Linear Opamp 07 : Instrumentation Amplifier

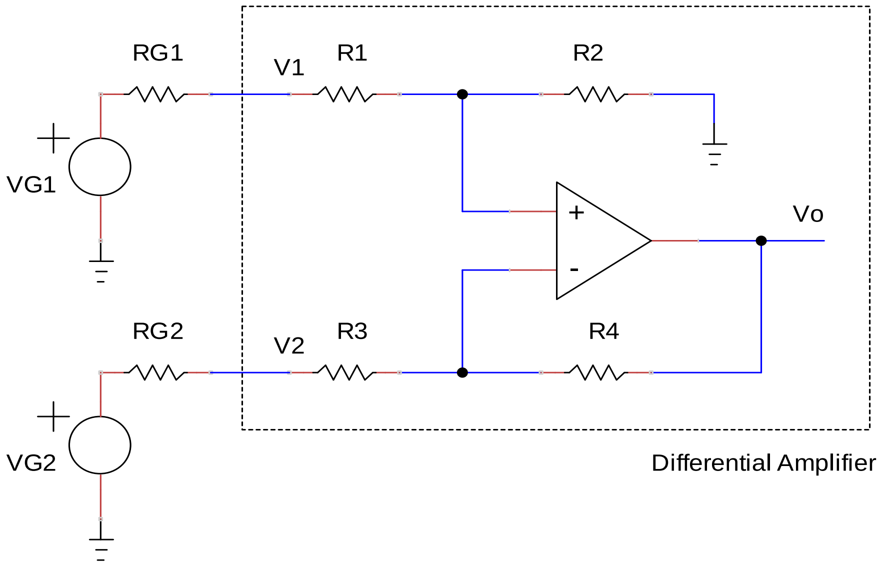
This project describes the Instrumentation Amplifier that is an improvement over the Differential Amplifier.

BOM	1x Dual Opamp MCP6002 Resistors: 2x 1 kΩ, 2,2 kΩ, 2x 5,6 kΩ, 2x 10 kΩ , 22 kΩ, 4x 4,7 kΩ or 2x 4,7 kΩ and 2x 3,3 kΩ Diodes: 2x 1N4148

Index

The Instrumentation Amplifier

The differential amplifier is a good circuit to amplify the difference of two input signals. However, due to its finite input resistance, it requires the input voltages to be independent on the input current. In a practical case where the input voltages VG1 and VG2 have associated source resistances RG1 and RG2 the voltages V1 and V2, as seen on the amplifier inputs, are not the same as the source voltages.



Moreover, as the common mode rejection ratio (CMRR) depends in ideal conditions to satisfy:

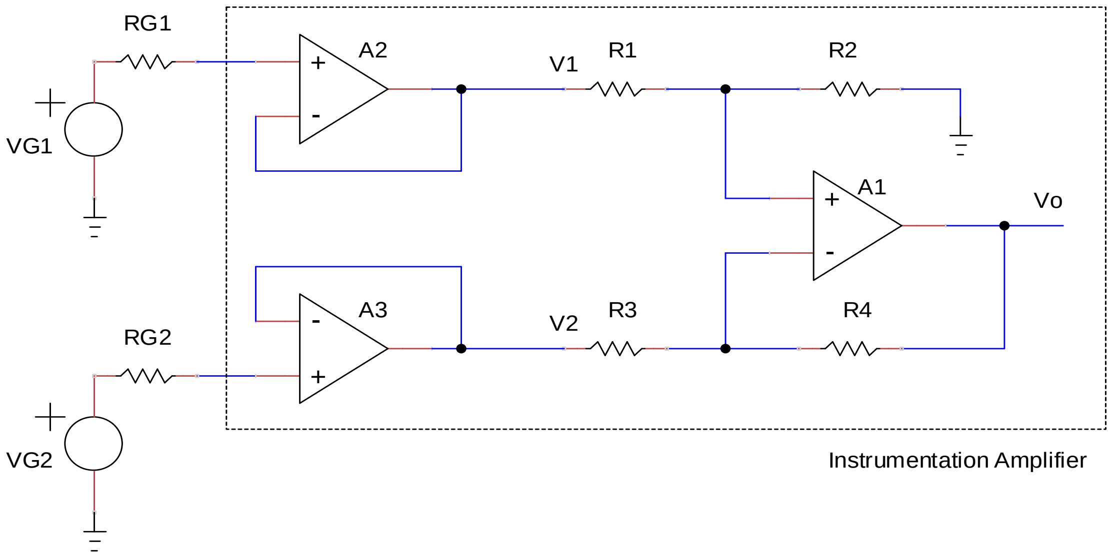
$$R1 \cdot R4 = R2 \cdot R3$$

That condition is easy to satisfy in ideal conditions, but when we add the source resistances RG1 and RG2 we have to verify:

$$(RG1 + R1) \cdot R4 = R2 \cdot (RG2 + R3)$$

And that is not easy to guarantee, especially if the details of the sources are not completely known on the design stage or they have temperature or other ambient dependences that are not common to the amplifier R1 to R4 resistors.

The first improvement over the differential amplifier is adding to the original A1 operation amplifier two amplifiers A2 and A3 on its inputs, in follower mode, so that the input voltages V1 and V2 are not dependent on the source voltages.



We can see that, as current inputs on the operational amplifiers are ideally zero, then:

$$V1 = VG1 \qquad V2 = VG2$$

This is the simplest Instrumentation Amplifier, or AI for short. In general, an AI shall have, at least, the following requirements:

- Constant, known, differential amplification
- High Common Mode Rejection (CMRR)
- High input resistance

Using ideal opamps, if we satisfy the matching equation:

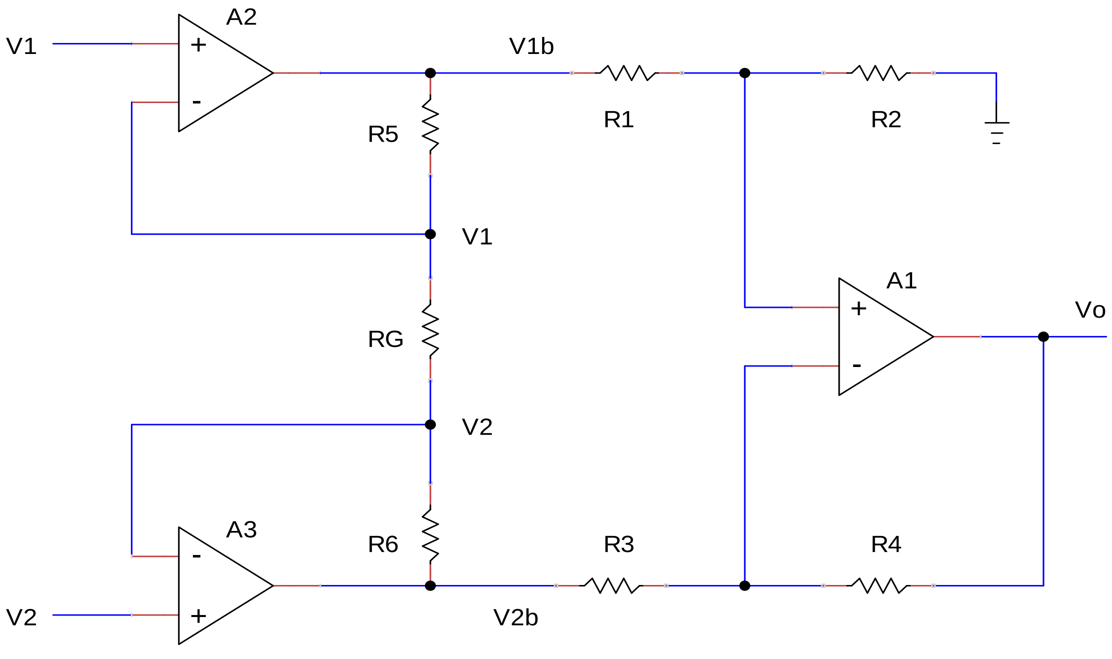
$$R1 \cdot R4 = R2 \cdot R3$$

Then:

- Differential amplification is R2/R1
- CMRR is infinite
- Input resistance is infinite

We can use the above circuit as a simple AI, but usually we also want to be able to change the gain. In the differential amplifier gain is R2/R1, but as we also need to satisfy the matching equation, we cannot change R2 alone; we need to change R4 at the same time and with the same value.

