Linear Opamp 06 : Differential Amplifier

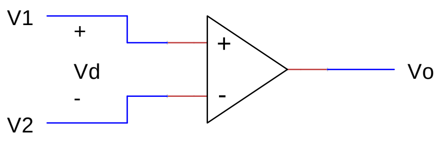
This project describes the opamp based differential amplifier. This block will be the basis of the instrumentation amplifier that follows. It will also be used to demonstrate the power of differential signaling.

BOM	1x Dual Opamp MCP6002 Resistors: 2x 1 kΩ, 2x 2.2 kΩ, 2x 33 kΩ Additional resistors: Either 4x 4.7 kΩ or 2x 4.7 kΩ and 2x 3.3 kΩ Extra components are needed for the last proposed lab work.

Index

Differential and Common Mode Gains

We have seen that we can use an opamp to provide positive gain, negative gain or to add several signals together. Now we will use it to build a circuit that amplifies the difference between two signals.



We have a circuit, like in the above image, with two inputs V_1 and V_2. And we want the output to be proportional to the difference between the inputs.

$$V_O = A_d(V_1 - V_2) = A_d V_d$$

So we have defined the constant as $\mathbf{A_d}$ or **Differential Gain** and the voltage difference as **Vd** or **Differential Voltage**.

We want the circuit to depend only on the voltage difference, not on the voltage that is common to both inputs. That way we define a **Common Mode Voltage**, or V_{CM}, as the voltage that is common to both signals:

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

In the same way that we can obtain V_d and V_{cm} from V_1 and V_2, you can obtain V_1 and V_2 given V_d and V_{cm}. So knowing V_1 and V_2 gives the same information that knowing V_d and V_{cm}.

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

We want the output to not depend on V_{cm} and only depend on V_d. We can define the undesired output dependence on V_{cm} as the **Common Mode Gain** $\mathbf{A_{cm}}$. That way, we have the following expression that models the dependence on V_d and V_{cm}.

$$V_O = A_d(V_1 - V_2) + A_{cm}\frac{V_1 + V_2}{2} = A_d V_d + A_{cm} V_{cm}$$

Ideally we want A_{cm} to be as low as possible. Unfortunately it is impossible for real circuits to have no dependence at all to the common mode voltage, so we can quantify how big is the differential gain compared to the common mode gain comparing both gains. That yields the **Commom Mode Rejection Ratio** figure, usually given in dB.

$$CMRR = dB\left(\frac{A_d}{A_{cm}}\right) = 20 \cdot log_{10}\left(\frac{A_d}{A_{cm}}\right)$$

The higher the CMMR the better the differential amplifier is.

If the circuit is linear, the output shall be a linear function of all its inputs, if we only have two inputs V_1 and V_2 the following expression shall hold:

$$V_O = f_{Linear}\,(V_1, V_2)$$

If the circuit has no time dependences, the only possible linear function is the linear combination so it also shall hold:

$$V_O = K_1 V_1 + K_2 V_2$$

We can substitute V_1 and V_2 with the expression that relates them to V_{cm} and V_d:

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

That yields:

$$V_O = (K_1 + K_2)V_{cm} + \frac{K_1 - K_2}{2} V_d$$

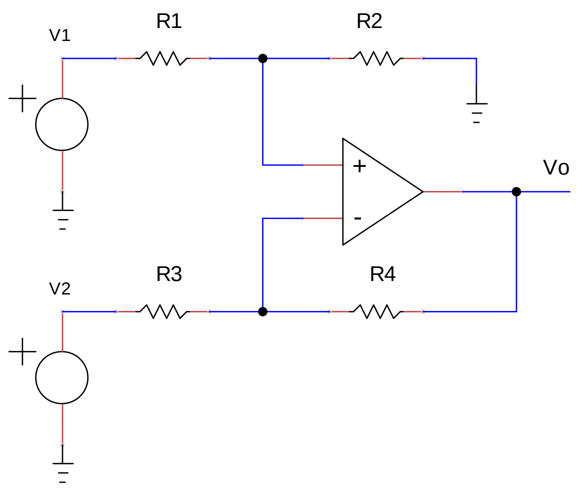
So differential and common mode gains can be calculated from K1 and K2:

$$A_d = \frac{K_1 - K_2}{2} \qquad A_{cm} = K_1 + K_2$$

That provides us a way to analyze a differential amplifier, we obtain K1 and K2 and, from that information we can get A_d, A_{cm} and the CMRR.

