	Linear Opamp 04 : Inverting Amplifier

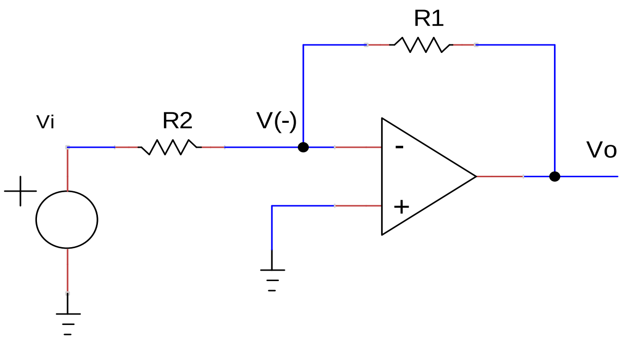
This project deals with another common opamp topology: the inverting amplifier.
Using this circuit more formal feedback calculations are made.

BOM	1x Dual Opamp MCP6002 Resistors: 100Ω, 1kΩ, 2.2kΩ and 2x 33kΩ Capacitors: 100nF and 10µF (Electrolytic)

Index

The inverting circuit topology

The following circuit is an inverting amplifier. It generates an output voltage that is inverted compared to the input. That is, features a 180º phase difference. Depending on the resistors the output amplitude can be higher, equal or lower than the input. This makes a difference respect to the non inverting circuit that cannot attenuate the input signal.



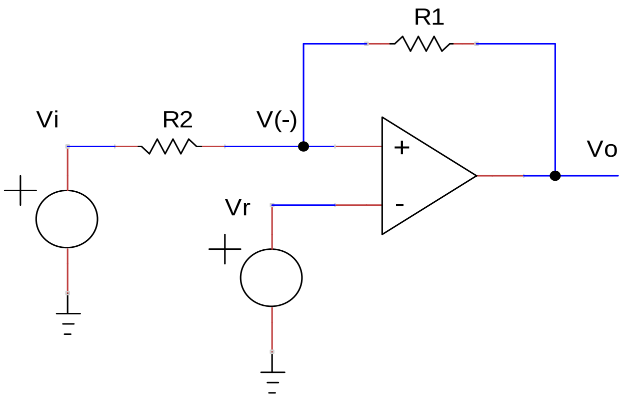
As in the non inverting amplifier, we could also use the first order model of the opamp to solve the circuit but we will start with the easy **virtual short** circuit model path.

	1	Obtain the voltage gain G1 = Vo/Vi as function of R1 and R2 for the circuit using the virtual short circuit model.

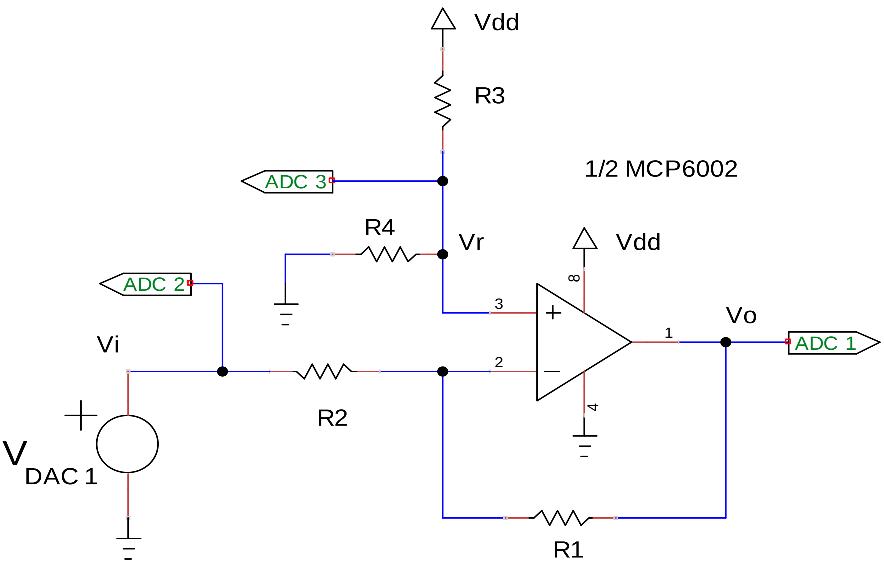
You should note that the gain G1 is negative. That is, inverts the phase. You should also note that the absolute value of the gain can be greater that 1 (Amplifying) or less than 1 (Attenuating) depending on the value of the resistors. This is different from the non inverting amplifier that could not attenuate the signal.

As the gain is negative, a positive input value will generate a negative output value. If the opamp has is fed by a unipolar supply, between Vdd and GND, the output cannot be negative so this amplifier is useless as is.

We can add a reference voltage source for the $V_{(+)}$ terminal. That will enable work with voltage differences between the input Vi and the reference voltage Vr.



Using a voltage source to generate Vr is overkill. Specially if you use an opamp, like the MCP6002 that has near zero input current. You can substitute this voltage source by a voltage divider.



	2	Obtain Vr as function of R3, R4. Obtain Vo as function of Vi, Vr, R1 and R2.

The expression can be written as:

$$V_O = V_r + G1 \cdot (V_i - V_r)$$

Where G1 should be a negative value.