The following circuit improves the above one so that we can control the amplifier gain by changing the value of only one resistor. We have taken the sources from the circuit, to simplify the diagram, as they do not affect the circuit operation.



Observe that, due to the virtual short circuit on A2 and A3, the voltage on the two terminals of RG are the input voltages of the amplifier V1 and V2.

As the output resistance of A2 and A3, if they are ideal, is zero, we don't disturb the matching equation of the differential amplifier defined by A1, R1, R2, R3 and R4. So, the output of the full amplifier will be:

$$V_O = \frac{R2}{R1}(V1b - V2b) = A_{d2}(V1b - V2b)$$

We have defined the differential gain of the differential amplifier as A_{d2}.

We can now calculate the voltages on the inputs of the differential amplifier V1b and V2b.

We can also define an input stage gain A_{d1} as the ratio of the differential amplifier differential voltage V1b-V2b to the input differential voltage V1-V2.

$$A_{d1} = \frac{V1b - V2b}{V1 - V2}$$

That way, the output of the AI will be:

$$V_O = A_{d1}A_{d2}(V1 - V2) = A_d(V1 - V2) = A_d V_d$$

So the full AI gain is the product of the first stage and second stage gains.

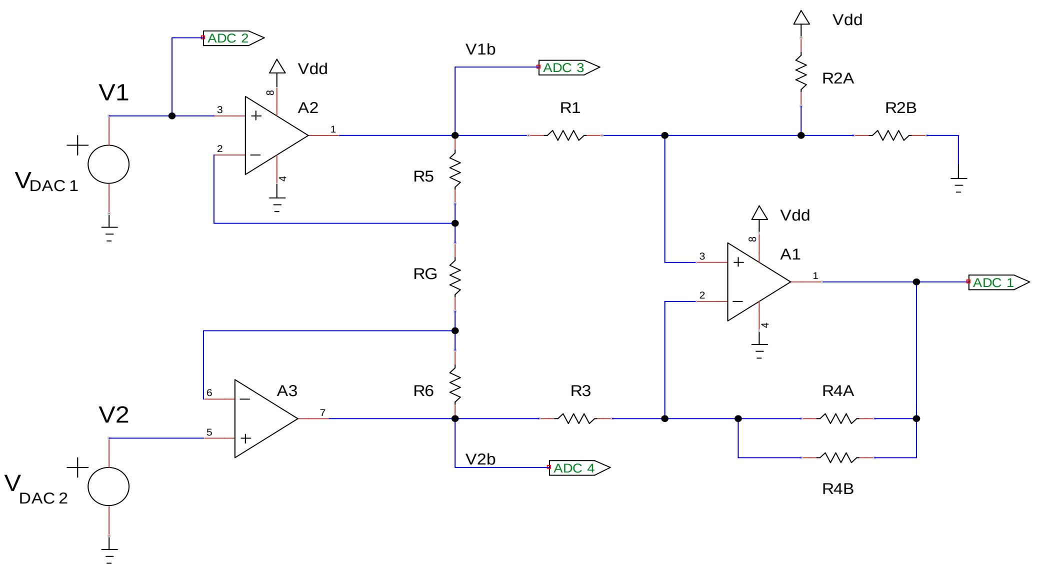
$$A_d = A_{d1} A_{d2}$$

1	Obtain the voltages V1b and V2b as function of V1, V2, R5, RG and R6. Use the virtual short circuit on A2 and A3. Obtain the input stage differential gain A_{d1}. Obtain the full IA gain.

Observe that we can change the differential gain of the full amplifier just by changing one resistor. We usually set R5 equal to R6 and change RG to change the amplification. See that gain is the same as the original differential amplifier if we make RG infinite, and that we get higher gain the lower the RG value we use.

Measuring the gain

We will now test the amplifier. The circuit to implement, together with the DAC and ADC connections is in the figure below. We will use one MC6002 IC for opamp A1 and another MCP6002 for opamps A2 and A3.



The differential amplifier will use:

$$R1 = R3 = 1\ k\Omega \qquad R2A = R2B = R4A = R4B = 4.7\ k\Omega$$

If you don't have four equal 4.7 kΩ resistors you can use:

$$R1 = R3 = 1\ k\Omega \qquad R2A = R4A = 4.7\ k\Omega \qquad R2B = R4B = 3.3\ k\Omega$$

The input stage will use the resistors:

$$R5 = R6 = 10\ k\Omega \qquad RG = 22\ k\Omega$$

	2	Obtain, for the selected resistor values, the gains of the first stage A_{d1}, the second stage A_{d2} and the full AI A_d.

