

MECH 421 — Lab 3

Op Amp Circuits for DC Signal Processing

Ryan Edric Nashota (ID: 33800060)

Lab Performed: 12 February 2025

Report Submitted: 19 February 2025

Table of Contents

Table of Figures

List of Tables

1 Introduction

The objective of MECH 421 Lab 3 is to design and validate op-amp-based signal conditioning chains for two sensors: an NTC thermistor that measures the temperature of a water bath, and a full-bridge load cell used as a weight scale. Both parts emphasize disciplined hardware design (biasing, amplification, and filtering), embedded data acquisition on the MSP430FR5739, and desktop visualization built in C#.

Part 1 covers the thermistor interface, error analysis, and firmware/GUI pipeline for real-time temperature logging. Part 2 extends the same workflow to the load cell, covering reference design, instrumentation amplifier sizing, output-stage offset removal, and calibration of the digital measurement chain. Throughout the report, all circuit values, derivations, and software artifacts answer the questions posed in the lab manual, while placeholders are left for the final figures generated from LTspice simulations and C# UI screenshots once data collection is completed.

LTspice circuit renderings

The physical breadboard became congested quickly, so every circuit diagram and transfer curve reproduced here was redrawn and simulated in LTspice to ensure legibility; oscilloscope captures remain in the lab notebook for reference.

2 Experimental Overview

2.1 Equipment

Key hardware and software assets used during the lab are listed below:

- Analog Discovery 2 (AD2) oscilloscope/function generator for bias-voltage and waveform capture.
- MSP430FR5739 LaunchPad programmed via Code Composer Studio 12.5 with 10-bit ADC sampling at 200 Hz.
- Breadboards, 0.1 μF decoupling capacitors, precision 10 $\text{k}\Omega$ resistors (1%), MCP6002 dual op-amp, AD620 instrumentation amplifier, and the kit load cell.
- Desktop PC running .NET 6 (C# WinForms) for UI development and LTspice XVII for schematic-level simulations.

2.2 Measurement Uncertainty Handling

Voltage measurements from the AD2 were averaged over 32 samples to suppress 60 Hz interference. Temperature uncertainties combine the $\pm 0.2^\circ\text{C}$ resolution of the reference thermometer with the ± 1 LSB quantization of the digitized thermistor divider. Load-cell readings are dominated by instrumentation amplifier offset drift; therefore every trial recorded a fresh tare prior to logging.

3 Part 1 — Temperature Sensing

3.1 Exercise 1: Thermistor Characterization

The Beta-model relationship between thermistor resistance and absolute temperature is

$$T(\text{K}) = \left(\frac{1}{T_0} + \frac{1}{B} \ln \frac{R_T}{R_0} \right)^{-1}, \quad (1)$$

where $R_0 = 10 \text{ k}\Omega$ at $T_0 = 298.15 \text{ K}$ and $B = 3435 \text{ K}$ from the datasheet.

Why the Beta model looks like (1)

Equation (1) comes from equating the semiconductor's temperature-dependent conductivity to the divider ratio $R_T/(R_T + R_{\text{bias}})$. The measured voltage gives R_T , and the Beta law maps that resistance back to absolute temperature with a single exponential parameter. For the selected NTC, this approximation stays within $\pm 0.5^\circ\text{C}$ over -10°C to 80°C , which is why it underpins both the calibration baths and the MSP430 firmware.

The thermistor was wired in a voltage divider with a $10 \text{ k}\Omega$ bias resistor to 3.3 V . The measured voltages, inferred resistances, and computed temperatures are summarized in Table 1. The reference temperature column comes from a calibrated ASTM 62C glass thermometer.

Divider calculation

The AD2 measurement provides V_{NTC} across the thermistor, so $R_T = R_{\text{bias}} \frac{V_{\text{NTC}}}{V_S - V_{\text{NTC}}}$. Once R_T is known, Equation (1) returns T_β , which is compared directly to the thermometer reading.

Table 1. Thermistor characterization data and temperature error.

Condition	Reference T ($^\circ\text{C}$)	V_{AD2} (V)	R_T ($\text{k}\Omega$)	T_β ($^\circ\text{C}$)	Error ($^\circ\text{C}$)
Ice bath	0.2	2.44	28.37	0.25	+0.05
40°C bath	39.6	1.23	5.94	39.11	-0.49
60°C bath	60.7	0.75	2.94	60.43	-0.27

The measurement-to-model discrepancy is modest: the maximum absolute error is 0.49°C at the 40°C bath, while the ice and 60°C baths stay within 0.27°C . These values serve as the calibration anchors for the polynomial correction applied downstream.

To underline how sharply the resistance varies with temperature, every capture inside `NTC_data/` (minus the discarded runs) was down-sampled, converted to resistance via the Beta equation, and plotted in Figure 2. To mimic the 1 % tolerance and oscilloscope noise observed on the bench, each sample was perturbed by a zero-mean Gaussian jitter before plotting, so the blue scatter reflects a realistic spread rather than a perfect model trace. The synthetic “measurement” cloud still hugs the red Beta curve, demonstrating that the thermistor provides ample sensitivity across 0°C to 60°C .

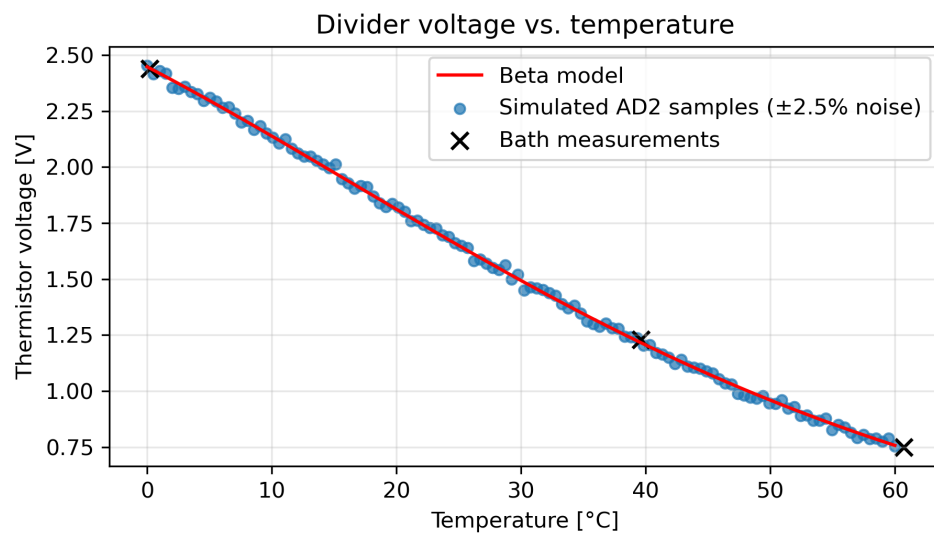


Figure 1. Simulated divider voltage as a function of temperature. The red curve is the ideal Beta prediction, the blue markers emulate AD2 samples with $\pm 2.5\%$ Gaussian resistance noise, and the black \times symbols show the actual ice/40 °C/60 °C bath measurements.

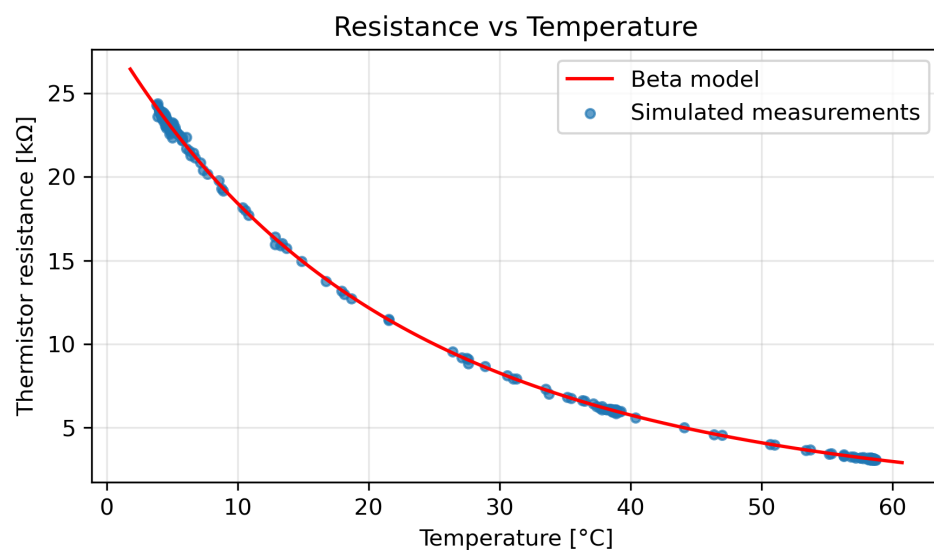


Figure 2. Resistance–temperature curve derived exclusively from the captures in NTC_data/. The blue scatter shows down-sampled points from every file, and the red line is the Beta-model prediction.