If the opamp has full rail output, the reference voltage usually is set at half Vdd. So you only need to make R3 and R4 equal. We will use the following set of values:

R1 = 2,2 kΩ   R2 = 1 kΩ   R3 = R4 = 33 kΩ

	3	Calculate Vr and G1 for this set of components.

Measuring the circuit

Now we can build and measure this circuit. Remember to import and connect as needed.
First, we can measure the reference voltage Vr.

```
>>> slab.readVoltage(3)
```

Now we can obtain the DC curve:

```
>>> dc.curveVV(0,3.2,0.1)
```

We can plot the data using the voltage reference as the zero point for input and output values. Remember that the dcSweep command returns a 5 element list with the chosen DAC at position 0 and ADC values at positions 1 to 4.

```
>>> data = slab.dcSweep(1,0,3.2,0.1)
>>> vi = data[0] – data[3]
>>> vo = data[1] – data[3]
>>> slab.plot11(vi,vo,”Inverting Amplifier”,”Vi(V)”,”Vo(V)”)
```

We can also plot the response of the circuit against a 100 Hz sine wave:

```
>>> slab.waveSine(1.2,2.1,100)
>>> slab.setWaveFrequency(100)
>>> slab.tranStore(500,3)
>>> slab.wavePlot()
```

You can see that the reference voltage at ADC 3 is constant whereas the output at ADC 1 has gain respect to the input at ADC 2. Both signals have the same mean that is the reference voltage.

If we shift the input signal respect to the reference, as this shift is affected by the negative gain, the output will shift in the opposite direction.

```
>>> slab.waveSine(1.3,2.2,100)
>>> slab.wavePlot()
```

We would like to measure the bandwidth of the circuit, but as in the non inverting amplifier with low gain, it is too high to be measured with the SLab system.

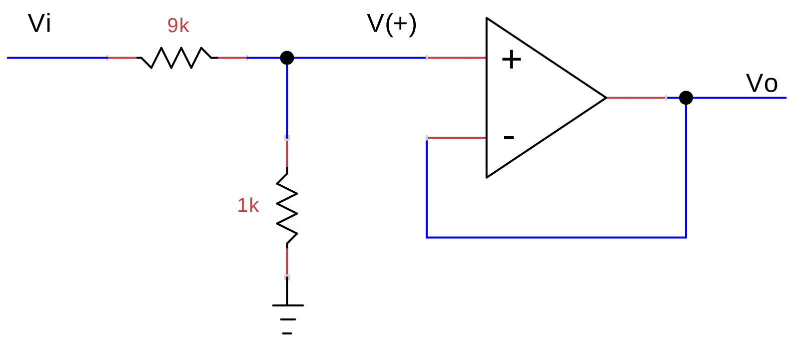
🛠	**4**	Perform the requested measurements. Are the reference voltage and the gain as expected? Check the shift of the signals when they are not centered in Vr.

Losing bandwidth

Turns out, that this circuit does not preserve the GBW product. At least it does not preserve the product of the signal gain and the bandwidth.

As we remember from previous projects, both the follower and the non inverting amplifier feature the same GBW product as the open loop amplifier. Comparing those circuits with the inverting amplifier you should note that this circuit has one difference: The input is not connected to any opamp terminal. And here relies the problem. The GBW product is conserved in the opamp component between its inputs and outputs, not at the whole circuit.

Take, for instance, this circuit:



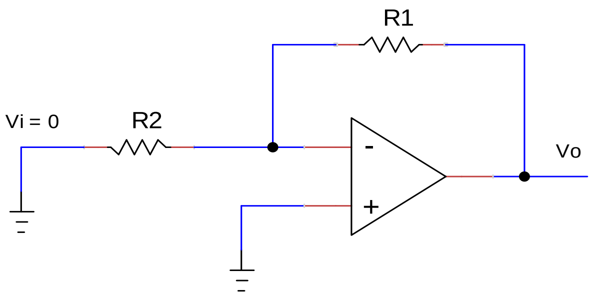
This is a voltage divider followed with a follower (pun intended). As resistors are 9kΩ and 1kΩ, the gain for the circuit is 1/10. The GBW for the follower, between $V_{(+)}$ and Vo is the same that the one of the opamp in open loop. But the GBW for the whole circuit, between Vi and Vo is 10 times lower.

Remember this mantra:
"The GBW product conservation is a property of the operational amplifier, not of the whole circuit that uses it".

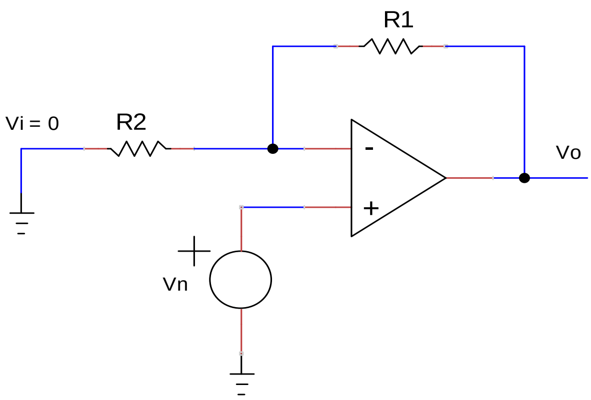
Let's return to the original inverting circuit. You could be tempted to check if the GBW is conserved by calculating the amplifier gain between V(-) and Vo. But, thanks to the virtual short circuit, the gain between those points is infinite.
That's because the signal at node V(-) is not voltage but current: The one that flows through R2.

