attenuator and sign changer unit. This potentiometer is of 10-turn linear wire-wound type and is coupled with a stepper motor. The input of the bridges are taken from the output of the unit-gain differential amplifiers as shown in figure 4.2.6. As shown in figure 4.2.7, A is the input to each bridge and the output is taken from B and C via another
Figure 4.2.6.: Whole bridge circuit diagram. The op-amps used in each instrumentation amplifier is TL084. The op-amp used in the adder is LF351.
4. AC susceptometer

Figure 4.2.7.: Circuit diagram of the bridge portion

differential amplifier. The output of the two bridges are connected to an op-amp based adder circuit along with the output from the compound secondary. The op-amp used in the adder is LF351. The input resistances and the feedback resistance of the adder are of the same values \(10\,\text{K}\Omega\). The output of the adder is fed to a home-made digital LIA [Sec. 2.2]. We will discuss the details in the “null detection” unit. The PCB layout is shown in figure 4.2.8 and the photograph of the component side of the circuit board is shown in figure 4.2.9.

4.2.2.1. Null algorithm

Two stepper motors are coupled with two potentiometers, as discussed above. A stepper motor is a brushless dc electric motor that divides a full rotation into a number of steps. The stepper motors which are used here are collected from old 5½” floppy drives. These stepper motors complete one full rotation in 400 steps. As a 10-turn potentiometer is coupled with each stepper motor the full range of the potentiometer is available over 4000 steps of the stepper motor. The stepper control circuit is shown in figure 4.2.10. In this circuit, the stepper motors are controlled by a single set of “stepper control outputs” from the PHOENIX kit. The selection of stepper is done with the help of a relay which is also controlled by “digital outputs” of the PHOENIX kit.

Stepper algorithm

1. At the very beginning, the potentiometers (coupled with steppers) are turned to the extreme end. This is done manually. The positions of two steppers are now \((0,0)\) positions. The current positions are noted down in a temporary file of the PC.

2. The control program is written in C++ language. This program takes two integers as its input. First integer is the step number of stepper 1 and the second integer is the step number of stepper 2.

3. The program first checks the current positions from the temporary file.
4. AC susceptometer

Figure 4.2.8.: Solder-side PCB layout of the bridge circuit

Figure 4.2.9.: Photograph of the component side of the bridge circuit. The blue-coloured 10-turn potentiometers are coupled with stepper motors.
Figure 4.2.10.: Stepper motor control circuit. The SL100 transistors are diode protected inside the PHOENIX kits.
4. AC susceptometer

4. Comparing the desired position to the current position the stepper turns clockwise or anticlockwise. This is done through the sequence of bit patterns generated from the “stepper outputs” of the PHOENIX kit.

5. After movement of the first stepper, the second stepper is selected with the help of the relay present in the circuit.

6. Now the second stepper turns clockwise or anticlockwise depending upon the current position.

7. After final rotation of both the steppers the new positions are written to the existing temporary file as current position.

8. The destructor saves the final positions in a file in the hard disc. The constructor reads this position into the temporary file at the beginning. This allows subsequent runs without manual intervention. However, if the steppers need to be turned manually then one has to start from step 1.

The output of the bridge circuit, which is taken from the adder, is connected to the home-made “Computer-controlled digital LIA” [Section: 2.2]. It is seen from figure 4.2.6 that the adder output will be zero when the bridge circuit is perfectly balanced. Without any sample, the ideally balanced condition can be attained by turning the steppers to the mid positions of the potentiometer (ideally 2000, 2000). Any rotation of the steppers causes an imbalance which appears as a voltage at the adder output. The stepper motors are first calibrated. The calibration is done considering the cross-talk between the in-phase and in-quadrature parts of the circuit. We have adopted the following formalism to calculate the calibration constants of the steppers.

If cross-talk is taken into consideration, then the in-phase and in-quadrature voltages of the steppers are written as $V_p$ and $V_q$ respectively ($p$ and $q$ are the phase and quadrature step counts). The voltage and the stepper counts are related by

$$
\begin{pmatrix}
V_p \\
V_q
\end{pmatrix} = \begin{pmatrix}
V_{pp} & V_{pq} \\
V_{qp} & V_{qq}
\end{pmatrix} \begin{pmatrix}
p \\
q
\end{pmatrix} + \begin{pmatrix}
C_p \\
C_q
\end{pmatrix}
$$

In the matrix notation the above can be written as:

$$\vec{V} = A\vec{n} + \vec{C}$$

where the elements of $\vec{V}$ are the in-phase and in-quadrature voltages, the elements of $A$ are calibration constants, the elements of $\vec{n}$ are stepper step-counts. The $\vec{C}$ will depend on sample and also on the settings of the manual zero-setting
4. AC susceptometer

mutual inductance apart from circuit and coil imbalances. We are interested in nulling the bridge from the current state. Thus we need the slope matrix $\vec{A}$ only.