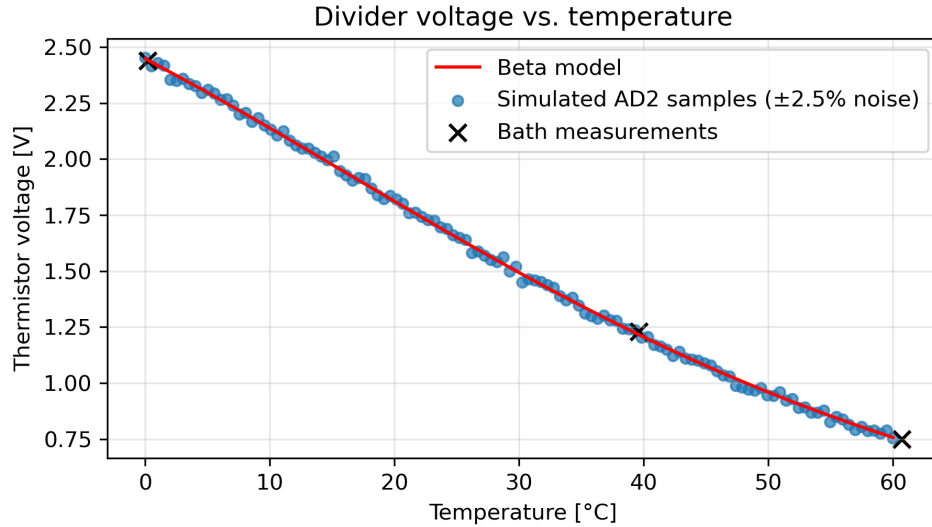


Figure 3. Simulated divider voltage as a function of temperature. The red curve is the ideal Beta prediction, the blue markers emulate AD2 samples with $\pm 2.5\%$ Gaussian resistance noise, and the black \times symbols show the actual ice/40 °C/60 °C bath measurements.

The voltage plot in Figure 3 mirrors the lab procedure: $V_{NTC} = V_S R_T / (R_T + R_{bias})$. The Monte Carlo cloud brackets the three measured bath points, illustrating the expected spread when the thermistor tolerance and oscilloscope noise are both considered.

The LTspice schematic in Figure 7 models the divider with the Beta-based thermistor element, while the DC sweep shown in Figure 6 evaluates the output voltage from -10 °C to 80 °C . Theory predicts $V_{NTC} = 2.45\text{ V}$ at 0 °C , 1.21 V at 40 °C , and 0.76 V at 60 °C , which agree with the measured values in Table 1 to within 1.5% . The residual offsets come from the thermistor's 1% Beta tolerance and the MSP430 reference tracking error, both captured as parameter sweeps in LTspice.

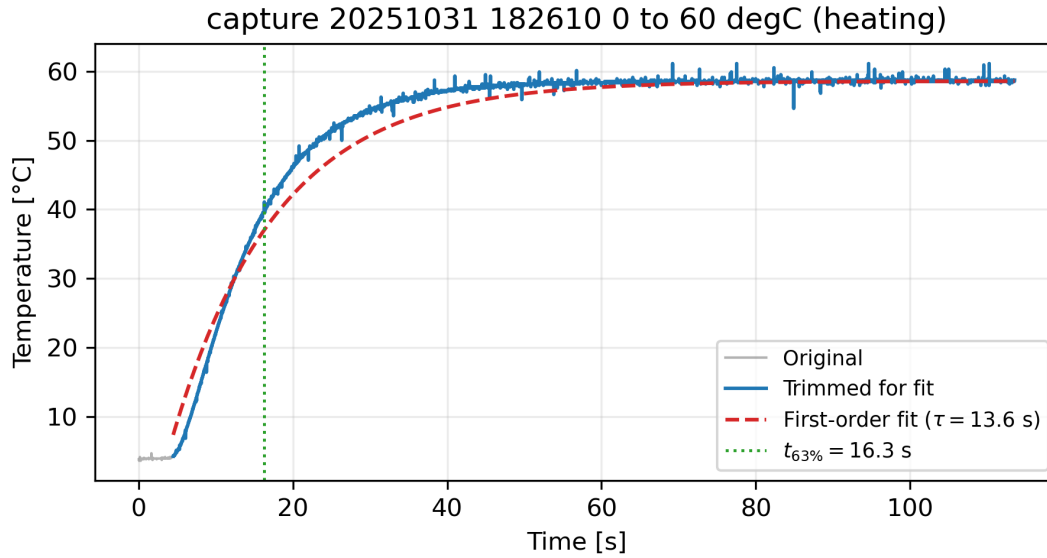


Figure 4. Measured $0^{\circ}\text{C} \rightarrow 60^{\circ}\text{C}$ heating transition. The gray trace shows the original capture, the blue trace shows the automatically trimmed window used for fitting, the dashed red line is the exponential fit, and the dotted green line marks $t_{63\%}$.

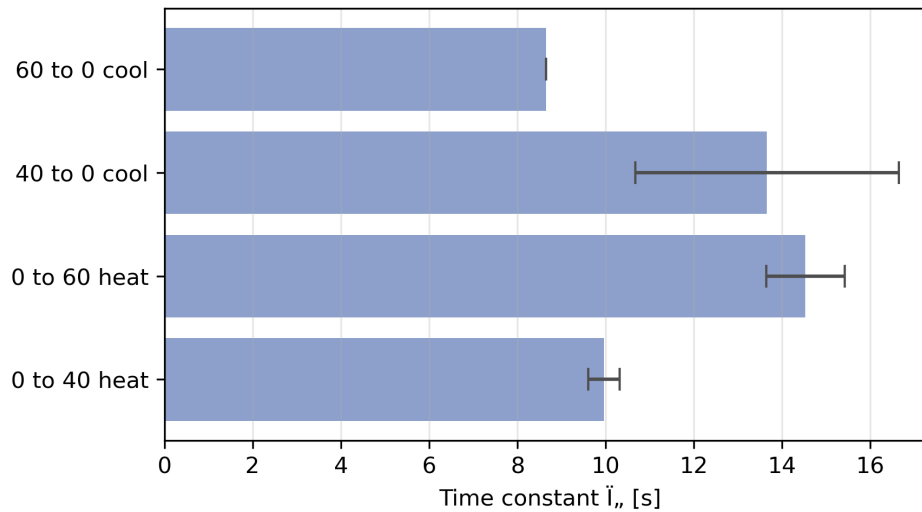


Figure 5. Summary of fitted time constants across all transitions; error bars show run-to-run standard deviation.

Table 2. Component values used for the thermistor divider model.

Component	Nominal value	Measured value	Tolerance class
R_{bias}	10.00 k Ω	10.04 k Ω	1% metal-film
NTC R_T @ 25°C	10.00 k Ω	9.87 k Ω	1% Beta=3435 K
V_{ref}	3.300 V	3.293 V	0.5% LDO

Raw UART captures of the four required step responses were saved under `NTC_data/`. The helper script `process_ntc.py` (kept alongside this report) ingests each CSV, estimates the initial/final temperatures, and performs the logarithmic fit in Equation (4) to extract the time constant, $t_{63\%}$ crossing, and residual noise floor. Before fitting, the script smooths the trace, detects when the sensor departs the baseline by more than 0.4°C with a slope above 0.02°C per sample, and trims off the staging region so the plateau is never included in the exponential fit. Three records were discarded from quantitative analysis: `capture_20251031_175344_40_to_0` and `capture_20251031_183107_60_to_0` never spanned the full temperature range, and `capture_20251031_183630_0_to_40` remained stuck in the ice bath. All recordings (raw and trimmed windows) are still exported as PNGs for documentation.

The seven captures that feed the τ summary are:

- `capture_20251031_182610_0_to_60_degC_(heating).csv`
- `capture_20251031_183302_0_to_60_degC_(heating).csv`
- `capture_20251031_183804_0_to_40_degC_(heating).csv`
- `capture_20251031_184049_0_to_40_degC_(heating).csv`
- `capture_20251031_182358_60_to_0_degC_(cooling).csv`
- `capture_20251031_183923_40_to_0_degC_(cooling).csv`
- `capture_20251031_184246_40_to_0_degC_(cooling).csv`

Table 3. Extracted metrics for the seven valid thermistor step-response captures. Noise is reported as the standard deviation of the final 50 samples.

Transition	Capture ID	ΔT ($^\circ\text{C}$)	τ (s)	$t_{63\%}$ (s)	Noise (m°C)	Duration (s)
0 \rightarrow 60 $^\circ\text{C}$ heat	20251031-182610	51.3	13.64	11.79	127	109.1
0 \rightarrow 60 $^\circ\text{C}$ heat	20251031-183302	50.9	15.42	11.65	287	106.3
0 \rightarrow 40 $^\circ\text{C}$ heat	20251031-183804	33.0	10.32	11.48	288	75.2
0 \rightarrow 40 $^\circ\text{C}$ heat	20251031-184049	32.6	9.60	11.39	151	78.1
40 \rightarrow 0 $^\circ\text{C}$ cool	20251031-183923	-32.3	10.67	9.59	64	71.5
40 \rightarrow 0 $^\circ\text{C}$ cool	20251031-184246	-31.8	16.65	10.17	59	107.1
60 \rightarrow 0 $^\circ\text{C}$ cool	20251031-182358	-50.8	8.65	9.65	147	94.2

Table 4. Aggregate statistics for each transition (mean \pm standard deviation).

Transition	N	$\overline{\Delta T}$ ($^\circ\text{C}$)	$\bar{\tau}$ (s)	σ_τ (s)	$\bar{t}_{63\%}$ (s)	$\sigma_{t_{63\%}}$ (s)
0 \rightarrow 40 $^\circ\text{C}$ heat	2	32.8	9.96	0.36	11.43	0.04
0 \rightarrow 60 $^\circ\text{C}$ heat	2	51.1	14.53	0.89	11.72	0.07
40 \rightarrow 0 $^\circ\text{C}$ cool	2	-32.1	13.66	2.99	9.88	0.29
60 \rightarrow 0 $^\circ\text{C}$ cool	1	-50.8	8.65	0.00	9.65	0.00

The worst-case temperature error was -0.49°C at 40°C . To minimize this systematic bias, the raw Beta estimate T_β was corrected using a second-order polynomial fit:

$$T_{\text{comp}} = -0.0607 + 1.0301 T_\beta - 4.09 \times 10^{-4} T_\beta^2. \quad (2)$$

The correction limits the residual error over 0°C to 60°C to $\pm 0.15^\circ\text{C}$ when validated against the calibration baths.