Calculating noise gain

Fortunately there is a trick to calculate the gain at opamp pins. First we make zero all independents supplies, like Vi. As the reference acts as voltage source, we can also ground the $V_{(+)}$ terminal. We get:



Then we put an input voltage source Vn in series with the V(+) terminal.



Now, calculate the gain for this circuit using the virtual short circuit model:

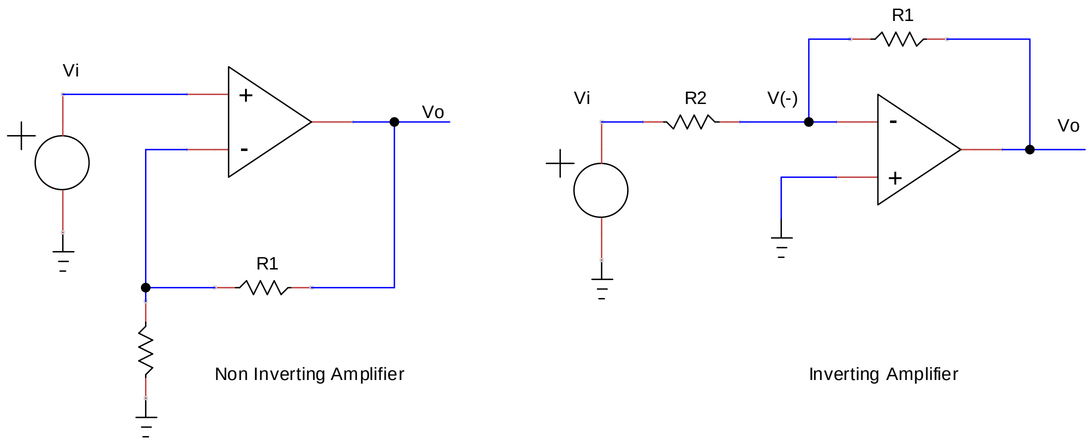
$$G_N = \frac{Vo}{Vi}$$

This G_N value, or noise gain, is the one we need to calculate the circuit bandwidth.

$$BW = \frac{GBW_{Opamp}}{G_N}$$

	5	Calculate the G_N gain value for the circuit as function of R1 and R2. Calculate G_N for the resistor values of the circuit measured in 4 . Calculate the circuit BW using G_N.

You can see that the G_N value equation, as function of R1 and R2, reminds us of the non inverting amplifier. In fact, it is the very same equation.
If you calculate the noise gain G_N in those two circuits you will get the same noise gain calculation final circuit and so, the same noise gain.



Perhaps you are perplexed by the fact that we are talking about noise gain when no noise is involved in any of the calculations. The fact is that this noise gain that can be used to predict the circuit bandwidth can also be used in noise calculations. As it is the same number, we use only one name to describe it although noise calculations are out of the scope of this document.

At this point it seems that the noise gain is the way to go to calculate the circuit bandwidth, but we have not demonstrated those results.

By calculating the noise gain, we have prevented the use of circuit calculations in the "s" domain. Nothing prevents you to calculate the circuit using the single pole operational amplifier model. You would obtain the very same results if you do. Try yourself if not convinced.

Doing it in another way

Beware: This section contains dragons.
Here we do some calculations to demonstrate the previous results. If you are happy enough believing that the circuit bandwidth can be calculated from the noise gain G_N and don't want to do more calculations about that, you can skip this section and go to the "input resistance" section.
However, if you are not satisfied with believing but you also want to know why it works, keep reading.

Up to this point we have used the virtual short circuit model because it is the easiest way to calculate the circuit operation. This model, however, has some serious limitations:

- Cannot provide any insight about the circuit stability. This model depends on the assumption that the circuit is stable and its output is not saturated.
- Cannot provide any frequency response data. That's why we have used the GBW product conservation to estimate its bandwidth.

We will use the first order model directly on the circuit to analyze, but we will follow an indirect path to it. For all those calculations you can connect $V_{(+)}$ to GND. No need for Vr.

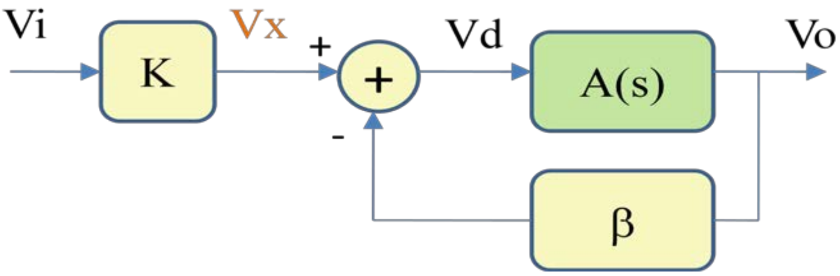
First, we will calculate the Vd value as function of Vi and Vo. These calculations don't consider the operational amplifier at all. You will end up with something like that:

$$Vd = K \cdot Vi - \beta \cdot Vo$$

	6	Obtain K and β on the inverting amplifier as function of R1 and R2. Calculate them for the chosen resistor values.

In our case K should be negative and β should be positive because Vd lowers when either Vi or Vo increases.