If we keep $q=0$ (stepper 2 at position 0) and change $p$ (stepper 1 position) only from 0 onwards, the equation takes the simple form of a straight line

$$V_p = V_{pp}p + C_p$$

From a least-square fit on $V_p$ versus $p$ data, $V_{pp}$ may be found. Similarly,

$$V_q = V_{qp}p + C_q$$

will yield $V_{qp}$.

Now if stepper 1 is returned to zero i.e. $p$ to zero (when $q$ is already zero) and another run is taken with only $q$, then

$$V_p = V_{pq}q + C_p$$

$$V_q = V_{qq}q + C_q$$

Least square fits on these two will give us $V_{pq}$ and $V_{qq}$. Thus the matrix can be obtained.

Now $\vec{V} = a\vec{n} + \vec{C}$ gives $\Delta\vec{V} = A\Delta\vec{n}$ which gives $\Delta\vec{n} = A^{-1}\Delta\vec{V}$.

To attain the null condition, the condition must be $\Delta\vec{V} = -\vec{V}$. This will give us how many steps we need to move on each stepper to reach zero output. To compensate for non-linearities, offsets and noise, more than one iteration may be necessary. At each stage it must be remembered that the count should not overshoot the end points of the potentiometers. Under such a situation, either the signal strength should be changed or the manual control should be invoked.

The above calibration is dependent on the amplitude and frequency of the applied input ac signal. For each set of amplitude and frequency calibration should be done before starting experiment. A C++ program was written to calculate the calibration constant of the steppers. This program writes the calibration table in a file of the PC. Our home-made “computer-controlled digital LIA” has two resolution modes. The calibration of the steppers is done for both the resolution modes. The calibration algorithm is discussed as below:

1. Select the resolution mode of the LIA.

2. Both steppers turn to the initial step number according to the user.

3. Stepper 1 starts turning from initial step to final step with a step-increment supplied by the user.
4. After each step-increment the stepper stops and the corresponding in-phase and in-quadrature voltages are measured by the LIA.

5. The above gives step/count number versus voltage for stepper 1.

6. After reaching the final step, stepper 1 returns to the initial step number.

7. Stepper 2 follows the same procedure.

8. The above sequence is followed for the other resolution mode.

9. The final calibration set is written to a file.

4.2.3. Temperature controller

4.2.3.1. Introduction:

In order to accurately control and measure the sample temperature, a temperature controller is essential. There are many approaches control temperature like ON/OFF, Proportional (P), Proportional Derivative(PD) and Proportional Integral Derivative (PID). These control mechanisms are based on Conventional Control Logic (CCL). There is another control logic - Fuzzy Logic Control (FLC) which has become very popular recently. We have used the CCL PID control mechanism for temperature control using a microprocessor.

4.2.3.2. Hardware:

The microprocessor based system used here comprises a Microprocessor trainer kit with the 4MHz Zilog Z80A processor, built-in Counter/Timer CTC, a control
4. AC susceptometer

The pulse width modulated (PWM) heater current control circuit is shown in figure 4.2.11. Basically this circuit is controlled by the Counter/Timer circuit (CTC) of the Z-80 based microprocessor kit followed by a monostable multivibrator and a darlington pair. The CTC has four independent counter/timer circuits, each containing the logic as shown in figure 4.2.12. The channel control logic receives 8-bit channel control word when the counter/timer channel is programmed. The channel control logic decodes the channel control word and sets the following operating conditions:

- Interrupt enable (or disable).
- Operating mode (Timer/counter).
Each channel of the CTC is individually programmed with two words; a control word and a time-constant word. The control word selects the operating mode (counter or timer) and other operation parameters. If the timer mode is selected, the control word sets a prescaler, which divides the system clock by either 16 or 256. The time-constant word is a value from 1 to 256. The timer mode determines time intervals as small as $4\mu s$ (frequency $4MHz$) without additional logic or software timing loops. Time intervals are generated by dividing the system clock with a prescaler (16 or 256) that decrements a preset down counter. A timer is triggered automatically when its time constant value is programmed, or by an external CLK/TRG input. The channel control word is shown below:

<table>
<thead>
<tr>
<th>$D_7$</th>
<th>$D_6$</th>
<th>$D_5$</th>
<th>$D_4$</th>
<th>$D_3$</th>
<th>$D_2$</th>
<th>$D_1$</th>
<th>$D_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INTERRUPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 Enables Interrupt; 0 disables Interrupt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_6$</td>
<td>MODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 selects timer mode; 1 selects counter mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_5$</td>
<td>PRESCALAR VALUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = value of 256; 0 = value of 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_4$</td>
<td>CLK/TRG MODE SELECTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 selects falling edge; 1 selects rising edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_3$</td>
<td>TIMER TRIGGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = automatic trigger; 1 = CLK/TRG pulse starts timer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_2$</td>
<td>TIME CONSTANT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = no time constant follows; 1 = time constant follows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_1$</td>
<td>PRESET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = continued operation; 1 = software reset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_0$</td>
<td>CONTROL OR VECTOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Vector; 1= Control word</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. AC susceptometer

In our case we have cascaded two channels of the timer/counter to make it a ‘large timer’. First the timer mode is selected through control word which is followed by a time constant which is determined by the heater power controlling software. After predefined time period the output of the timer resets to zero which triggers the counter that is also operated through a control word followed by a value. This value is also calculated by the software to control the exact heater current. The control word for selecting timer mode and the counter mode is shown below:

<table>
<thead>
<tr>
<th>Control word for timer mode</th>
<th>Control word for counter mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_7 ) D_6 D_5 D_4 D_3 D_2 D_1 D_0</td>
<td>( D_7 ) D_6 D_5 D_4 D_3 D_2 D_1 D_0</td>
</tr>
<tr>
<td>0 0 1 0 0 1 0 1</td>
<td>0 1 1 0 0 1 0 1</td>
</tr>
<tr>
<td>((25)_{16})</td>
<td>((65)_{16})</td>
</tr>
</tbody>
</table>

The output of the cascaded timer/counter is taken as the trigger of an IC 555 timer-based monostable multivibrator. Since the on-time of the monostable multivibrator in fixed, by varying the time constant of the timer and hence the triggering interval, the duty cycle (on-time/total time period) of the heater can be controlled. Depending on the requirement of heater power, which is calculated with the help a software by sensing the temperature by a PT100 RTD, heater current is adjusted through a darlington pair (BC108 & 2N3055) which is driven by the output of the monostable multivibrator. The heater is connected to the collector of the power transistor 2N3055. We have chosen constantan wire as the heating element of resistance \( \sim 14\Omega \). The wire is first insulated with Teflon tape and it is wound non-inductively like a bobbin to make it shaped like a spiral disk. The size of our heater is almost the same as that of the sample and it is mounted on one face of the sample. On the other face a Pt-100 (Platinum Resistance Thermometer) thin film type temperature sensor is attached. The voltage source of the heater as shown in figure 4.2.11 is \( +12\,\text{V} \), dc, but the dc source voltage can be be selected to different voltages upto \( +30\,\text{V} \), if necessary.

**Temperature sensor circuit:** The circuit diagram for the temperature sensor is shown in figure 4.2.13 and 4.2.14. The main element of this circuit is a PT100 (PRT) which offers an excellent accuracy over a wide temperature range (from \(-200\) to \(+850^\circ\text{C}\)). Standard Sensors are available from many manufacturers with various accuracy specifications and numerous packaging options to suit most applications. The principle of operation is to measure the resistance of a platinum element. The PT100 has a resistance of 100 ohms at \( 0^\circ\text{C} \) and 138.4 ohms at
4. AC susceptometer

100°C. The relationship between temperature and resistance is approximately linear over a small temperature range. The relation between the resistance and the temperature is given below:

\[ R_t = R_0 (1 + \alpha t + \beta t^2 + \gamma (t - 100)t^3) \],

where

- \( R_t \) = Resistance at \( t \)°C
- \( R_0 \) = Resistance at 0°C = 100Ω
- \( \alpha = 3.9083 \times 10^{-3} \)
- \( \beta = -5.775 \times 10^{-7} \)
- \( \gamma = -4.183 \times 10^{-12} \) below (0°C)
  or \( \gamma = 0 \) (above 0°C)

For a PT100 sensor, a 1°C temperature change will cause a 0.384Ω change in resistance, so even a small error in measurement of the resistance (for example, due to the resistance of the wires leading to the sensor) can cause a large error in the measurement of the temperature. For high precision we have followed the four-probe method where four wires are used - two to carry the sense current, and two to measure the voltage across the sensor. The current through the sensor will cause some heating. If the sensor element is unable to dissipate this heat, it will reflect as a higher temperature. To eliminate this effect we have used very small ac constant current \( \sim 49\mu A \) through the sensor. A chopper amplifier is used to amplify the output signal of the PT100 to such a value that the voltmeter used here can read the voltage at better accuracy. Because of the low signal levels, the cables are kept away from electric cables, motors and other devices that may emit electrical noise.