Differential Amplifier Basics

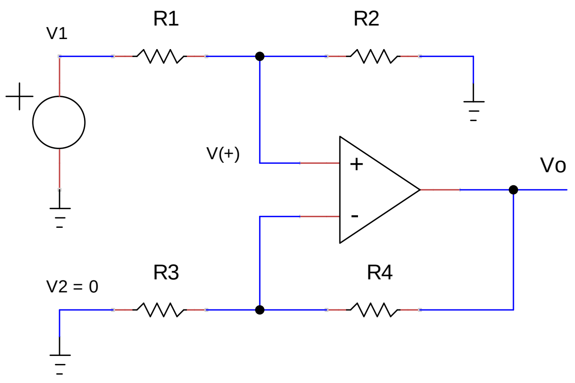
The following figure shows the basic four resistor differential amplifier.



This circuit is easy to analyze using superposition:

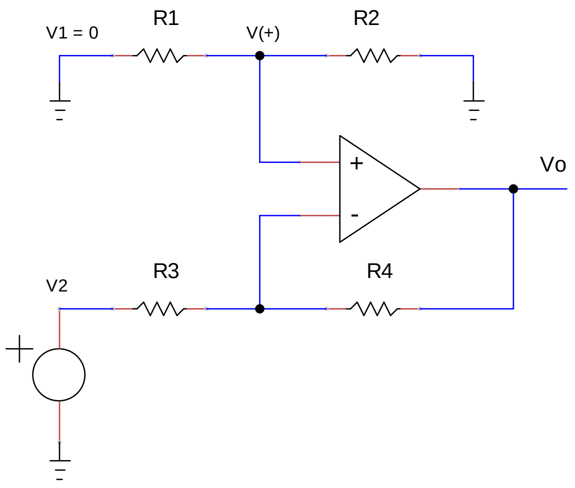
$$V_O(V_1, V_2) = V_O(V_1, 0) + V_O(0, V_2)$$

The first superposition case sets V_1 to zero, so the circuit to solve is:



We could solve this circuit using the **virtual shortcircuit** method, but this case is easier than that. You should see that the circuit resembles like a non inverting amplifier due to the R3 and R4 configuration. The input of this amplifier, instead of being directly introduced on the $V_{(+)}$ node, is obtained by a resistor divider composed of R1 and R2.

The second superposition case sets V_2 to zero, so the circuit is:



This circuit is easier to solve. As current entering the $V_{(+)}$ operational input is zero, the voltage at $V_{(+)}$ shall be zero and we end with the basic inverting amplifier.

Once you have solved both superposition cases you can obtain an expression for $V_O(V_1,V_2)$.

	1	Solve both superposition cases. Obtain the expression of $V_O(V_1,V_2)$ as function of the resistor values.

You should obtain an expression like:

$$V_O(V_1, V_2) = K_1 V_1 + K_2 V_2$$

Where K_1 and K_2 are function of the resistor values and K_1 should be positive and K_2 should be negative.

We can calculate the two A_d and A_{cm} gains from K_1 and K_2. We want to have a differential amplifier, with zero common mode gain, so we want A_{cm} to be zero.

	2	Obtain A_d and A_{cm} from the resistor values. Obtain relationship that the resistors must verify so that A_{cm} is zero. Rewrite A_d expression if the resistors verify the above condition.