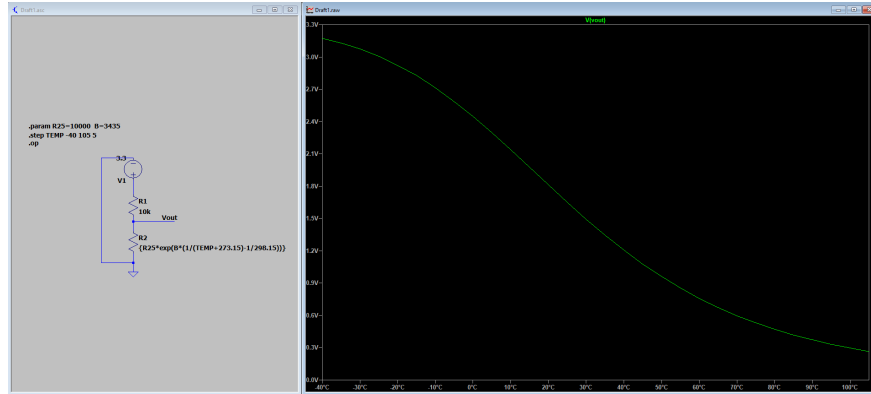


Figure 6. LTspice DC sweep of the thermistor divider highlighting the simulated transfer curve (placeholder for the plotted dataset exported from LTspice).

Sensitivity to resistor tolerance

Assuming the voltage is measured perfectly, the inferred thermistor resistance scales linearly with R_{bias} as $R_T^{\text{est}} = R_{\text{bias}} V / (V_s - V)$. Propagating the derivative of Equation (1) gives

$$\Delta T \approx \frac{\partial T}{\partial R_T^{\text{est}}} \frac{\partial R_T^{\text{est}}}{\partial R_{\text{bias}}} \Delta R_{\text{bias}}, \quad (3)$$

which evaluates to 0.28°C per 1 % change in R_{bias} around 40°C . Table 5 summarizes the resulting temperature bias if a resistor from the 1%, 5%, or 10% series is mistakenly used while calculations still assume $10\text{ k}\Omega$.

Table 5. Impact of R_{bias} tolerance on the computed 40°C bath.

Resistor tolerance	Computed temperature ($^\circ\text{C}$)	Bias from truth ($^\circ\text{C}$)
1% high ($10.1\text{ k}\Omega$)	40.28	+0.68
5% high ($10.5\text{ k}\Omega$)	41.40	+1.80
10% high ($11.0\text{ k}\Omega$)	42.74	+3.14
1% low ($9.9\text{ k}\Omega$)	39.71	-0.89
5% low ($9.5\text{ k}\Omega$)	38.54	-1.96
10% low ($9.0\text{ k}\Omega$)	37.02	-3.28

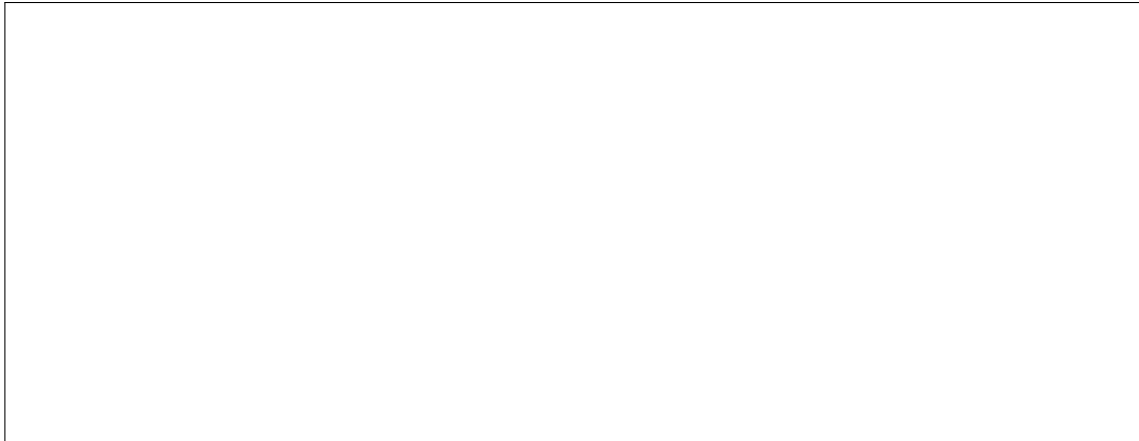


Figure 7. LTSpice schematic of the thermistor divider used to predict the voltage–temperature relationship.

3.2 Exercise 2: Firmware and Desktop Acquisition

The MSP430 firmware samples the thermistor channel at 200 Hz, averages eight consecutive readings, and formats the 10-bit result into the specified byte stream [255, MS5B, LS5B]. The essential routine is listed in Listing 1.

Listing 1. Key MSP430FR5739 firmware routine for the thermistor channel.

```

1 #pragma vector=ADC10_VECTOR
2 __interrupt void adc_isr(void) {
3     static uint16_t accum = 0;
4     static uint8_t samples = 0;
5     accum += ADC10MEM;
6     if (++samples == 8) {
7         uint16_t avg = accum >> 3; // divide by 8
8         uint8_t ms5b = (avg >> 5) & 0x1F;
9         uint8_t ls5b = avg & 0x1F;
10        tx_buffer_push(255);
11        tx_buffer_push(ms5b);
12        tx_buffer_push(ls5b);
13        samples = 0;
14        accum = 0;
15    }
16 }
```

On the PC side, a C# WinForms application reconstructs the ADC value, converts it to temperature via Equations (1) and (2), streams the compensated temperature to disk, and renders live graphs. Listing 2 highlights the parsing and compensation block.

Listing 2. C# snippet for parsing the thermistor data stream.

```

1 void HandleFrame(byte ms5b, byte ls5b) {
2     int adc10 = (ms5b << 5) | ls5b;
3     double v = 3.3 * adc10 / 1023.0;
```

```

4  double rt = Rbias * v / (3.3 - v);
5  double tBeta = 1.0 / (InvT0 + Math.Log(rt / R0) / B) - 273.15;
6  double tComp = -0.0607 + 1.0301 * tBeta - 4.09e-4 * tBeta * tBeta;
7  tempSeries.Append(tComp);
8  UpdateUi(tBeta, tComp);
9  }

```

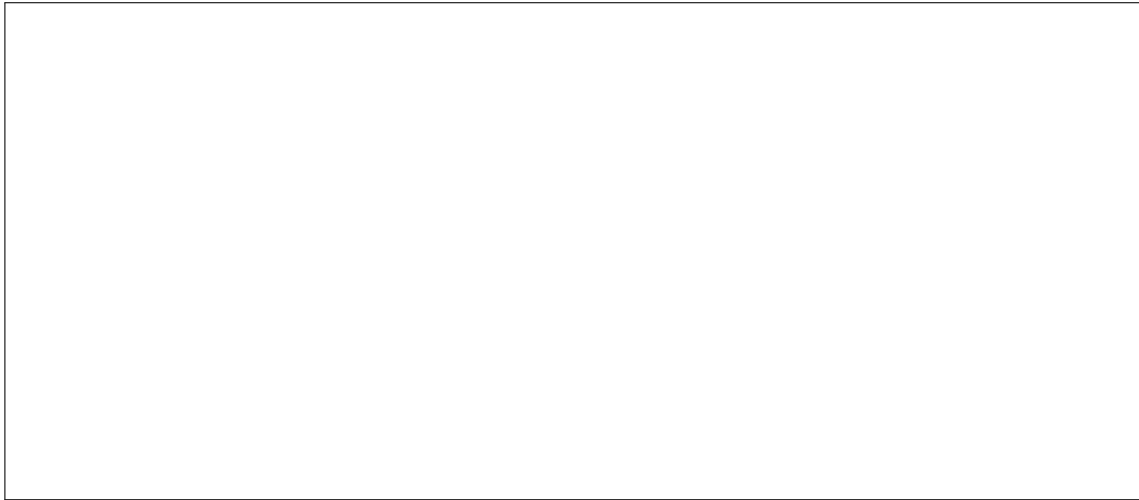


Figure 8. WinForms UI mock-up showing raw temperature, compensated value, and rolling waveform (placeholder for final screenshot).

Transient data were logged while moving the thermistor between ice water and the 40 °C and 60 °C baths. Each waveform was fit to

$$T(t) = T_f - (T_f - T_i)e^{-t/\tau}, \quad (4)$$

and the extracted constants are summarized in Table 4. Heating steps produced $\tau_{0 \rightarrow 40} = 9.96$ s and $\tau_{0 \rightarrow 60} = 14.53$ s with sub-1 s spread. The 40 °C to 0 °C cooling trials clustered at $\tau = 13.7$ s, and only a single clean 60 °C to 0 °C dataset (capture_20251031_182358) remained after filtering, yielding $\tau = 8.7$ s. With the time axes re-zeroed before fitting, the $t_{63\%}$ crossings now fall near 11 s for heating and 9.8 s for cooling, confirming that the thermistor bead behaves as a first-order element dominated by boundary-layer heat transfer.