Now it is time to take some measurements. After building the circuit we will set V1 to a 50mV amplitude 100 Hz sine wave centered at the common voltage Vdd/2. We will also set V2 to another 50mV sine wave but with opposite phase. That way we will have a 100mV differential voltage between V1 and V2.

Open the Python interpreter and don't forget to import the slab module and connect to the board. The common voltage to use can be calculated:

```
>>> vcm=slab.vdd/2
>>> vcm
```

That enables us to set the DAC1 and DAC2 values:

```
>>> slab.waveSine(vcm-0.05,vcm+0.05,100)
>>> slab.setWaveFrequency(100)
>>> slab.waveSine(vcm+0.05,vcm-0.05,100,second=True)
```

Now we want to store 5 sine cycles (500 points) and store the results on several variables to perform postprocessing. We will use the variable names associated to the ADC1 to ADC4 nodes. As we are using waveforms both in DAC 1 and DAC 2 we need to use the ***dual*** parameter on the ***wavePlot*** command.

```
>>> slab.tranStore(500,4)
>>> t,vo,v1,v1b,v2b=slab.wavePlot(dual=True,returnData=True)
```

	3	Perform the requested measurements. Check that the voltages at ADC1 to ADC4 have the expected values.

We can now dismiss the plot and work with the measurement data. We can now see the differential signals at the input, at the intermediate stage (v1b, v2b) and at the output.

```
>>> slab.plot1n(t,[2*(v1-vcm),v1b-v2b,vo-vcm]\
... ,labels=["Vd_in","Vd_b","Vd_Out"])
```

We can also calculate the gain of the first stage. Observe that the differential input voltage amplitude is not V1 amplitude but two times V1 amplitude as V2 has the same signal with opposite phase. Gain is then:

$$Ad1 = \frac{(V1b - V2b)_{peak\ 2peak}}{V1_{peak\ 2peak}} \cdot \frac{1}{2}$$

```
>>> Ad1=slab.peak2peak(v1b-v2b)/slab.peak2peak(v1)/2
>>> Ad1
```

And we can also calculate the gain of the second stage:

```
>>> Ad2=slab.peak2peak(vo)/slab.peak2peak(v1b-v2b)
>>> Ad2
```

And the full gain of the AI:

```
>>> Ad=slab.peak2peak(vo)/slab.peak2peak(v1)/2
>>> Ad
```

	4	Perform the requested measurements. Check that the gains are similar to the calculated ones.

Input differential voltage limits

The output voltage of our AI is:

$$V_O = V_r + A_d V_d$$

Amplifier A1, in our circuit, implemented with an MCP6002 is rail to rail, that means that Vo is only limited by the supply voltages:

$$0 \leq V_O \leq V_{dd}$$

	5	Determine the maximum Vdmax (positive) and minimum Vdmin (negative) V_d values that guarantee that V_O complies with the above requirement. Calculate those limits for the AI we have built and measured.

We will try to generate a signal just on this calculated limit.

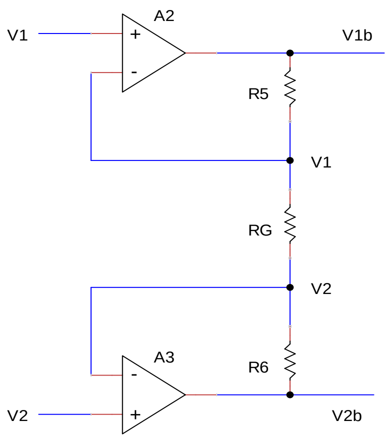
This time we only need to store information about V1 (ADC2) and Vo (ADC1). Remember that vdmax is positive and vdmin is negative.

```
>>> slab.waveSine(vcm+vdmin/2,vcm+vdmax/2,100)
>>> slab.waveSine(vcm+vdmax/2,vcm+vdmin/2,100,second=True)
>>> slab.tranStore(500,2)
>>> slab.wavePlot(dual=True)
```

If you have performed the calculations right, you should see an output voltage waveform that just reaches the Vdd and GND voltages at its peaks.