Figure 4.2.13 shows the square-wave generator part of the temperature controller circuit. Here an op-amp based astable multivibrator is used to generate square-wave. In order to get output pulses of equal on-off time, the saturation voltage is clipped to ±6.6V by two zeners (α). After reducing the peak voltages
4. AC susceptometer

Figure 4.2.14.: Pre-amplifier circuit of the temperature controller circuit

of the square wave it is fed to two LM339 comparators, which produce two out-of-phase square waves. The peak of the square waves are at $\pm 5\text{V}(\beta)$ and $\mp 5\text{V}(\gamma)$ with errors of $\sim 0.2\text{V}$ due to the saturation drop of the open-collectors stages. These square waves are used to control two CMOS (CD4066) switches, the outputs of which are shorted into one and inputs are connected to +5V and -5V respectively (figure 4.2.14). Thus the amplitudes of the output waveform are accurately tied to $\pm 5\text{V}$, which is maintained by accurate voltage regulator ICs. The constant current is maintained by the input resistance ($100K\Omega$) of the inverting amplifier configured as a voltage-to-current converter. The voltage across of the PT100 which is connected as the feedback resistance of the inverting amplifier, is fed through an instrumentation amplifier of 100 gain. Another pair of CMOS switches, driven by the same input signals $\beta$ and $\gamma$, are now used to rectify the signal (like a LIA) and the final outputs filtered by a RC stage before being fed to the input of the 16-channel voltmeter (Chapter 3).
4. AC susceptometer

![Frontend of the temperature controller daemon program](image.png)

Figure 4.2.15.: Frontend of the temperature controller daemon program

### 4.2.3.3. Software:

The software is basically based on MPI-based daemon programs written in C++ and tcl/tk languages. The main daemon program is “tempctld” (Temperature controller daemon) which controls the whole process. There is another helper program which sends requests (wake-up call) to the daemon program periodically. The daemon program senses temperature, calculates heater power and sends requests to the microprocessor CTC to feed desired pulses for maintaining heater temperature. The graphics frontend of “tempctld” is shown in figure 4.2.15. The working principle is as given below:

1. It initially sets “heater power”, “set temperature”, “proportional term (P)”, “integral term (I)” and “derivative term (D)” to ‘ZERO’.

2. Sends request to meter daemon (using channel 14) to measure the output of the temperature controller circuit.

3. Calculates the resistance value of PT100 and displays the “Actual temperature”.

4. The daemon has different input variables i.e. Set Temperature; P, I, D; Modes - Set / Sweep; Heater range- Low / Medium / High; Locations (6 locations). Any new entry can be made by directly inserting the values into the GUI or using a command-line program named “tempctld_set”. A library of subroutines enables other programmes to send requests to “tempctld”.
4. After getting new values, the daemon program first calculates the required heater power using PID algorithm.

6. The daemon has three heater power ranges each of which has predefined integer values.

7. Depending upon the range of the heater power, the program calculates the second integer value using heater power formula.

8. The daemon program sends request to microprocessor and dumps the calculated integers to the CTC.

9. The CTC sends pulses with desired ON/OFF ratio to the control circuit for heater.

10. The daemon now enters a blocking receive mode and waits till it is woken up either by the helper (to update the controller parameters periodically) or by some other program requesting specific actions, like changing PID or setting a new temperature, etc.

11. The temperature controller can be used in two modes; set mode and sweep mode. In set mode the daemon program sets the “set temperature” and waits for any call and in the sweep mode the program starts from the initial temperature and continually changes the target temperature with a temperature rate defined by the user or any other program.

12. The temperature controller daemon has six locations. The previous integral value is stored in a 6-member array. This function is very useful for controlling temperature at different positions by using a suitable starting heater power if needed.

The daemon programs and the helper programs are shown in Appendix C.