You should have obtained that, in order to have zero common mode gain, resistors shall verify:

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

You should also have obtained a simple expression for A_d:

$$A_d = \frac{R2}{R1}$$

We could use the following commercial values to obtain an A_d gain of 2.2 :

R1 = R3 = 1 kΩ R2 = R4 = 2.2 kΩ

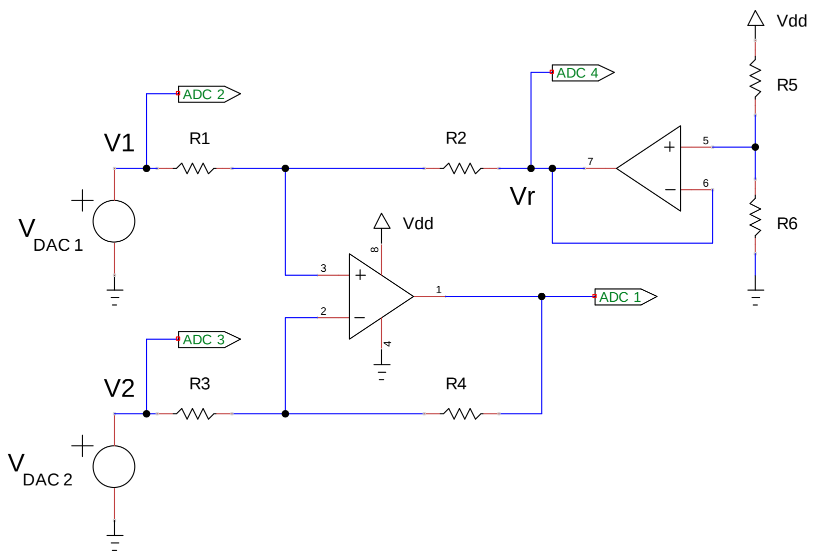
As we remember, we cannot generate negative voltages in the SLab setup because the power supply is unipolar. So we need to use a reference voltage instead of ground.

The following circuit is a possible solution. The reference voltage Vr is generated with a voltage divider from the supply Vdd voltage using two equal resistors R5 and R6. A voltage follower guarantees that the Vr reference has a near zero output resistance.

The circuit requires two opamps, so it uses the two devices of a MCP6002 package.

The list of resistor values is:

R1 = R3 = 1 kΩ R2 = R4 = 2.2 kΩ R5 = R6 = 33 kΩ



Note about PSRR

Generating the reference voltage using a divider like in the above circuit makes an easy path for noise on the supply to reach the amplifier output degrading the Power Supply Rejection Ratio (PSRR) of the amplifier. We can reduce the noise adding a capacitor in parallel with R6, but the best solution will be to use a proper voltage reference that is independent on the supply voltage.

First Measurements

We can start the measurements by obtaining the reference Vr voltage and storing it on a vr variable:

```
>>> vr = slab.readVoltage(4)
>>> vr
```

Now we can provide a differential 100 Hz sine signal with 0.5 V amplitude.

```
>>> slab.setVoltage(2,vr)
>>> slab.waveSine(vr-0.5,vr+0.5,100)
>>> slab.setWaveFrequency(100)
```

Then we can perform the measurement of the output respect to the Vd signal.

```
>>> slab.tranStore(400,4)
>>> slab.wavePlot()
```

We should see that $V_{DAC\ 2}$, read at ADC 3 and Vr, read at ADC 4 are equal and about half Vdd. We should also see that the output voltage read at ADC 1 is greater and in phase with the input signal read at ADC 2.

You can calculate the gain from the amplitudes, for instance:

```
>>> t,vo,v1,v2,vr = slab.waveResponse()
>>> Ad = slab.peak2peak(vo)/slab.peak2peak(v1)
>>> Ad
```

🛠	**3**	Mount the circuit and check the Vr reference voltage. Perform the requested measurements and obtain the Ad gain. Do you obtain the expected value?