	6	Check the Vd limits for the indicated case.

The previously calculated limit is the basic limit for V_d. In our circuit, however, the input stage can add additional limits on Vd. The following figure shows the input stage of the AI.



You can see that, unless RG is infinite, if V1 is greater than V1, V1b will be greater than V1 and V2b will be less than V2. As V1b and V2b are limited by the output voltage range of amplifiers A1 and A2, the voltage range at V1 and V2 is limited.

Let's suppose, like in our circuit, that R5 is equal to R6. We have calculated at 1 the value of V1b and V2b voltages as function of V1, V2, R5, RG and R6. We also know the gain of the first stage as A_{d1} function of the component values. Finally, we know the relationship between V1, V2 with Vcm and Vd.

$$V_1 = V_{cm} + \frac{V_d}{2} \qquad V_2 = V_{cm} - \frac{V_d}{2}$$

	7	Obtain V1b and V2b as function of V1, V2 and Ad1. Rewrite V1b and V2b as function of Vcm, Vd and Ad1.

In the same way that opamp A1 output voltage was limited between Vdd and GND, the same is true for opamps A2 and A3. We have just calculated V1b and V2b as function of Vcm, Vd and Ad1. We can now apply the supply restrictions:

$$0 \leq V_{1b} \leq V_{dd} \qquad\qquad 0 \leq V_{2b} \leq V_{dd}$$

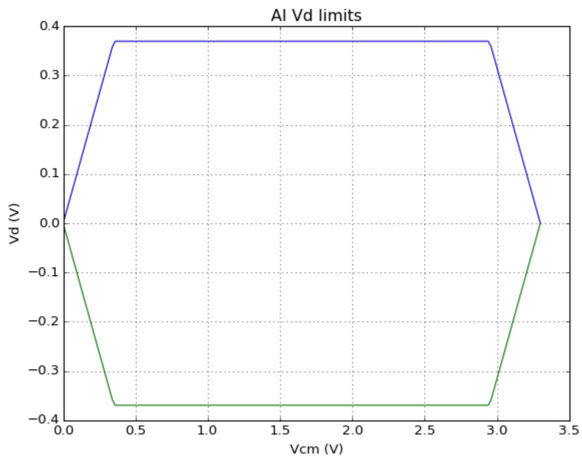
As a result you should obtain something like with k being a constant value that depends on Ad1:

$$\left.\begin{matrix} -k \cdot Vcm \\ -k(Vdd - Vcm) \end{matrix}\right\} \leq Vd \leq \left\{\begin{matrix} k \cdot Vcm \\ k(Vdd - Vcm) \end{matrix}\right.$$

If we add the generic limit obtained in 5 we will have six limits, three negative and three positive.

$$\left.\begin{matrix} Vdmin \\ -k \cdot Vcm \\ -k(Vdd - Vcm) \end{matrix}\right\} \leq Vd \leq \left\{\begin{matrix} Vdmax \\ k \cdot Vcm \\ k(Vdd - Vcm) \end{matrix}\right.$$

We can graphically show those limits. As there are three negative limits, we can show the one that is the more limiting for each common voltage value. The same can be done for the three positive limits. Joining all this information together we obtain the following figure.



As a side note, this figure has been drawn using the slab command *plot1n* and the curve calculations have been performed in a Python script.

8	Obtain the k value and obtain the Vd limits for our opamp circuit. See if the limits agree with the above figure.

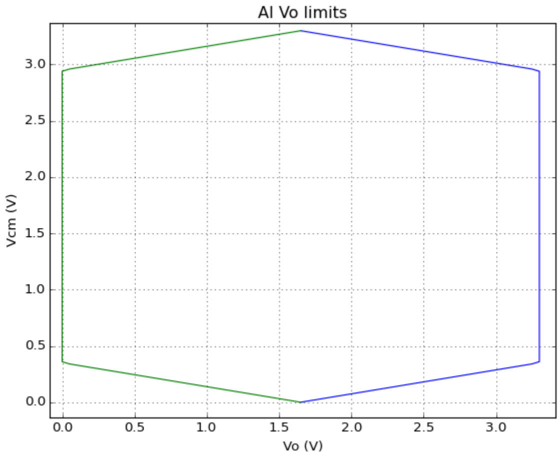
We remember that all three opamp are limited between zero and Vdd. That's two limits for each opamp or six limits in total. Just equal to the number of straight lines in the above figure. If you wish you can associate each line in the graph to one limit of one of the opamps.