4.2.4. Sample movement mechanism:

In this type of experiment double secondaries (connected in opposition) are used to minimize the air-core mutual inductance between the primary and the compound secondary without compromising on the sensitivity of the setup. For this purpose the sample should be moved from one secondary position to another while continuously measuring the signal in each coil. Since the two secondaries are roughly identical and in opposition, it is assumed that the part of the signal due to the sample will simply change sign as the sample is moved from one coil to
4. AC susceptometer

the other. By taking half of the difference between the two signals, it is thus possible to get the contribution due to sample only. For example, if $M \downarrow$ represents the value of the mutual inductance when the sample is in the ‘down’ position, and $M \uparrow$ the corresponding value when the sample is in the ‘up’ position, and $M_0$ and $M_{\text{sample}}$ are the mutual inductances due to the empty coil and the sample respectively, then we can write:

\[
M \downarrow = M_0 + M_{\text{sample}} \\
M \uparrow = M_0 - M_{\text{sample}} \\
\Rightarrow M_{\text{sample}} = (M \downarrow - M \uparrow)/2 \equiv \Delta M
\]

As the secondaries are very narrow, any error in the sample position relative to the position of the secondaries will cause an error in the mutual inductance. In order to minimize this error, we have made an arrangement for moving the sample rod with help of a PHOENIX-controlled dc motor using servomechanism technique.

4.2.4.1. Mechanical assembly:

As in our case the main coil assembly is placed horizontally, the movement of the sample should be in horizontal direction. In order to do this movement we have managed to use the frame and movement assemblies of an old unused “XY chart recorder” of our department. This chart recorder use the principle of potentiometric (servo) mechanism where a motor-operated pen is arranged to move the sliding contact of a potentiometer to back feed the pen position to an error amplifier. The amplifier drives the motor in such a direction as to reduce the error between desired and actual pen positions to zero. We have removed the pen and mounted a bench-top with wheels (in-house fabricated) in place of the pen stand (fig ). As we need the movement along one axis only, we have used a single servo motor. The whole movement mechanism is controlled by PHOENIX-based control circuitry.

4.2.4.2. Movement control circuit:

The movement control circuit is shown in figure 4.2.16. Here two SPDT relays are used to control the direction of rotation (clockwise / anticlockwise) of the motor. The motor is connected between these relays in such a way that by default it allows flow of current in one direction only. When the relays are operated simultaneously through the control, the current flows in the opposite direction
Figure 4.2.16.: Sample movement control circuit
4. AC susceptometer

through the motor. From figure 4.2.16 it is seen that the motor-with-relays are connected to the collector of a power transistor with darlington pair 2N6300. To protect the transistor from the back emf of the motor a diode (1N4007) is connected in parallel with the motor. The transistor is driven by the output of a comparator (LM339) which is controlled through microprocessor.

As the motor is coupled with a potentiometer, any rotation of the motor causes a change in the resistance of the potentiometer. The fixed terminals of the potentiometer have been connected between +12V and ground, whereas the variable terminal has been connected to an ADC channel of the PHOENIX kit. Thus, any rotation of the motor causes a change of voltage of the potentiometer which can be directly measured by the ADC channel of the PHOENIX kit.

We have mentioned earlier that in this experiment, the sample should be moved between two positions UP and DOWN. To control the motion, the motor rotation and hence sample position is first calibrated against the potential drop appearing across the coupled potentiometer. To move from one position to another the control unit first measures the present voltage across the potentiometer and send instructions to digital output channels to act accordingly.

4.2.4.3. Software details:

The softwares for controlling sample movement has been written in C++ and TCL/TK GUI languages. The movement is also controlled by the daemon programme 'phoenixd' with help of special library header file “susceptometer.h”. The algorithm is as given below:

1. At first it checks the current voltage through ADC channel of PHOENIX.
2. Then it calculates the error voltage i.e. the difference voltage between voltage which is to be fixed and the present voltage.
3. If the error voltage is zero, it stays at the current position.
4. Depending upon the polarity of the error voltage it operates the relays simultaneously to set the motor to rotate in the desired direction.
5. The rotation or movement is done using short pulses from the PHOENIX digital outputs.
6. While in motion the current voltage is measured repeatedly and it stops moving when the desired voltage level is attained.
7. Wooden stays have been used to improve the accuracy of the movement.

The movement supporting programs are shown in the appendix.
4. AC susceptometer

4.2.5. Null detection unit:

We describe here an algorithm for operating an AC susceptibility setup which eliminates much of the problems associated with coil drift and temperature stabilization of the sample. This increases the reproducibility of the experiments.