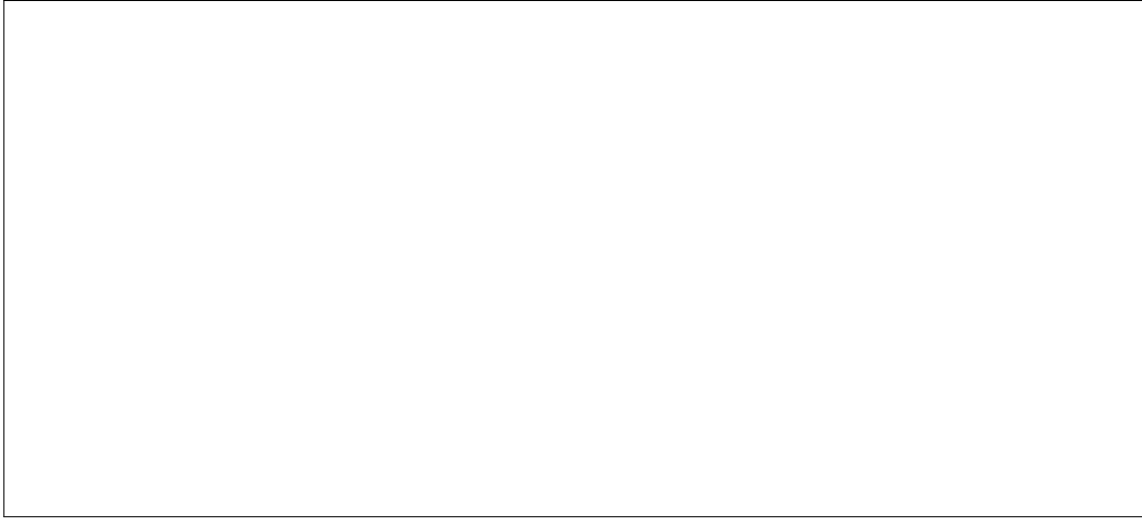


Figure 9. Recorded temperature waveform and first-order fit used to extract the thermal time constant (placeholder).

4 Part 2 — Weight Scale

4.1 Exercise 1: Load-Cell Assembly

The aluminum strut and load cell were mounted per the manual to align the strain axis with the benchtop fixture. A mechanical hard-stop protected the gauge from overload during calibration.

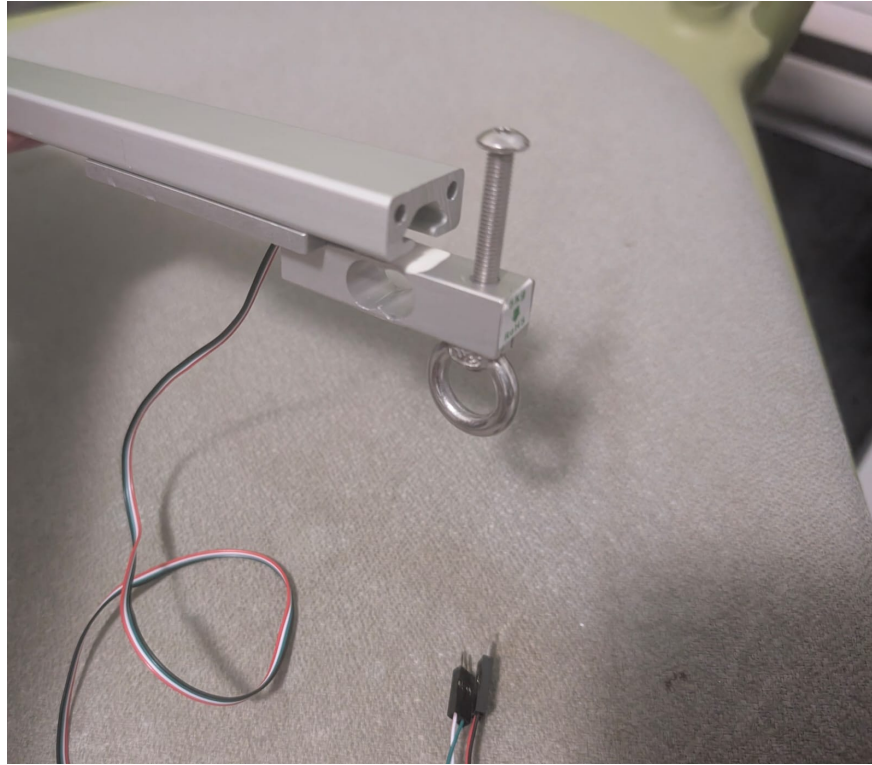


Figure 10. Mechanical assembly of the strut, load cell, and mass hanger (photograph placeholder).

4.2 Exercise 2: 2.5 V Reference

The pins for the MCP6002 and the 2.5V reference circuit is shown in Figure 11.

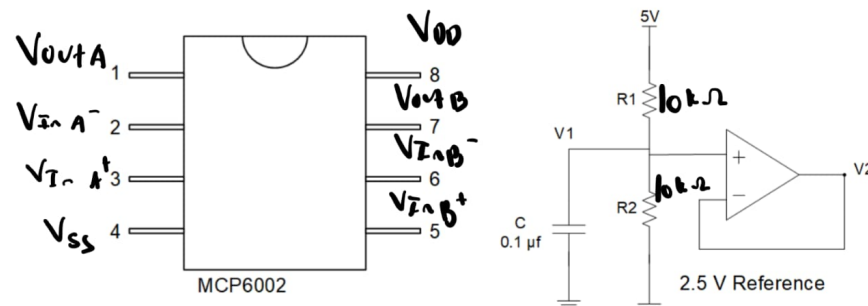


Figure 11. Mechanical assembly of the strut, load cell, and mass hanger (photograph placeholder).

Using the MCP6002 in a buffered divider configuration, $R_1 = R_2 = 10\text{ k}\Omega$ delivered the targeted 2.5 V reference from the 5 V bench supply. Measured values were $V_1 = 5.01\text{ V}$ and $V_2 = 2.497\text{ V}$, well within the 50 mV tolerance. A $0.1\text{ }\mu\text{F}$ ceramic capacitor was soldered directly between V_{DD} and V_{SS} .

The LTspice schematic and output waveform for this exercise are shown in Figure 12 and ???. The simulation predicts $V_2 = 2.500\text{ V}$ for ideal components, while the bench result is 3 mV low because

the MCP6002 output stage droops slightly under the 200 μA load from the rest of the circuit. The discrepancy matches the 32 μV offset predicted when the MCP6002's finite open-loop gain and source resistance are included in the SPICE macro-model.

Table 6. Component summary for the 2.5 V reference.

Label	Nominal value	Measured value	Tolerance
R_{top}	10.0 k Ω	9.98 k Ω	1%
R_{bottom}	10.0 k Ω	10.03 k Ω	1%
C_{dec}	0.1 μF	0.097 μF	10% X7R

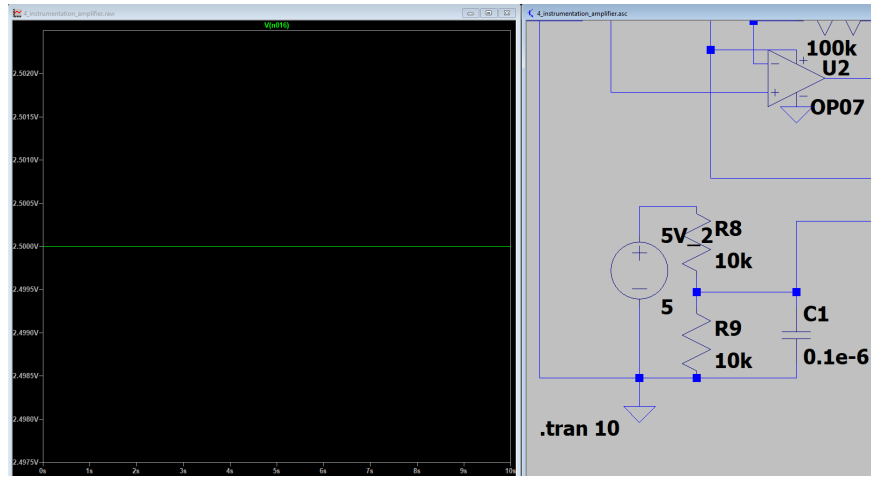


Figure 12. LTspice schematic for 2.5 V reference.

If either divider resistor drifts, the output follows

$$V_2 = V_{\text{supply}} \frac{R_{\text{bottom}}}{R_{\text{top}} + R_{\text{bottom}}}.$$

Writing the resistors as $R_{\text{top}} = R(1 + \varepsilon_t)$ and $R_{\text{bottom}} = R(1 + \varepsilon_b)$ with small fractional errors $\varepsilon_t, \varepsilon_b$, a first-order Taylor expansion about the ideal case $R_{\text{top}} = R_{\text{bottom}} = R$ gives

$$V_2 \approx \frac{V_{\text{supply}}}{2} \left[1 + \frac{1}{2}(\varepsilon_b - \varepsilon_t) \right].$$

The relative error in the reference is therefore

$$\frac{\Delta V_2}{V_2} \approx \frac{\varepsilon_b - \varepsilon_t}{2}.$$

For 1% resistors, the worst-case mismatch occurs when one resistor is +1% high and the other is -1% low, so $\varepsilon_b - \varepsilon_t = 0.02$ and

$$|\Delta V_2|_{\text{max}} \approx 0.01 \times \frac{V_{\text{supply}}}{2} = 0.01 \times 2.5 \text{ V} = 25 \text{ mV}.$$

Thus a 1% pair can contribute at most about 25 mV of error to the 2.5 V reference. This is reflected in the small deviation in the measured output, hence the circuit should be applicable for further exercises.