Now we want to obtain the CMRR, so we need to response to a common mode signal:

Disconnect the DAC2 output from the V_2 input of the circuit and connect this V_2 input to the same node as the V_1 input.

You can now see the Vo voltage due to common mode gain:

```
>>> slab.wavePlot()
```

You will probably need to zoom in the vertical axis as the output signal should be small. If the resistors are well matched you will see the quantization levels of the ADC converter. If the resistors are very well matched you won't see the sine wave on the output signal. In that case, CMRR cannot be calculated.

If there is a sine wave on Vo, you can calculate the Acm gain in a similar way as the Ad gain:

```
>>> t,vo,v1,v2,vr = slab.waveResponse()
>>> Acm = slab.peak2peak(vo)/slab.peak2peak(v1)
>>> Acm
```

Now you can obtain the CMRR. The **dB** command is part of the AC module so you will need to import it.

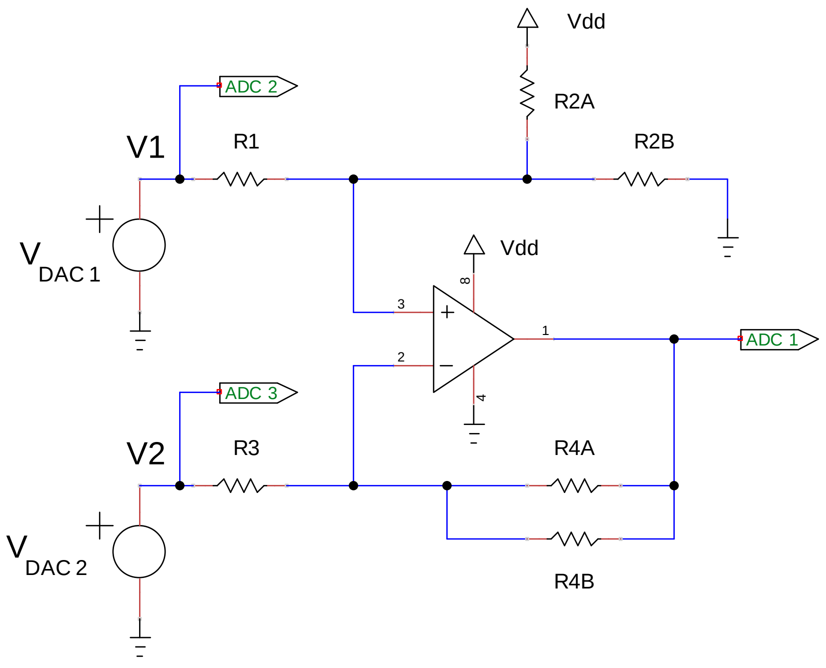
```
>>> import slab_ac as ac
>>> cmrr = ac.dB(Ad/Acm)
>>> cmrr
```

CMRR is probably between 40 dB and 60 dB. If it is much lower than 40 dB, your resistors are badly matched (they are not equal when they should be). If it is much higher than 60 dB, then there is a very good matching.

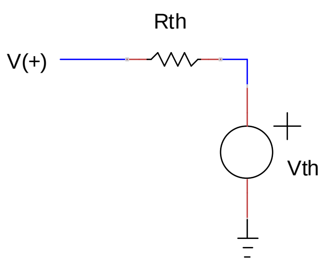
🛠	**4**	Obtain the CMRR of the amplifier.

Reducing the number of components

Using an opamp to generate the reference voltage adds complexity to the system. The following circuit simplifies the design by using the Thévenin's theorem.



The circuit defined by resistors R2A, R2B and the Vdd supply are equivalent to a Thévenin circuit:



If R2A and R2B are equal, then we can obtain:

$$Vth = \frac{Vdd}{2} \qquad Rth = R2A||R2B = \frac{R2A \cdot R2B}{R2A + R2B}$$

We use Vth as the Vr reference voltage and Rth as the previous R2 resistor.