When the sample is moved UP or DOWN within the main coil system of our coil assembly, a change occurs in the temperature of the sample due to different environment. Since \( M \downarrow \) and \( M \uparrow \) should be measured at the same temperature, which is very hard to maintain in high temperature ac experiments, we have adopted the following procedure:

1. As discussed earlier, our temperature controller daemon program has two modes to fix the temperature; “set” and “sweep” mode. Before starting the experiment the sample is placed in one position and is allowed to attain the “starting temperature” in “set” mode. This part is done manually with user defined P, I and D. When it attains stabilization to the desired temperature, the sample is placed in another position of the coil, the temperature controller daemon memorizes the integral value of the first position and writes the value as a “location number”. The same procedure is then followed for the other position of the secondary also. Thus the program can “remember” location-wise integral value upto six locations. As the heater power depends upon P, I, D and offset temperature whereas present “I term” depends upon the last “I term(locationwise)” , heater power must be different for different positions of the sample.

2. After attaining the start temperature for both sample positions, the daemon program switches itself to “sweep mode” and starts increasing/decreasing the with a userdefined rate (\( \Delta T/min \)). When the sample is moved to one position the program first finds the location and starts sweeping temperature from the last “I term(locationwise)” i.e. integral value. The procedure which is adopted here for maintaining locationwise temperature, minimizes errors due to difference in the environment of the sample.

The block diagram of the whole system is shown in figure 4.2.17. The main program is the “scan” [Appendix D.1] program which calls daemon programs tempctl\_d and phoenix\_d. The measurement has been done by placing the sample in each position of the secondaries. The change in magnetization (\( \frac{dM}{dH} \)) has been measured by our bridge, using the digital LIA[2.2] as a null detector. The ideal situation would be when measurements are taken at the two sample positions (inside the secondary) at the same temperature and the difference of the two values taken. However, since the two positions have different thermal environments, the
temperature controller needs time to stabilize the temperature in each. Circuit and mechanical coil drifts occur in the mean time and accuracy is greatly reduced. Moreover, the procedure is time consuming. We adopt a slightly different approach with the \textit{scan} programme. The temperature controller is now set in the sweep mode with a sweep rate of typically 1-2 degrees per minute. The sample is moved up, the bridge is balanced, the values and final temperature recorded in a file. The sample is now moved down and the same procedure followed, with the data stored in another file. This process repeats till the final temperature is attained. After the full run is complete, we thus have two files, one containing the values for the ‘up’ position and the other for the ‘down’ position. The temperatures for corresponding points for the two files is not the same, as the sweep setting keeps on changing the temperature. In order to take differences for values corresponding to the same temperature, we use a processing program \textit{scan\_processor}[Appendix D.2]. This programme reads in the data from the two files into arrays, sorts them, does a running-average smoothing on them (the number of values to average over has to be set by the user) and then interpolates the data to get values at matching temperatures. The difference (divided by two) is now stored in an output file, which now contains the usable data from the experiment.
4.2.6. Calibration and performance:

Calibration of a susceptometer is complicated by the fact that the signals depend strongly on the geometry of the samples. Thus values of the measured susceptibility as a function of mass will vary for the same material if the samples have different geometry. In our case a further difficulty arises because we are operating in the high temperature range where paramagnetic materials which are typically used for calibration will give very low signals. We have used samples of high purity nickel for calibration. The samples were in the form of discs of diameter 7.5 mm which is the same as that of the samples we have studied later. Thus geometric considerations have been taken care of to a certain extent. On the other hand ferromagnetic materials are hardly suitable for calibration - (i) they are inherently nonlinear, (ii) they follow history dependent loops which makes the magnetization a multiple valued function of applied field. In our case the applied fields are small and so the non-linearity is not expected to show up predominantly. To get an idea of the multiple-valued nature of the M-H curve of the nickel sample we made an enhanced susceptibility study (enhancement has been discussed in Chapter 5). The sample shows almost negligible enhancement probably because it is of very high purity and relatively strain and defect free. So the hysteresis curve is expected to be very narrow and can be roughly approximated by a line.

Two samples of 277 mg and 571 mg were used. The calibration constant from both came out to be $7 \times 10^{-8}$ kg-equivalent of nickel per step of the stepper motor. Since ratios are the relevant quantities in most cases, we have shown all our data in step numbers.

An example of data using this setup with a sample of double perovskite structured $Sr_2FeMoO_6$ (SFMO) is shown in figure 4.2.18.
4. AC susceptometer

Figure 4.2.18.: Test data taken with the susceptometer on a sample of SFMO. $\chi'$ is in steps number.