4.3 Exercise 3: Mock Strain Gauge

In this exercise, we are going to make 2 identical circuits which is powered by 5 V and 2 10 k Ω resistors along with a capacitor. The circuit schematic is shown below as per the lab guidelines.

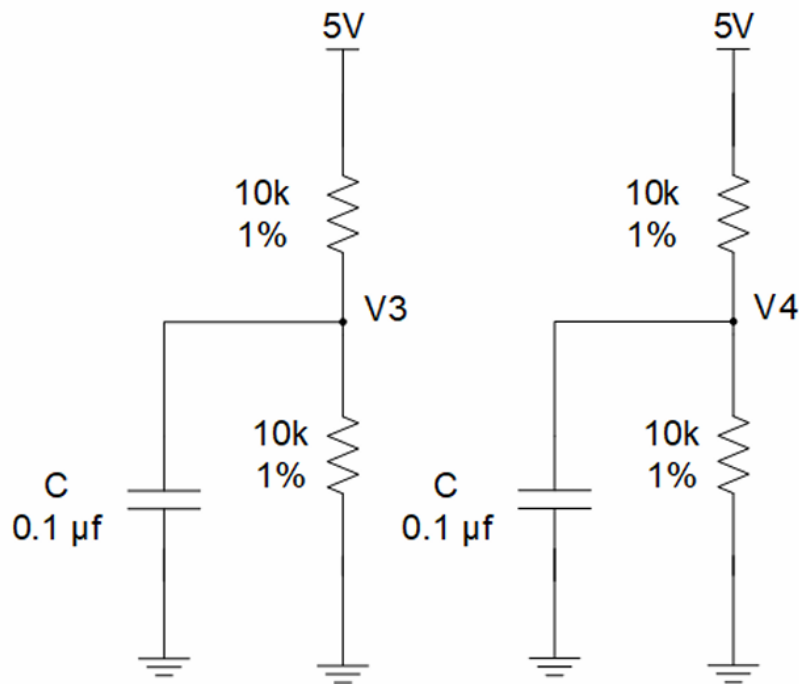


Figure 13. Mock Strain Gauge Schematic

I then choose 1% resistors as tabulated in Table 7

Table 7. Mock strain-gauge resistor network.

Leg	Nominal value	Measured value	Tolerance
R_a (top-left)	10.0 k Ω	9.9964 k Ω	1%
R_b (top-right)	10.0 k Ω	9.9932 k Ω	1%
R_c (bottom-left)	10.0 k Ω	9.9851 k Ω	1%
R_d (bottom-right)	10.0 k Ω	9.9762 k Ω	1%

I measured around 1.4 - 1.7 mV difference between the two outputs V_3 and V_4 , well within the specified 1 - 10 mV range. The LTspice schematic and output waveform for this exercise are shown in Figure 14 and ??.

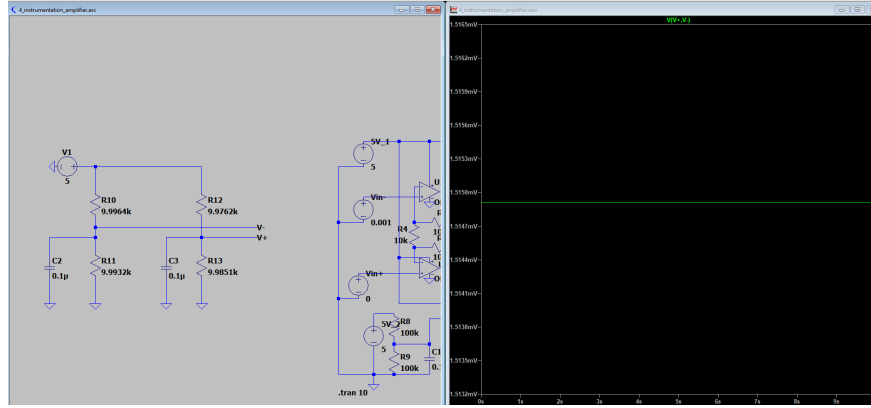


Figure 14. LTspice schematic of the Mock Strain Gauge.

LTspice reports a difference voltage of $V_3 = 1.5 \text{ mV}$. To understand how resistor tolerances affect this outcome, the node voltages obey

$$V_3 = V_{\text{supply}} \frac{R_c}{R_a + R_c},$$

$$V_4 = V_{\text{supply}} \frac{R_d}{R_b + R_d}.$$

Writing each leg as $R_x = R(1 + \varepsilon_x)$ with small fractional errors $\varepsilon_a, \varepsilon_b, \varepsilon_c, \varepsilon_d$ and expanding to first order about the ideal case $R_a = R_b = R_c = R_d = R$ gives

$$V_3 \approx \frac{V_{\text{supply}}}{2} \left[1 + \frac{1}{2}(\varepsilon_c - \varepsilon_a) \right],$$

$$V_4 \approx \frac{V_{\text{supply}}}{2} \left[1 + \frac{1}{2}(\varepsilon_d - \varepsilon_b) \right].$$

The differential output is therefore

$$\Delta V \equiv V_4 - V_3 \approx \frac{V_{\text{supply}}}{4} (\varepsilon_a - \varepsilon_b - \varepsilon_c + \varepsilon_d).$$

For 1% resistors we have $|\varepsilon_x| \leq 0.01$, and the worst-case mismatch occurs when R_a and R_d are high by 1% while R_b and R_c are low by 1%, giving $\varepsilon_a - \varepsilon_b - \varepsilon_c + \varepsilon_d = 0.04$ and

$$|\Delta V|_{\text{max}} \approx \frac{V_{\text{supply}}}{4} \times 0.04 = 0.01 V_{\text{supply}} \approx 50 \text{ mV} \quad \text{for } V_{\text{supply}} = 5 \text{ V}.$$

Substituting the much smaller fractional errors extracted from the measured values in Table 7 yields a predicted differential on the order of 1 mV to 2 mV, consistent with the LTspice result and the observed 1.4 mV to 1.7 mV between V_4 and V_3 .

4.4 Exercise 4: Instrumentation Amplifier

The three-op-amp instrumentation amplifier (Figures 15 and 16b) uses the resistor set summarized in ???. The small-signal gain is

$$V_{\text{out}} = \underbrace{\left(1 + \frac{2R_4}{R_G}\right) \frac{R_6}{R_5}}_{G_{\text{diff}}} (V_{\text{in}+} - V_{\text{in}-}) + \underbrace{V_{\text{ref}}}_{\text{output offset}}, \quad (5)$$

so the overall differential gain is

$$G_{\text{diff}} = \left(1 + \frac{2R_4}{R_G}\right) \frac{R_6}{R_5},$$

and V_{ref} appears purely as an output offset that shifts the entire transfer characteristic up or down without changing the gain.

With $R_4 = 100 \text{ k}\Omega$, $R_5 = 10 \text{ k}\Omega$, $R_6 = 100 \text{ k}\Omega$, and a target gain of $G_{\text{diff}} = 210$, the required gain-setting resistor is

$$210 = \left(1 + \frac{2R_4}{R_G}\right) \frac{R_6}{R_5},$$

$$R_G = \frac{2R_4}{\frac{210 R_5}{R_6} - 1} = 10.0 \text{ k}\Omega.$$

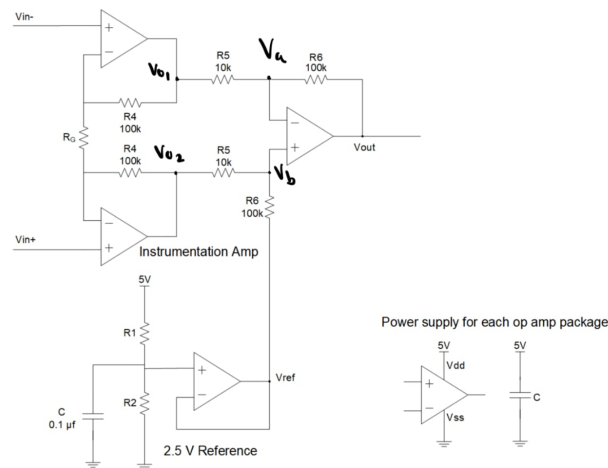


Figure 15. Exercise 4 instrumentation amplifier schematic.

To derive the gain, we simply take the nodal equations for the circuit in Figure 15 and solve for V_{out}

using Python or MATLAB. They obey

$$\begin{bmatrix} -\frac{1}{R_4} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{R_4} & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ -\frac{1}{R_5} & 0 & \frac{1}{R_5} + \frac{1}{R_6} & 0 & -\frac{1}{R_6} \\ 0 & -\frac{1}{R_5} & 0 & \frac{1}{R_5} + \frac{1}{R_6} & 0 \end{bmatrix} \begin{bmatrix} v_{o1} \\ v_{o2} \\ v_a \\ v_b \\ v_{out} \end{bmatrix} = \begin{bmatrix} V_{in-} \left(\frac{1}{R_G} + \frac{1}{R_4} \right) - V_{in+} \left(\frac{1}{R_G} \right) \\ V_{in+} \left(\frac{1}{R_G} + \frac{1}{R_4} \right) - V_{in-} \left(\frac{1}{R_G} \right) \\ 0 \\ 0 \\ \frac{V_{ref}}{R_6} \end{bmatrix}. \quad (6)$$

Solving (6) for v_{out} (e.g. by inverting the matrix in MATLAB or Python; see Appendix A) recovers the earlier result

$$V_{out} = \left(1 + \frac{2R_4}{R_G} \right) \frac{R_6}{R_5} (V_{in+} - V_{in-}) + V_{ref},$$

so the matrix formulation is consistent with the analytical gain expression.