The main problem is conserving the matching to obtain a good CMRR. In order to make R2 as similar as possible as R4, resistor R4 is also obtained by two resistors R4A and R4B equal to the R2A and R2B ones.

Proposed resistor values for the circuit are:

R1 = R3 = 1 kΩ   R2A = R2B = R4A=R4B = 4.7 kΩ

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

R1 = R3 = 1 kΩ   R2A = R4A = 4.7 kΩ   R2B=R4B = 3.3 kΩ

As resistors are different from the previous circuit, the theoretical A_d gain is slightly different from the one on the previous circuit. You have the tools so you can calculate it.

We don't have vr in any node, we can calculate it from the supply voltage or we can use the previously used vr value. The amplifier has enough low gain that we don't need to exactly match the reference voltage value. So, you probably don't need to get the Vr value. If all four resistors R2A, R2B, R4A and R4B are equal:

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

If you have used different resistor values for R2A and R2B, Vr will be different than before. Tou can work out the reference voltage from the resistor values:

```
>>> vr = slab.vdd*R2B/(R2A+R2B)
```

Now we can see the differential response. As we don't have the Vr node, we won't use ADC4 so we set the transient storage to only 3 channels. If Vr has not changed we will use the same waveform of the previous analysis, so it won't be input again.

```
>>> slab.setTransientStorage(400,3)
>>> slab.wavePlot()
```

If Vr has changed we would have to recalculate the input signals:

```
>>> slab.setVoltage(2,vr)
>>> slab.waveSine(vr-0.5,vr+0.5,100)
>>> slab.setTransientStorage(400,3)
>>> slab.wavePlot()
```

Now we can calculate, again, the differential gain Ad:

```
>>> t,vo,v1,v2 = slab.waveResponse()
>>> Ad = slab.peak2peak(vo)/slab.peak2peak(v1)
>>> Ad
```

We can also follow the same previous method to obtain the A_{cm} and the CMRR.

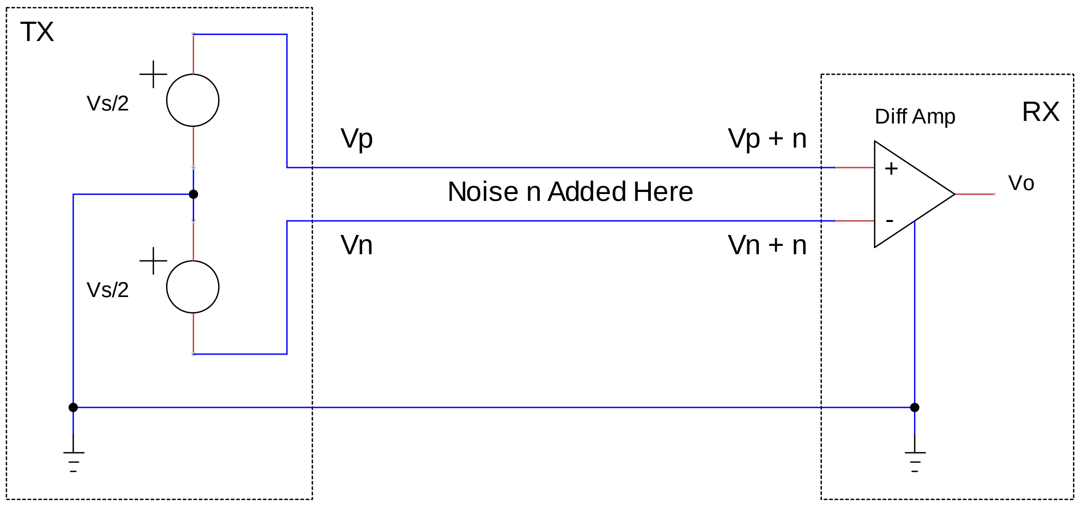
🛠	5	Mount the circuit and measure A_d and A_{cm} and obtain the CMRR value.