The obtained Vd expression lead us to a description o the system like the one in the following figure.



The equation data calculated in 6 relates to the yellow blocks in the figure. As we see Vd has two contributions, one from the input voltage Vi and one from the output voltage Vo. Don't worry about Vx for now, it is just an aid for what follows.
The operational amplifier relates Vd with Vo, so we insert here the A(s) first order operational amplifier model.
This model is useful because you can always obtain Vd as function of Vi and Vo and calculate the K and β values. In this case both are constants, but the method will be the same if K and β are arbitrary transfer functions in the "s" domain.

In order to solve the circuit we need to obtain its transfer function:

$$H(s) = \frac{Vo}{Vi}$$

So we need to take Vd out of the equations.
First we will deal with the right side blocks of the diagram we want to calculate $H_F(s)$ as:

$$H_F(s) = \frac{Vo}{Vx}$$

We call it $H_F(S)$ because it includes the feedback path β that relates the output and the input of the operational amplifier. As β defines the feedback path, we call its value as the feedback gain. Do not confuse the feedback gain β with the bipolar transistor β current gain. They are completely different concepts.

We have made similar calculations to the ones needed to obtain $H_F(s)$ before, but we will include them once more:

$$V_o = A(s)V_x - V_o\,\beta\,A(s) \qquad V_o\{1 + \beta\,A(s)\} = A(s)V_x$$

$$H_F(s) = \frac{V_o}{V_x} = \frac{A(s)}{1 + \beta\,A(s)}$$

This equation is the main equation for feedback systems and is the same regardless of the current values of A(s) or β.
In our particular case we know that A(s) models the opamp so we will use its first order equation:

$$A(s) = \frac{Ao}{1 + \frac{s}{p1}}$$

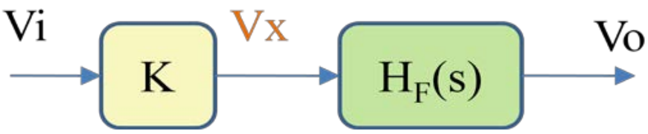
That gives:

$$H_F(s) = \frac{A(s)}{1 + \beta\,A(s)} = \frac{\frac{Ao}{1+\frac{s}{p1}}}{1 + \beta\,\frac{Ao}{1+\frac{s}{p1}}} = \frac{Ao}{1 + \frac{s}{p1} + \beta\,Ao}$$

As usual, we can assume Ao much greater than one.

$$H_F(s) \approx \frac{Ao}{\beta\ Ao + \frac{s}{p1}} = \frac{1}{\beta} \cdot \frac{1}{1 + \frac{s}{\beta\ Ao\ p1}}$$

Once we have calculated $H_F(s)$ we can simplify the system description:



Now, we can obtain the transfer function all the system:

$$H(s) = K \cdot H_F(s) = \frac{K}{\beta} \cdot \frac{1}{1 + \frac{s}{\beta\ Ao\ p1}}$$

The previous equation is an important equation to remember because is valid for any opamp circuit when we use its first order model. The equation indicates that the system transfer function, if K and β are constant values, is also first order with gain and bandwidth:

$$G = \frac{K}{\beta} \qquad BW = \beta\ Ao\ p1 = \beta \cdot GBW_{Opamp}$$

If you remember the previous way to do calculations we came up to the noise gain G_N. You should note how the noise gain G_N and feedback gain β relate to each other:

$$G_N = \frac{1}{\beta}$$

So we demonstrate that, when the first order operational model holds, the bandwidth is:

$$BW = \beta \cdot GBW_{Opamp} = \frac{GBW_{Opamp}}{G_N}$$

7

Take the K and β calculated in 6.

Calculate the circuit gain G and bandwidth BW.

Compare with the results obtained in 5.

Input resistance

The non inverting opamp circuit had the Vi input connected to the V(+) input of the opamp. As an ideal opamp has no input current, this amplifier didn't consume any current from the Vi source.

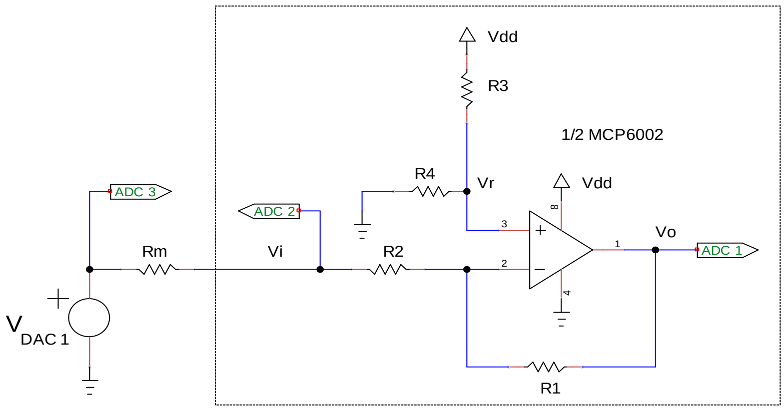
This is no longer the case with the inverting opamp. There will be some current entering resistor R1. From that, we can define an input resistance as the ratio of the input voltage to the input current in the circuit (entering R1).

$$Ri = \frac{Vi}{Ii}$$

This Ri input resistance is easy to calculate in DC using the virtual short circuit model.