For common mode voltages too close to zero or Vdd (3.3V in the figure), the first stage is limiting the Vd voltage and the Vd limit depends on the Vcm value. If Vcm is far enough from zero or Vdd, then there is a constant limit in Vd defined by the result obtained in 5.

AI manufacturers usually provide the above figure changing the axes and providing the output voltage Vo instead of the differential voltage Vd:

$$V_O = V_r + A_d V_d$$

In our case we obtain the following figure:



You can compare this figure to the red figure below for the Instrumentation Amplifier AD8422 provided by Analog Devices; the one associated to V_{REF} = 2.5V. The curve is not exactly equal as the gains are not the same, and the opams used are also different, but it also features six straight lines.

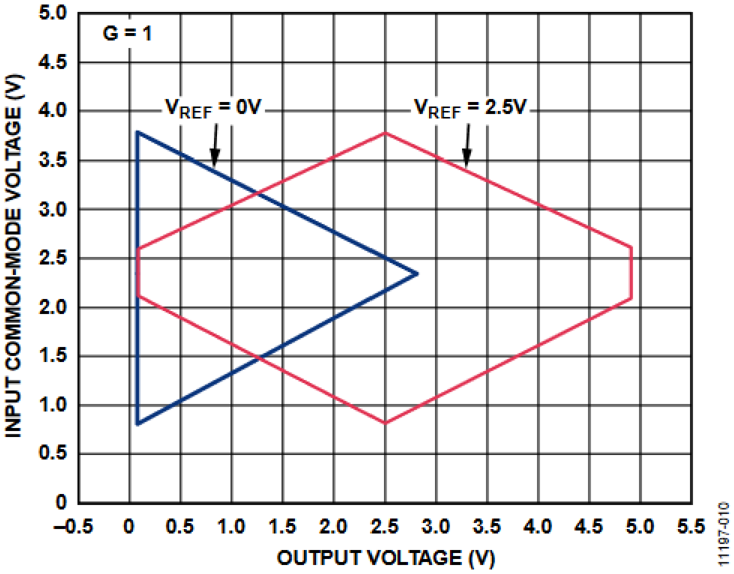


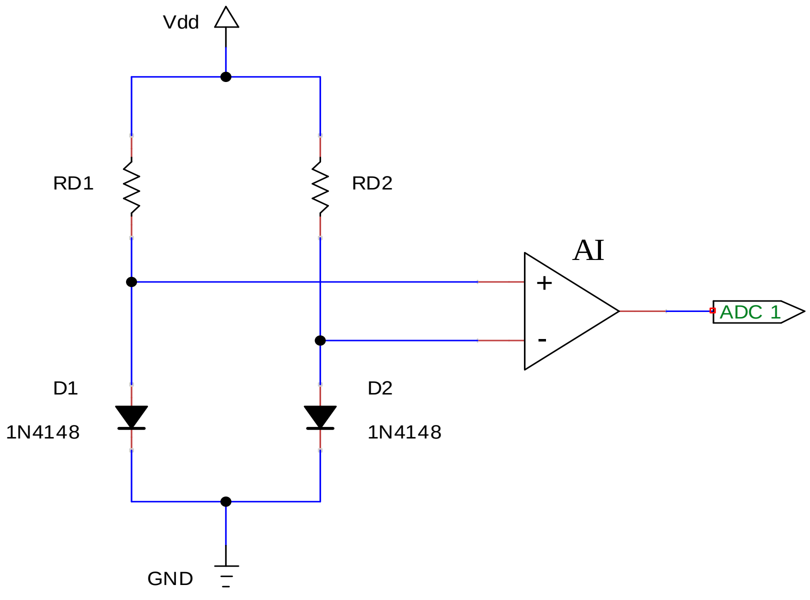
Figure 11. Input Common-Mode Voltage vs. Output Voltage (G = 1), Single-Supply, V_S = 5 V

Commercial AI devices, like the AD8422, usually provide the option for a total unity gain. As the first stage gain Ad1 cannot be lower than 1, and as the second stage has a fixed gain Ad2, this fixed A2 gain is usually one. That means that all the AI gain is usually implemented in the first stage. Having most of the gain in the first stage is usually good also to reduce the overall noise of the system. This solution, however, reduces the usable common mode range.