Depending on the resistors used, the CMRR could be better or worse than in the previous circuit but it should not be very far from the other circuit value.

Differential Signaling

In this section we will deal with one of the uses of differential amplifiers.
When we want to send a signal over a long distance, it is usual that noise is added to the signal during its path from the emitter to the receiver. If the noise is in the same frequency band of the signal, it is not easy to filter out the noise to recover the original signal.

A very powerful approach in this case is the use of differential balanced signaling. This method gives us the power of symmetry. The following circuit shows a basic description of the transmission system.



We have a transmitter (TX) that wants to send a signal Vs(t). As we see, it is connected to a reference ground terminal. In order to send a signal, it generates two halves of the signals and sends them, with opposite phases respect to the ground node, to two dedicated lines P and N.

The two halved signals Vp and Vn come out clean from the transmitter, but, during its way to the receiver (RX), noise is added to the signals.

Now, drum roll, behold the power of symmetry!

If the construction of the P and N lines is symmetric respect to ground, the very same noise should be added to both signals. To preserve symmetry, care should be taken in the wiring of the P and N lines. Usually they are twisted together forming a twisted pair.
Symmetry shall be preserved also in the receiver, so both receiver inputs shall have the same characteristics down to input impedance and bandwidth.

Thanks to symmetry, there is no way the noise can be different on both lines as both lines are seen as exactly equal from any external interfering source.

To reconstruct the original signal we only need to obtain the difference of the two P and N lines:

$$V_O = (V_P + n) - (V_N + n) = \left(\frac{V_S}{2} + n\right) - \left(-\frac{V_S}{2} + n\right) = V_S$$

It could seem a simple trick, but it works wonderfully because it is so simple. A lot of signaling systems, both analog and digital, use the power of differential signaling to fight noise. For instance:

- Balanced audio for microphone and line audio signals
- RS-422 digital serial communication
- Ethernet data physical layer

We will try that in our SLab system. We could try to mimic a complete system with the transmitter, receiver and the noise source but, to simplify the related circuit, we will use a synthetic generation of the signals P and N that arrive to the receiver. Once those signals are generated, we will feed them to DAC1 and DAC2. That way we can use the circuit analyzed on the previous section to reconstruct the output signal.

First, we generate the signal we want to send. It will be 4 cycles (40 ms) of a 0.1V amplitude 100Hz sinewave. We will need the numpy package to generate the signal so an import is in order. The following code generates and shows that signal:

```
>>> import numpy as np
>>> t = np.arange(0,0.04,0.0001)
>>> w = 100*2*np.pi
>>> s = 0.1*np.sin(w*t)
>>> slab.plot11(t,s)
```

Now we can generate the P and N signals at the output of the transmitter:

```
>>> sp = 1.5+s/2
>>> sn = 1.5-s/2
>>> slab.plot1n(t,[sp,sn])
```

We also generate the noise using the numpy package. Noise is generated from an uniform distribution between 0 and 1. Using the following code we are sure that the noise is bound to the -1V to 1V interval.

```
>>> n = 2*np.random.random(size=len(s))-1
>>> slab.plot11(t,n)
```

A noise defined by a normal distribution could also be used, but we could have problems with dynamic range. If the amplifier input signal goes out of its common mode input range we lose the benefits of using differential signaling. In practice that is not a problem because RX amplifiers are designed with input dynamic range high enough to prevent this problem. But in our case we are working using a simple receiver and we are adding a lot of noise.

Now we add the noise to both differential lines. Thanks to the power of symmetry, the same noise is added to both channels.

```
>>> sp_n=sp+n
>>> sn_n=sn+n
>>> slab.plot1n(t,[sp_n,sn_n])
```

See that the signal has disappeared inside the noise. The noise amplitude is much greater than the signal amplitude so the signal to recover is not apparent on the noisy P and N signals that reach the receiver.