	8	Obtain the inverting circuit input resistance as function of its R1 and R2 resistance values. Calculate Ri for the chosen R1 and R2 values.

In order to work with this concept, we will use the following circuit:



In the figure we have our opamp inverting circuit (in a dashed box) and one external Rm resistor that we will use to measure the circuit input resistance.
We don't want to affect too much our original circuit so, as R2 has a value of 1kΩ, we will use a much lower value of 100Ω for Rm.

This is not a requirement, however. As long as we measure Vi with ADC2 and Vo with ADC1, the response of the circuit will be properly measured regardless of the value of Rm. The same applies to the calculation of the input resistance.

The only impact that will produce the use of a Rm value not negligible compared to R2 is a reduction of the dynamic range of the Vi signal we can input to the circuit, as part of the original DAC1 voltage will drop on Rm.

	8	Calculate inverting circuit gain G2, bandwidth BW and input resistance Ri. Leave the 1kΩ resistor alone and don't use it in the calculations as it is not part of the dashed inverting amplifier box.

As in other cases, there are several ways to perform a measurement.

❶ In order to measure the input resistance for the circuit we can use:

$$R_i = \frac{\Delta V_i}{\Delta I_i}$$

First we sweep the DAC 1 values:

```
>>> data = slab.dcSweep(1,0.0,3.2,0.1)
```

We check that data is ok by showing the Vo(Vi) curve. We can also see how the input current depends on vi.

```
>>> vi = data[2]
>>> vo = data[1]
>>> ii = (data[3]-data[2])/100
>>> slab.plot11(vi,vo)
>>> slab.plot11(vi,ii)
```

Observe that the current is sometimes negative. This is consequence of the reference voltage Vr. Whenever Vi is less that Vr, the input current is negative.

Using the last ii(vi) curve you can obtain the input resistance from the slope in the center region. As the voltage is in the X axis, and the current in the Y axis, the slope is, in fact, 1/Ri.

❷ We can also use an alternative method using the first differences. Differences are, in discrete values, the equivalent to derivatives. The first difference of a vector of discrete values is defined as:

$$diff\ V[n] = V[n+1] - V[n]$$

The SLab module does not include the first difference function, we the NumPy module for that. We will import the module, calculate the Vi and Ii differences and obtain the input resistance from their division.

```
>>> import numpy as np
>>> dvi = np.diff(vi)
>>> dii = np.diff(ii)
>>> ri = dvi/dii
```

We can only plot two vectors of equal size, the first difference vector has one element less than the original vector, so we need to take out one element.

```
>>> slab.plot11(vi[0:-1],ri)
```

The higher and lower zones of the curve will be bad. In the center zone we should see a constant value equal to the input resistance Ri.

❸ There is a third method to obtain the input resistance. We can input a wave signal and obtain the information from the waves registered at the ADCs.
We use the ***returnData*** paramter so that the ***wavePlot*** command returns the plotted data and we can store it.

```
>>> slab.waveSine(1.2,2.1,100)
>>> slab.setWaveFrequency(100)
>>> slab.setTransientStorage(500,3)
>>> data = slab.wavePlot(returnData=True)
```

As Rm is small compared with R2, V_{DAC1} measured at ADC3 should be similar to Vi measured at ADC2. Now we can obtain the information about all the waves:

```
>>> import slab_meas as meas
>>> meas.analyze(data)
```

We can now calculate Ri from the amplitudes peak to peak (p2p) measured at ADC3 and ADC2.

$$i_{p2p} = \frac{\Delta V_{DAC1} - \Delta V_i}{100\Omega} = \frac{ADC3_{p2p} - ADC2_{p2p}}{100\Omega} \qquad Ri = \frac{\Delta V_i}{\Delta I_i} = \frac{ADC2_{p2p}}{i_{p2p}}$$

If you are lazy to perform the calculations yourself, you can use the SLab module for that. We use in this case the ***peak2peak*** command to analyze the data returned by ***wavePlot***.

```
>>> vdac = slab.peak2peak(data[3])
>>> vi = slab.peak2peak(data[2])
>>> ii = (vdac-vi)/100
>>> ri = vi/ii
>>> ri
```

At 100Hz, Vi and Vo should be in phase. If they weren't in phase, we should talk about **input impedance**, not input Resistance.

How to improve measurement quality

As commented before, the use of a low value 100Ω for the Rm resistor is not a requirement in order to obtain proper measurements.

If you obtain measurements that are not good enough due to the small voltage drop on Rm, you can improve them increasing the Rm value and repeating the measurements.
Note that increasing Rm reduces the effective voltage at Vi so you could need to increase the Vdac amplitude to compensate.