AI usage example

In this last section we will make use of the AI to perform an example of AI usage. In particular, we will perform differential temperature measurements.
It is know that semiconductor diodes have temperature dependence on their characteristics. The details are out of the scope of this document. We just need to know that the voltage of a diode, biased at constant current, decreases as the temperature increases.

Using this information we have designed a differential temperature measurement system based on two diodes D1 and D2, two resistors RD1 and RD2 and the Instrumentation Amplifier (AI) we have just built.



If both diodes D1 and D2 are equal and both resistors RD1 and RD2 are equal, due to symmetry, the voltage at both inputs of the AI will be the same, so the differential voltage will be zero and the output, as read by ADC1, will be the reference voltage of the AI; vdd/2 in our case.

If we heat one of the diodes, D2, for instance, its voltage will drop. As it is connected to the negative terminal of the AI, we will obtain a positive differential voltage that will be amplified on the output read by ADC1.

For the diodes we will use two 1N4148 devices and for RD1 and RD2 we will use two 5,6 kΩ resistors. The exact value is not important as long as the circuit is symmetric.

In order to increase the gain of the AI we will substitute the RG resistor by a 2,2 kΩ resistor.

9	Calculate the new Ad total gain for the AI after the RG change.

Before starting the measurements, we will set the number of readings for each ADC measurement to 100 to reduce the random noise.

```
>>> slab.setDCreadings(100)
```

Now we will use the ***dcLive*** command to show the ADC 1 readings in real time.

```
>>> slab.dcLive(1)
```

You should see the ADC 1 reading updated about five times each second. The readings should be around the reference Vdd/2 value.
Now touch diode D2 with two fingers. Remember that D2 is the diode connected to the negative input of the AI. As you transfer heat to the diode and rise its temperature you should see that the readings rise also.

If you remove the figures, the temperature will drop towards ambient temperature and the readings will return to Vdd/2.

You know that the readings are dependent on temperature as the times needed to stabilize the readings are long. If it was an artifact of current conduction trough your skin, the response time would be fast.

As the ***dcLive*** command performs an infinite loop of readings, you need to stop it by a keyboard interrupt. In particular, using CTRL+C.

You can represent your measurements graphically by indicating the ***dcLive*** command to store the captured data.

```
>>> data = slab.dcLive(1,returnData=True)
```

After you heat the diode, release it, and the temperature returns to ambient value, you can hit CTRL+C and all the captured information will be in the **data** variable.
To represent the information of the data variable just use the ***plot11*** command.

```
>>> slab.plot11([],data,"","Sequence number","Vo (V)")
```

We have used an empty list [ ] for the X axis as in this case we only have a sequence of values. The ***dcLive*** command, by default waits 0.2 second between two consecutive measurements, but the measurement time is not included in the count so we don't have a precise time axis to show.

	10	Perform the requested measurements. See how the temperature rise and fall time constants are not the same.

Last comments

In this document we have built and measured an Instrumentation Amplifier (AI). This circuit is the base of a lot of measurement instruments; hence its name. The AI depends on having a good matching of resistors. There are a lot of available integrated circuits that contain one or several AIs, with its three opamps and the connecting resistors, with all resistors precisely matched. Usually only the RG resistor is external to the device in order to set the total gain of the AI.

There are other AI topologies different from the three opamp topology we have used, but they are less common.

In the last section we have built a simple temperature difference measurement device. As this example, there are a lot of other examples that use an AI to extract information from a sensing device. In fact, a lot of instruments are built around an AI, a precision voltage reference and an analog to digital converter.

References

SLab Python References

Those are the reference documents for the SLab Python modules. They describe the commands that can be carried out after importing each module.
They should be available in the **SLab/Doc** folder.

TinyCad

Circuit images on this document have been drawn using the free software TinyCad
https://sourceforge.net/projects/tinycad/

SciPy

All the functions plots have been generated using the Matplotlib SciPy package.
https://www.scipy.org/