Here ends the synthetic generation of the P and N signals. Now we want to feed them to the amplifier using the two DACs. Signal P will be output on DAC1 and sent to V1 of the differential amplifier and signal N will be output on DAC2 and sent to V2.
Wave frequency is set to 25 Hz because it relates to the full loaded wavetable that includes 4 cycles of the 100 Hz signal.

```
>>> slab.loadWavetable(sp_n)
>>> slab.loadWavetable(sn_n,second=True)
>>> slab.setWaveFrequency(25)
>>> slab.wavePlot(dual=True)
```

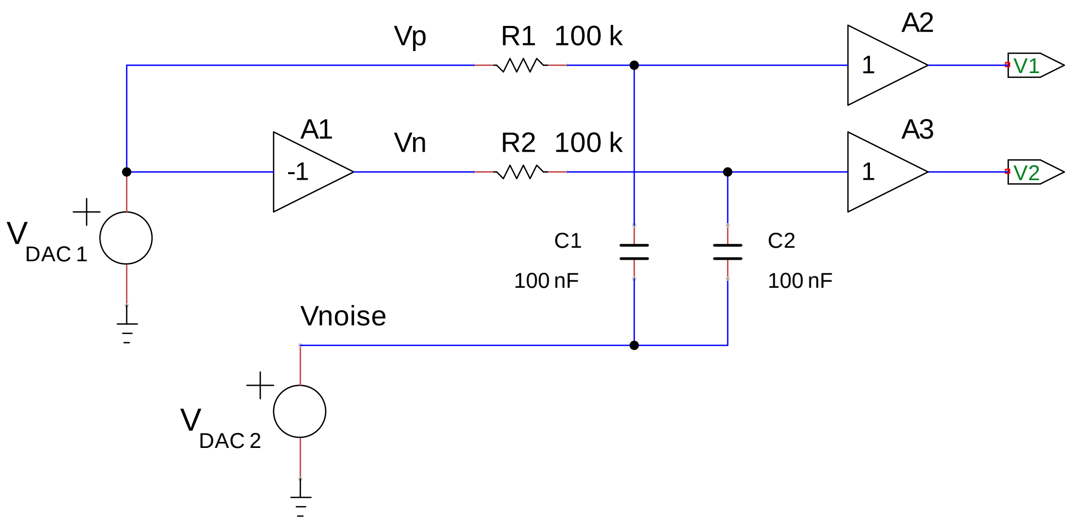
You should see a clean reconstruction of the input signalon the Vo output of the amplifier that is read on ADC1. The signal should be amplified due to the A_d gain, but in every other way it should have the same shape as the original signal.

🛠	**6**	Generate the noisy synthetic P and N signals. Feed them to the DACs and check that the differential amplifier can reconstruct the original signals.

Only for the Brave

In the previous section we have used a synthetic generation of the differential amplifier inputs V_1 and V_2. Perhaps you are not convinced that synthetic signals represent real world cases. Perhaps you want to see the real deal. Perhaps the previous circuits were too easy to build. If this is the case you have come to the right place in this section.

The following schematic shows a circuit that uses real differential signal generation and uses real noise injection. It is a close to real world implementation of the synthetic V_1 and V_2 generation on the previous section.



Signal $V_S/2$ that drive the V_P line is generated at DAC 1. As our signal V_S has 0.1 V amplitude, we have to generate a 50 mV amplitude signal around the Vdd/2 reference voltage Vr.
Inverting amplifier A1, with a -1 gain, inverts V_P around Vr to generate V_N.
At this point we have developed the generation of V_P and V_N on the transmitter.

Noise will be generated on DAC2. We will generate a voltage between 1V and 2V using an uniform random distribution and we will load it as a secondary wavetable on DAC2 with the same length, although it is not a requirement, as the signal generated on DAC1.