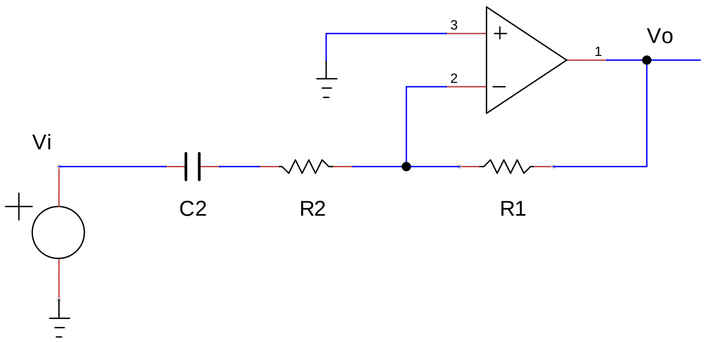
🛠	9	Mount the proposed circuit. Measure the input resistance in all the proposed ways. Do they all coincide with the theoretical calculations?

DC Decoupling

In this section we return to the "s" domain circuit analysis. This is the basic tool to any frequency analysis on a circuit.

As in the non inverting amplifier, sometimes you don't want to amplify the DC component of a signal. Moreover, if the signal to amplify has no low component frequencies, using a band pass amplifier eliminates low frequency noise.

You can block DC and convert the circuit to band pass operation just adding a C2 capacitor in series with R2. The order doesn't matter but it is usually located before R2.



We can calculate the K and β values for this circuit in order to obtain the complete transfer function. But we don't want to do that. We will use the virtual short circuit model to obtain the low frequency operation of the circuit and then we will add the high frequency limit due to the GBW product.

Observe that we connect the $V_{(+)}$ terminal to GND. That greatly eases the calculations and, as we will see, has low impact on the final results. So we will leave Vr alone for now.

	10	Obtain the $H_{LF}(s)$ low frequency transfer function of the system using the virtual short circuit model. Leave it as function of R1, R2 and C2 How many poles and zeros do we have? Where are they located?

You should have found a zero at zero frequency and a real pole p_L. That makes the circuit high pass and the corner frequency is p_L. It is easy to see that the circuit is high pass in its limits because:

$$H_{LF}(0) = 0 \qquad H_{LF}(\infty) = -\frac{R1}{R2}$$

In fact, $H(\infty)$ was the DC gain H(0) before we added C2.

The glory details

We have another way to see that. As we know, a capacitor has a value that is related to the "s" transform as:

$$Z_C(s) = \frac{1}{C \cdot s}$$

When we have sinusoidal signals, the long term response is:

$$Z_C(f) = \left.\frac{1}{C \cdot s}\right|_{S=j\omega} = \frac{1}{j\omega C}$$

Remember now that C2 is in series with R2, then:

$$Z_{Series} = \frac{1}{j\omega C2} + R2$$

Then, for high enough frequencies:

$$Z_{Series}\,|_{\omega\to\infty} = \left.\frac{1}{j\omega C2} + R2\right|_{\omega\to\infty} = R2$$

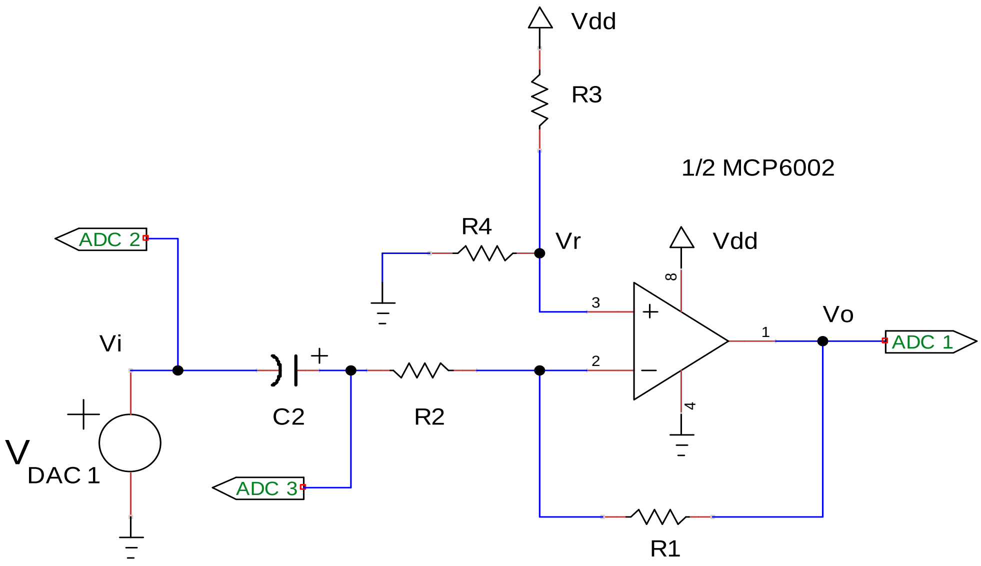
So, at high frequencies our circuit behaves just like C2 was shorted.

The circuit, in practice, cannot have a constant $H(\infty)$ due to the GBW limitation. We see that for frequencies much greater than p_L the circuit behaves like the previous circuit in DC, with C2 shorted. For high frequencies we will have the same GBW limitation as before:

$$p_H = \frac{GBW}{G_N}$$

Don’t forget that in this circuit noise gain is not equal to signal gain.