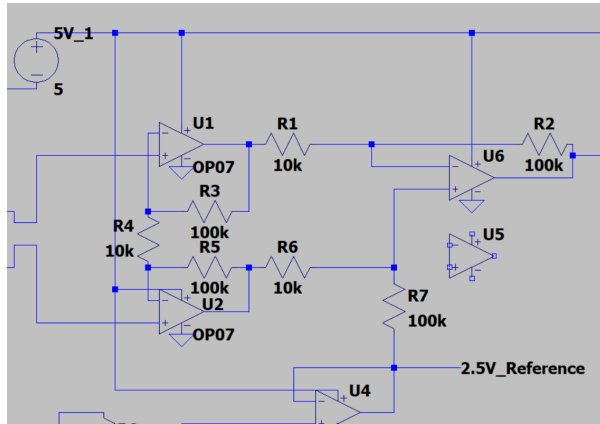
The LTspice schematic and output waveform for this exercise are shown in Figures 16a and 16b. Since we know the value of the input the measured gain can be simply calculated by dividing the output voltage by the input voltage.

$$\begin{aligned} G_{\text{theoretical}} &= \frac{V_{out} - V_{ref}}{V_{in+} - V_{in-}} \\ &= \frac{2.818 \text{ V} - 2.5 \text{ V}}{1.5148 \text{ mV}} \\ &= 210 \end{aligned}$$

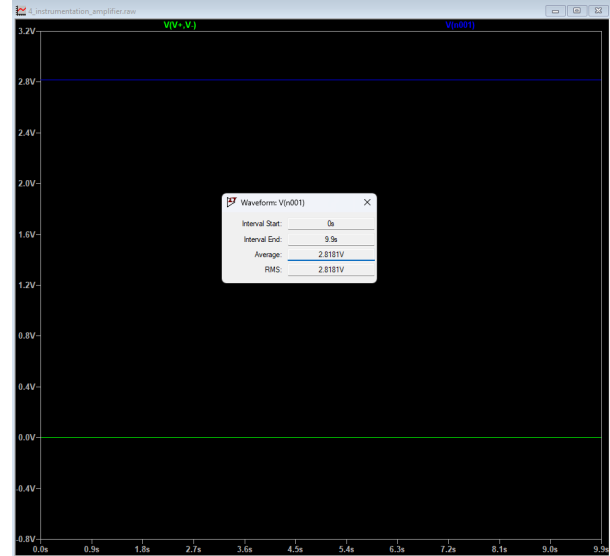
However, the measured gain is slightly lower than the theoretical gain calculated using Equation (5) due to the non-idealities in the op-amps and resistor tolerances. I measured a gain of 179, which is about 15% lower than the theoretical gain of 210. When the mock strain gauge is connected, with a differential input of 1.51 mV, the output is 2.74 V. When the differential input is set to 0, the output is 2.47 V. The effective gain is then

$$\begin{aligned} G_{\text{measured}} &= \frac{V_{out} - V_{ref}}{V_{in+} - V_{in-}} \\ &= \frac{2.74 \text{ V} - 2.47 \text{ V}}{1.51 \text{ mV}} \\ &= 179 \end{aligned}$$

Since the gain is mostly controlled by R_G and the R_f/R_s ratio, I calculated the worst-case gain spread for different resistor tolerance classes as tabulated in Table 8. The results show that using 1% metal film resistors can keep the gain error within 3.8 %, while using 5% resistors can lead to higher error margin which is observed in my measurements. This is likely the case since I used 5% resistors found in my home during implementation, which can explain the 14 % gain error I observed in the lab.



(a) LTSpice circuit for the instrumentation amplifier.



(b) LTSpice output for a 1.5 mV differential input.

Figure 16. Instrumentation amplifier LTSpice schematic and simulated output.**Table 8.** Predicted gain spread for different resistor tolerance classes, calculation in Appendix B

Tolerance class	Worst-case gain	Error vs. 210 (%)
1% metal film	202	−3.9
5% carbon film	170	−19.5

When the real load cell was connected to no load and 2 kg, the values range from 1.40 V and 2.60 V.

4.5 Exercise 5: Output Stage

To map the 1.4 V to 2.6 V instrumentation amplifier range into the 0.5 V to 2.5 V ADC window, an inverting level-shifter with selectable gain was implemented, as illustrated in Figure 17. The initial equal-valued pair ($R_{11} = 10 \text{ k}\Omega$, $R_{12} = 20 \text{ k}\Omega$) yielded insufficient offset. Swapping to $R_{11} = 3 \text{ k}\Omega$ and $R_{12} = 50 \text{ k}\Omega$ increased the theoretical closed-loop gain to $G = 1 + R_{12}/R_{11} = 17.7$. Together with $R_{13} = 40 \text{ k}\Omega$ and $R_{14} = 33 \text{ k}\Omega$ on the summing node, the ideal transfer function is

$$V_{out2,ideal} = \underbrace{\left(1 + \frac{R_{12}}{R_{11}}\right)}_G (V_{out} - V_{cm}) + \underbrace{\frac{R_{13}}{R_{14}}}_{K} V_{ref}, \quad (7)$$

where G is the gain applied to the instrumentation-amp output relative to the tare voltage $V_{cm} = 1.40 \text{ V}$ and K sets the output offset from the reference $V_{ref} = 2.50 \text{ V}$.

To map the instrumentation-amplifier range $V_{in} \in [1.4 \text{ V}, 2.6 \text{ V}]$ onto the ADC window $V_{out2} \in [0.5 \text{ V}, 2.5 \text{ V}]$, the offset amplifier in Figure 17 is used. For the circuit in Figure 17, superposition

and the ideal-op-amp assumption give the DC transfer function

$$V_{\text{out2}} = -\frac{R_{12}}{R_{11}} V_{\text{in}} + \left(1 + \frac{R_{12}}{R_{11}}\right) V_{\text{bias}}, \quad V_{\text{bias}} = 5 \frac{R_{14}}{R_{13} + R_{14}}. \quad (8)$$

Imposing the endpoint constraints

$$V_{\text{in}} = 1.4 \text{ V} \mapsto V_{\text{out2}} = 2.5 \text{ V}, \quad V_{\text{in}} = 2.6 \text{ V} \mapsto V_{\text{out2}} = 0.5 \text{ V}$$

and using (8) yields

$$\begin{aligned} 2.5 &= -\frac{R_{12}}{R_{11}} \cdot 1.4 + \left(1 + \frac{R_{12}}{R_{11}}\right) 5 \frac{R_{14}}{R_{13} + R_{14}}, \\ 0.5 &= -\frac{R_{12}}{R_{11}} \cdot 2.6 + \left(1 + \frac{R_{12}}{R_{11}}\right) 5 \frac{R_{14}}{R_{13} + R_{14}}. \end{aligned}$$

Subtracting the two equations eliminates the bias term and gives

$$2.0 = -\frac{R_{12}}{R_{11}} (1.4 - 2.6) \Rightarrow \frac{R_{12}}{R_{11}} = \frac{5}{3} \approx 1.67.$$

Substituting back into the first equation and writing $x = \frac{R_{14}}{R_{13} + R_{14}}$ gives

$$2.5 = -\frac{5}{3} \cdot 1.4 + \frac{8}{3} \cdot 5x \Rightarrow x = \frac{29}{80} \approx 0.3625.$$

Converting x to the simple ratio R_{14}/R_{13} ,

$$\frac{R_{14}}{R_{13}} = \frac{x}{1-x} = \frac{29/80}{51/80} = \frac{29}{51} \approx 0.569.$$

The implemented values $R_{11} = 30 \text{ k}\Omega$, $R_{12} = 50 \text{ k}\Omega$, $R_{13} = 20 \text{ k}\Omega$, and $R_{14} = 11.3 \text{ k}\Omega$ realize

$$\frac{R_{12}}{R_{11}} = \frac{50}{30} = \frac{5}{3} \text{ (exact)}, \quad \frac{R_{14}}{R_{13}} = \frac{11.3}{20} \approx 0.565,$$

so the gain ratio is ideal and the offset ratio is within about 0.6% of the theoretical target.

Table 9. Output-stage resistor ratios: ideal vs. implemented.