The injection of noise on the signal lines will be carried out with the R1, C1 and R2, C2 networks. If you take ADC readings on the two sides of the resistors you will see that noise is added on the right hand side of the resistors.
The networks injection circuits feature a first order high pass behavior with the pole at 16 Hz. That's enough for the kind of noise we are injecting.

Differential amplifiers are very sensitive to the output resistance of the circuits connected to their inputs. For our differential amplifier to work as expected, output resistance of the driving circuit must be close to zero. So we need two gain 1 amplifiers A2 and A3 to drive our differential amplifier inputs V_1 and V_2.

Building this circuit is not an easy task. A part from the differential amplifier, not shown in the schematic, that uses one opamp and six resistors, we have three more amplifiers A1, A2 and A3. A1 is an inverting amplifier with reference at Vdd/2 so it is composed by one opamp and four resistors.
A2 and A3 are unity gain amplifiers so they are just voltage followers, each one using one opamp.
The full circuit uses four opamps (two MCP6002 dual opamp integrated circuits), two 100nF capacitors and a bunch of resistors.
You will probably fill the available breadboard space with this circuit.

After building the circuit you can check that signals V_1 and V_2 on the receiver are noisy and that the output V_O of the differential amplifier really rejects the noise and the original signal, generated on DAC1 is recovered.

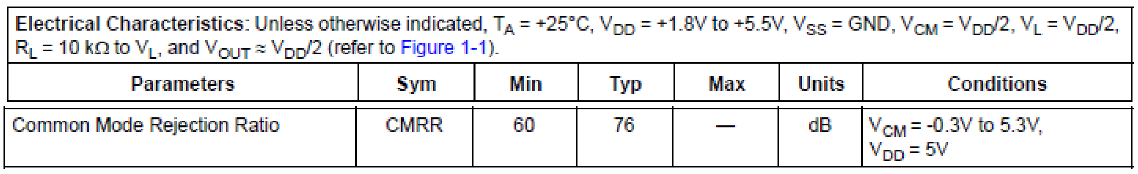
You are on your own for the Python code. At this point you have examples of all the commands you will need.

🛠	7	Build the proposed circuit. Check that noise is injected on V_1 and V_2 and that you can recover the original signal.

Last comments

The differential amplifier is one of the fundamental opamp linear circuits. It is also the foundation for the Instrumentation Amplifier that we will see shortly.
In this document we have also shown the importance of component matching to obtain good performance circuits. In this amplifier the CMRR is very sensitive to resistor tolerances.

The opamps we are using, in the MCP6002 package, are, in fact, a special kind of differential amplifiers with very high gain. The CMRR is one of the parameters indicated on the component datasheet:



Electrical Characteristics: Unless otherwise indicated, T_A = +25°C, V_{DD} = +1.8V to +5.5V, V_{SS} = GND, V_{CM} = V_{DD}/2, V_L = V_{DD}/2, R_L = 10 kΩ to V_L, and $V_{OUT} \approx V_{DD}/2$ (refer to Figure 1-1).

Parameters	Sym	Min	Typ	Max	Units	Conditions
Common Mode Rejection Ratio	CMRR	60	76	—	dB	V_{CM} = -0.3V to 5.3V, V_{DD} = 5V

Observe how the component has a typical CMRR of 76 dB and a minimum value of 60 dB. You can compare those values with the ones you have obtained. As we are using one MCP6002 opamp to build the differential amplifier, the CMRR of the opamp limits the CMRR we can obtain on our circuit. The relevant calculations are outside of the scope of this document.

In the document we have also used the differential amplifier to expose the benefits of the using differential signaling. It is a powerful way to fight noise when transmitting a signal in long wires that can receive induced noise and is used in a lot of communication standards.

Finally, in the last section, we have proposed a complex circuit project where a full transmission system is build from the transmitter to the receiver and including noise injection. If you are able to complete the proposed task, then you are quite good on the SLab system workings.

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/