We will now add the Vr generation on the circuit.



As Vr is constant, and $H_{LF}(0) = 0$, the only effect of R3 and R4 is setting the DC voltage of the output to the Vr voltage regardless of the DC voltage at Vi. You can solve the above circuit at DC using virtual short circuit if you are not convinced. Remember that DC current at C2, as in any capacitor at DC, is zero.

You can also use the superposition property of linear circuits. Remember that Vr acts as a voltage source. As the circuit is linear, its output will be a linear function of its inputs:

$$V_O = f_{Lin}(V_i, V_r)$$

If f_{Lin} is linear, it satisfies the superposition property:

$$V_O = f_{Lin}(V_i, 0) + f_{Lin}(0, V_r)$$

The first term $F_{Lin}(V_i,0)$ is the previously response calculated for the circuit when Vr = 0. The second term $F_{Lin}(0,Vr)$ is the response of the circuit when Vi = 0. As in this case the circuit doesn’t include any time dependent signal, this function is a time independent constant.
At DC, current at C2 is zero, so the circuit behaves like a follower and Vo = Vr.

As a conclusion it is demonstrated that the only effect of connecting $V_{(+)}$ to Vr instead of GND is changing the DC voltage of the circuit from zero to Vr.

We will consider the same previous set of values for this circuit and we will add a capacitor.

R1 = 2,2 kΩ R2 = 1 kΩ C2 = 10 µF R3 = R4 = 33 kΩ

The capacitor is probably electrolytic so it will have polarity. If we connect the positive longer terminal to R2, as shown in the figure; we need to guarantee that Vi voltage generated on DAC1 is always below the reference voltage Vr of Vdd/2.

	11	Calculate, for the indicated values, the p_L pole position. The p_H pole position, due to the GBW, and the medium frequency gain G = -R1/R2 are known values from previous calculations.

Now we can mount the circuit to measure it. Set DAC voltages to zero by issuing the zero command so that you guarantee that C2 is never reverse polarized before starting the measurements:

```
>>> slab.zero()
```

We can perform a bode plot for low frequencies of operation. We set the peaks of the DAC 1 generated waveform so that it is always below Vr. That way we guarantee that we never reverse polarize C2. Remember we have to import the AC module.

```
>>> import slab_ac as ac
>>> ac.bodeResponse(0.4,1.2,5,1000,10)
```

We should see that the bode starts with a 20 dB/dec slope due to the zero a zero frequency. We should also see the p_L pole position and the constant medium frequency gain for frequencies above the pole.

We can also see the operation of the system against a waveform. This time, to make things different, we will use a triangle wave:

```
>>> slab.waveTriangle(0.4,1.2,100)
>>> slab.setWaveFrequency(100)
>>> slab.tranStore(500,2)
>>> slab.wavePlot()
```

You should see both the input signal at ADC 2 and the output signal at ADC 1. Perhaps the output signal is not exactly triangular. This could be due to the fact that we are not too far from the pole.

Observe that, as the circuit blocks DC, the output is exactly the same if we change the offset of the input signal:

```
>>> slab.waveTriangle(0.6,1.4,100)
>>> slab.wavePlot()
```

Regarding the wave distortion due to the pole proximity, you can try to generate a wave just over the pole frequency to see the effects on the output signal:

```
>>> slab.setWaveFrequency(16)
>>> slab.wavePlot()
```

In general, you need to be in a flat zone of the bode plot in order to prevent this kind of distortion. The system, however, is linear, so a sine wave won't be distorted at all:

```
>>> slab.waveSine(0.6,1.4,100)
>>> slab.wavePlot()
```

Remember that we need to guarantee that the capacitor C2 is never reverse polarized. You can add ADC3 to the ploted data by changing the transient storage:

```
>>> slab.tranStore(500,3)
>>> slab.wavePlot()
```

	12	Mount and perform the proposed measurements on the circuit that includes C2. Compare the pole position with the calculated value. Check also that C2 is not reverse polarized.