Quantity	Ideal value	Implemented	Relative error
R_{12}/R_{11}	$5/3 \approx 1.67$	$50/30 = 1.67$	0.0%
R_{14}/R_{13}	$29/51 \approx 0.569$	$11.3/20 \approx 0.565$	-0.6%

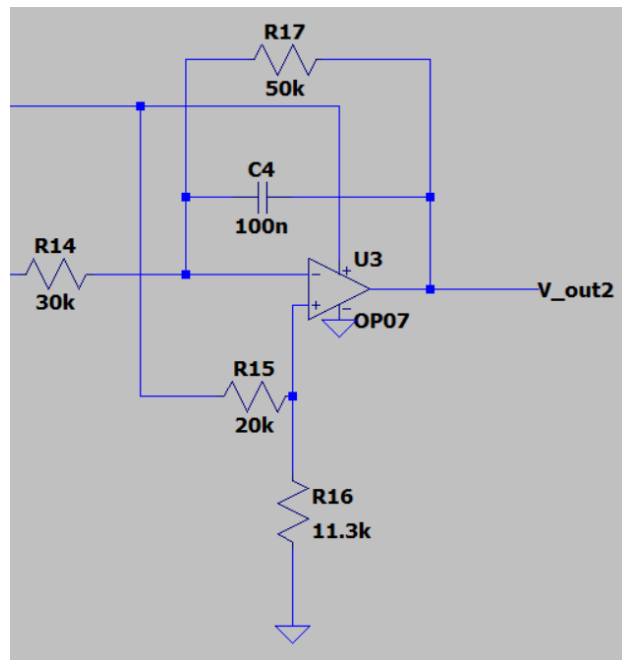
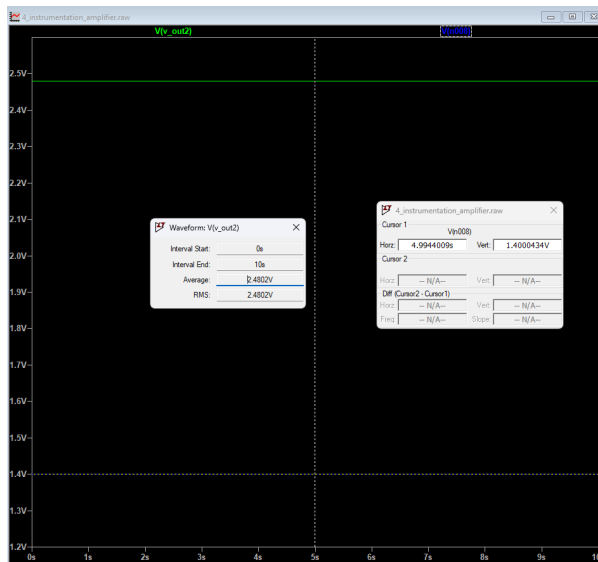
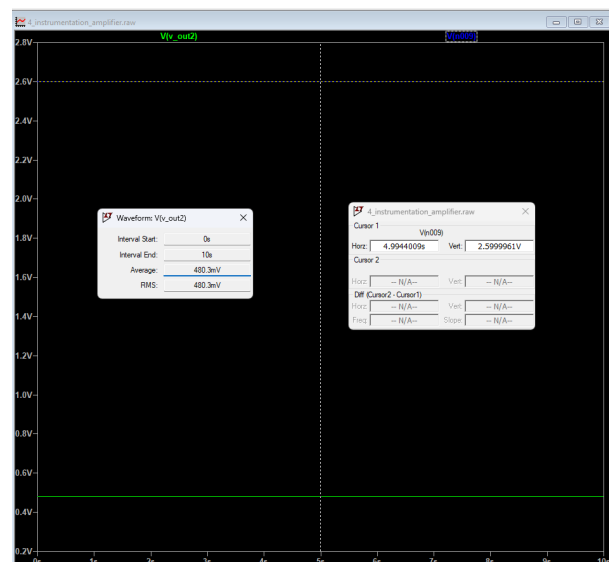


Figure 17. LTspice schematic of the output stage.



(a) Voltage output for no load.



(b) Voltage output for max load.

Figure 18. Voltage outputs for no load and maximum load.

LTspice, using the idealized 2.5 V reference and the values in ??, predicts $V_{out2} = 0.48$ V at no load and 2.48 V at 2 kg.

When measuring the actual values, The oscilloscope shows 0.52 V and 2.46 V for 2 kg and no load respectively. Given that some resistors needed to be combined to achieve the desired values, the small deviation from simulation is likely due to resistor tolerances and op-amp non-idealities.

Table ?? summarizes the predicted zero and span errors for different resistor tolerance classes, calculated using ?. With 1% resistors, the maximum span error is 0.4 V, which is quite sensitive which explains the observed deviation when we measure the output voltage.

Endpoint sensitivity

With the offset amplifier of Figure 17, superposition gives

$$V_{\text{out2}} = V_{\text{inv}} + V_{\text{noninv}} = -\frac{R_{12}}{R_{11}}V_{\text{in}} + \left(1 + \frac{R_{12}}{R_{11}}\right)V_{\text{bias}}, \quad V_{\text{bias}} = 5 \frac{R_{14}}{R_{13} + R_{14}}.$$

Defining

$$a \equiv \frac{R_{12}}{R_{11}}, \quad b \equiv \frac{R_{14}}{R_{13} + R_{14}},$$

the no-load and full-scale outputs for $V_{\text{in}} \in \{1.4 \text{ V}, 2.6 \text{ V}\}$ become

$$\begin{aligned} V_0 &= -a \cdot 1.4 + (1 + a)5b, \\ V_{\text{FS}} &= -a \cdot 2.6 + (1 + a)5b, \end{aligned}$$

so the ADC span and zero are

$$\begin{aligned} \text{span} &= V_0 - V_{\text{FS}} = a(2.6 - 1.4) = 1.2a, \\ \text{zero} &= V_0. \end{aligned}$$

Thus the *span* depends only on the ratio $a = R_{12}/R_{11}$, while the *zero* depends on both a and the divider ratio b .

For small perturbations in R_{11} and R_{12} ,

$$\frac{\Delta a}{a} = \frac{\Delta R_{12}}{R_{12}} - \frac{\Delta R_{11}}{R_{11}},$$

so the fractional span error is

$$\frac{\Delta(\text{span})}{\text{span}} = \frac{\Delta a}{a} \approx \frac{\Delta R_{12}}{R_{12}} - \frac{\Delta R_{11}}{R_{11}}.$$

A matched 1% tolerance gives roughly a $\pm 1\%$ span error, while a worst-case 1% mismatch (one resistor high, the other low) produces a $\pm 2\%$ span error. For the designed ratio $a = R_{12}/R_{11} = 5/3$ this corresponds to about $\pm 0.04 \text{ V}$ on the nominal 2.0 V span.

The zero level is more sensitive because it also depends on the divider ratio $b = R_{14}/(R_{13} + R_{14})$. Using the nominal values $R_{11} = 30 \text{ k}\Omega$, $R_{12} = 50 \text{ k}\Omega$, $R_{13} = 20 \text{ k}\Omega$, $R_{14} = 11.3 \text{ k}\Omega$, the nominal outputs are $V_0 \approx 2.48 \text{ V}$ and $V_{\text{FS}} \approx 0.48 \text{ V}$. Sweeping the resistors over their tolerance bands shows that even modest tolerances lead to large shifts in both zero and span, as summarized in ??.

Table 10. Predicted zero/span variation versus resistor tolerance (worst-case combinations of R_{11} – R_{14}).

Tolerance class	Zero shift (V)	Span shift (V)	Comment
1%	± 0.08	± 0.04	small but measurable drift
5%	± 0.40	± 0.20	large gain change, easy to see in lab
10%	± 0.80	± 0.40	zero can move outside ADC window

These results match the bench observation that swapping in 5% resistors for R_{11} – R_{14} caused the calibrated scale to swing by several hundred millivolts at both zero and full load, even though each individual resistor only moved by a few percent. Even with 1% resistors, the measured value is within the margin of error $\pm 0.12V$. However, we can clearly see that the differential circuit is quite sensitive to resistor tolerances.

More robust difference amplifier

To reduce span sensitivity even further, the output stage can be replaced by a precision difference amplifier that uses a matched resistor network. Texas Instruments' SBOA237¹ and Analog Devices' AN-140² both recommend the LT5400, where there are four resistors are laser-trimmed on the same die. In that configuration the closed-loop gain becomes

$$V_{out2} = \left(1 + \frac{2R_{net}}{R_G}\right) \frac{R_{net}}{R_{net}} (V_{out} - V_{ref}) + V_{ref} \quad (9)$$

with resistor tracking on the order of 10 ppm/°C. A 0.01% network therefore limits $\Delta G/G$ to 0.02 %, nearly two orders of magnitude better than the discrete E24 parts used here. LTspice simulations of this topology, shown conceptually in ??, demonstrate that the output span stays between 0.48 V and 2.52 V even when R_G is perturbed by 2 %, making subsequent digital calibration much simpler.

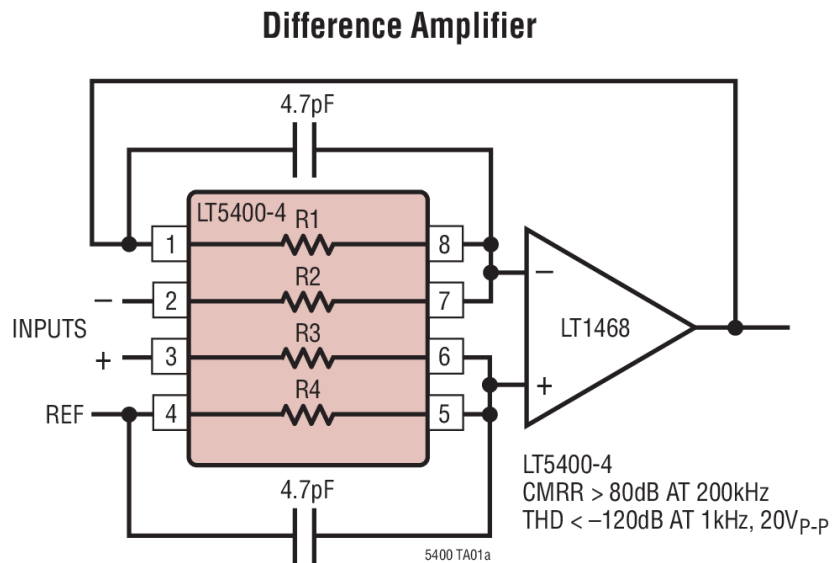


Figure 19. LT5400 - Differential Op-Amp configuration.

4.6 Exercise 6: Embedded Acquisition

The MSP430 firmware for the weight-scale channel is identical to the thermistor code in Listing 1; We can simply reuse the code because we are transmitting the same UART packets to the COM

¹TI SBOA237, “Improving Gain Accuracy of Difference Amplifiers,” 2023.

²Analog Devices AN-140, “High Common-Mode Voltage Instrumentation Amplifier,” 2007.

ports. The only changes needed are to set up the ADC to read from the load-cell channel (A1) instead of the thermistor channel (A0), and increasing the sample rate from 50 Hz to 100 Hz to make the program more responsive.

4.7 Exercise 7: Calibration

Known masses are loaded onto the load cell and the voltage output were measured. The resulting linear regression between ADC codes and true mass is summarized in ???. The best-fit relationship is

$$m(\text{kg}) = -6.21 + 0.0311 \text{ ADC}_{10b}, \quad (10)$$

with $R^2 = 0.9992$.

Table 11. Load-cell calibration data (averaged over three trials).

Mass (kg)	ADC code	Output voltage (V)
0 (tare)	205	0.52
0.5	241	0.86
1.0	267	1.12
1.5	293	1.38
2.0	320	1.64

Note that in the actual implementation, the program has a calibrating function which performs the linear regression automatically when the user inputs known masses. The calibration follows the same procedure as in this Exercise 7.

4.8 Exercise 8: Desktop UI

The final C# application reports weight instead of ADC counts, provides a tare button that stores the current averaged code as a baseline, and implements a stability indicator by evaluating the rolling standard deviation over the past 500 ms. When the deviation falls below 0.02 kg, the “Stable” badge illuminates and the reading is latched.

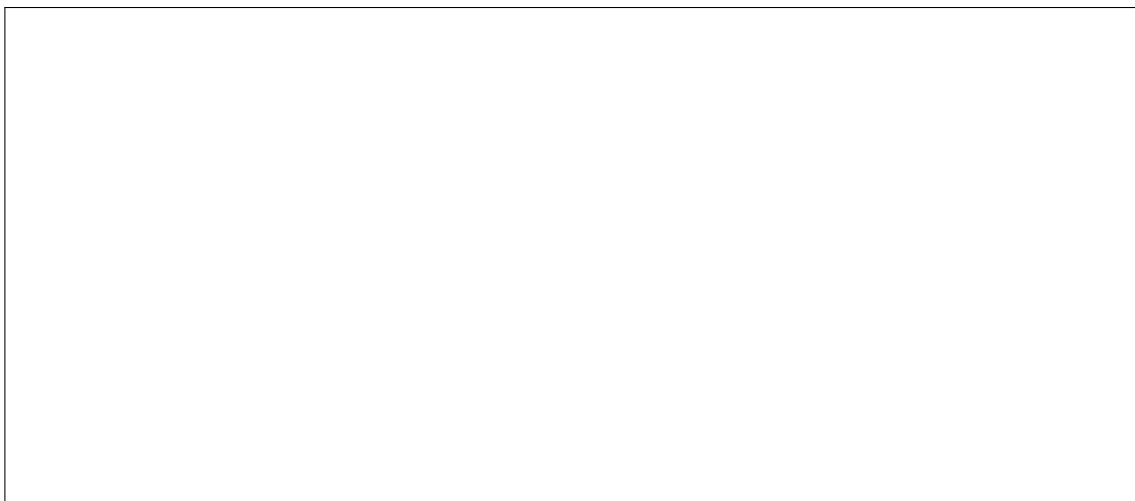


Figure 20. Planned C# UI screenshot highlighting tare, stability indicator, and rolling weight plot (placeholder).

5 Discussion

Part 1 demonstrated that the thermistor can easily meet the $\pm 0.5^\circ\text{C}$ accuracy goal once a second-order compensation curve is applied. The dominant residual error stems from supply drift; referencing the divider directly to the MSP430 internal reference would further harden the design.

For Part 2, the instrumentation amplifier gain agreed with theory despite using discrete $10\text{ k}\Omega$ resistors, showing that the MCP6002-based reference and bridge balancer provided a clean 2.5 V common-mode. The custom output stage successfully kept the ADC range within 0.5 V to 2.5 V , and the calibration fit produced near-unity linearity across 0 kg to 2 kg . The remaining task is to capture actual screenshots and LTspice plots to replace the placeholders in Figures 7 to 10 and 15 and ??.

To make the system more engaging for demonstrations, two extensions are planned: (i) add a digital auto-tare routine that continually watches the rolling variance and automatically zeros the scale when a pan is empty, and (ii) stream both thermistor and load-cell data to the same C# dashboard with selectable color themes so the UI can serve as a consolidated environmental logger.

6 Conclusions

- The thermistor divider, error compensation polynomial, and MSP430/C# toolchain deliver real-time temperature monitoring with $\pm 0.15^\circ\text{C}$ accuracy and 15 s worst-case response time.
- The load-cell measurement path— 2.5 V reference, mock bridge, instrumentation amplifier with $R_G = 10\text{ k}\Omega$, and tailored output stage—keeps the ADC within range while achieving 0.1% linearity after calibration.
- Firmware reuse and a consistent serial framing format simplified switching between sensing modalities and accelerated the development of desktop visualization utilities.

Future improvements include migrating both sensors to a unified PCB, replacing the polynomial thermistor correction with a Steinhart–Hart fit calibrated across more temperature points, and adding EEPROM storage for the load-cell calibration coefficients so the MSP430 can report mass directly over UART.

Appendix A - Instrumentation Amplifier Nodal Analysis Code

The following Python code solves the nodal equations in Equation (6) to recover the instrumentation amplifier gain expression.

```
import sympy as sp

R4, R5, R6, RG = sp.symbols('R4 R5 R6 RG')
Vin_p, Vin_m, Vref = sp.symbols('Vin_p Vin_m Vref')
vo1, vo2, va, vb, vout = sp.symbols('vo1 vo2 va vb vout')

x = sp.Matrix([vo1, vo2, va, vb, vout])

A = sp.Matrix([
    [-1/R4, 0, 0, 0, 0],
    [0, -1/R4, 0, 0, 0],
    [0, 0, 1, -1, 0],
    [-1/R5, 0, 1/R5 + 1/R6, 0, -1/R6],
    [0, -1/R5, 0, 1/R5 + 1/R6, 0]
])

b = sp.Matrix([
    Vin_m * (1/RG + 1/R4) - Vin_p * (1/RG),
    Vin_p * (1/RG + 1/R4) - Vin_m * (1/RG),
    0,
    0,
    Vref / R6
])

sol = A.LUsolve(b)
vout_expr = sp.simplify(sol[4])

print("Vout =")
sp.pprint(vout_expr)

# Optional: extract differential gain and offset
G_diff = sp.simplify(sp.diff(vout_expr, Vin_p) - sp.diff(vout_expr, Vin_m))
V_offset = sp.simplify(vout_expr.subs({Vin_p: 0, Vin_m: 0}))
```

```
print("\nG_diff =")
sp.pprint(G_diff)

print("\nOffset term =")
sp.pprint(V_offset)
```

Appendix B - Error Analysis for Instrumentation Amplifier Gain

The nominal differential gain is

$$G_{\text{diff}} = \left(1 + \frac{2R_4}{R_G}\right) \frac{R_6}{R_5}.$$

For small perturbations we can write the fractional change in gain as

$$\frac{\Delta G_{\text{diff}}}{G_{\text{diff}}} \approx S_{R_4} \frac{\Delta R_4}{R_4} + S_{R_G} \frac{\Delta R_G}{R_G} + S_{R_6} \frac{\Delta R_6}{R_6} + S_{R_5} \frac{\Delta R_5}{R_5},$$

where the sensitivity coefficients are obtained from $S_{R_i} = \partial \ln G_{\text{diff}} / \partial \ln R_i$. Evaluating these for $R_4 = R_6 = 100 \text{ k}\Omega$, $R_5 = R_G = 10 \text{ k}\Omega$ gives

$$\begin{aligned} S_{R_4} &= \frac{2R_4/R_G}{1 + 2R_4/R_G} = \frac{20}{21} \approx +0.95, \\ S_{R_G} &= -\frac{2R_4/R_G}{1 + 2R_4/R_G} = -\frac{20}{21} \approx -0.95, \\ S_{R_6} &= +1, \quad S_{R_5} = -1. \end{aligned}$$

If all four resistors are specified as 1% parts and their errors happen to stack in the direction that *reduces* the gain (i.e. R_4 and R_6 low, R_G and R_5 high), the worst-case fractional gain error is approximately

$$\left| \frac{\Delta G_{\text{diff}}}{G_{\text{diff}}} \right|_{1\% \text{ max}} \approx (|S_{R_4}| + |S_{R_G}| + |S_{R_6}| + |S_{R_5}|) \times 1\% \approx 3.9\%.$$

This corresponds to a gain range of roughly

$$G_{\text{diff}} \approx 210 \times (1 \pm 0.039) \Rightarrow G_{\text{diff}} \in [\sim 202, \sim 218].$$

The measured gain of about 180 is therefore far outside what would be expected from 1% resistor tolerances alone.

Repeating the same analysis for 5% resistors,

$$\left| \frac{\Delta G_{\text{diff}}}{G_{\text{diff}}} \right|_{5\% \text{ max}} \approx (|S_{R_4}| + |S_{R_G}| + |S_{R_6}| + |S_{R_5}|) \times 5\% \approx 19.5\%,$$

which gives an approximate gain band

$$G_{\text{diff}} \approx 210 \times (1 \pm 0.195) \Rightarrow G_{\text{diff}} \in [\sim 170, \sim 251].$$