Limiting the bandwidth

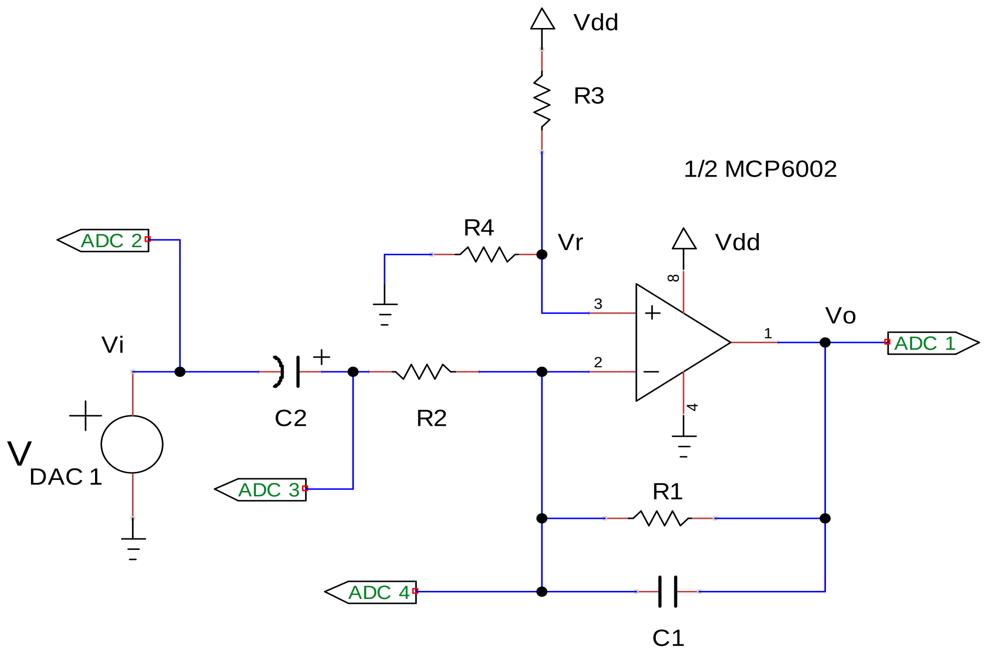
Up to this point, all the opamp circuits we have build had its high frequency operation limited by the GBW product. That means that low *noise* gain circuits have high bandwidth and high *noise* gain circuits have low bandwidth. Sometimes, however, we want to control the circuit high cut-off frequency. As you know, amplifying at frequencies out of the input signal range can only increase noise.

That's easy on this circuit, because of its basic DC gain formula:

$$G = -\frac{R1}{R2}$$

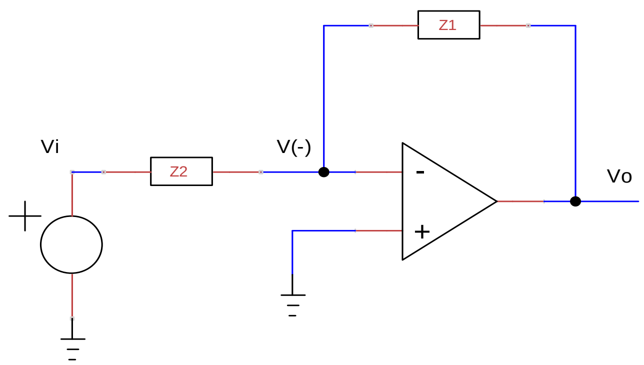
In order to block DC we have made the denominator infinite at zero frequency by adding C2. In order to reduce the bandwidth we can we can make the numerator go to zero from a given frequency.

One good way to do that is putting a capacitor C1 in parallel with R1.



The structure of the circuit is similar DC circuit but instead of R2 we have R2 in series with C2 and instead of R1 we have R1 in parallel with C1. In this case C2 adds a zero and a pole whereas C1 only adds a pole.

You can think of the circuit as a particular case where instead of R1 and R2 we have two arbitrary sets of resistors and capacitors that define two impedances Z1 and Z2. Remember that we can add the effect of Vr later so, for now it is easier to perform calculations if we connect $V_{(+)}$ to GND.



In our case Z2 is:

$$Z2 = R2 + C2 = R2 + \frac{1}{C2 \cdot s} = \frac{1 + R2 \cdot C2 \cdot s}{C2 \cdot s}$$

Note that Z2 at zero frequency is infinite and at infinite frequency is R2. This is coherent with the analyzed and measured circuit behavior.

In the case of Z1 we have:

$$Z1 = R1||C1 = \frac{R1 \cdot \frac{1}{C1 \cdot s}}{R1 + \frac{1}{C1 \cdot s}} = \frac{R1}{1 + R1 \cdot C1 \cdot s}$$

Observe now that at zero frequency Z1 is R1 and that at infinite frequency Z1 is zero, just like we want it to be.

We can now obtain the circuit transfer function using the VSC (virtual short circuit) model. In fact, as the circuit is the same as we had at the start of this document, we just need to replace R1 and R2 with Z1 and Z2:

$$H(s) = -\frac{Z1(s)}{Z2(s)}$$

The transfer function will have a zero frequency zero and two real two poles. That means that it is a pass band circuit: Gain is zero at zero frequency. Due to the zero, increases at 20dB/decade up to the first pole where it goes flat. At the second pole it decays at -20dB/decade.

$$H(s) = -\frac{Z1(s)}{Z2(s)} = \frac{-K \cdot s}{\left(1 + \frac{s}{p_1}\right)\left(1 + \frac{s}{p_2}\right)}$$

The G_{MF} medium frequency gain of the circuit is defined as the gain when the response is flat, that is, for frequencies between the two real poles p1 and p2. As we include C1 to define the high frequency cut-off, we can assume that the pole p1 related to C1 is at higher frequency that the pole p2 related to C2. **Poles are not in order**.

We can determine the G_{MF} medium frequency gain as the value of H(s) when we can consider s much greater than the first pole and much lower that the second pole.

$$G_{MF} = \left.\frac{-K \cdot s}{\left(1 + \frac{s}{p_1}\right)\left(1 + \frac{s}{p_2}\right)}\right|_{\substack{Between \\ Poles}} \approx \frac{-K \cdot s}{(1)\left(\frac{s}{p_2}\right)} = -K \cdot p_2$$

	13	Obtain the circuit transfer function H(s) as functions of R1, R2, C1 and C2. What are the values of H(0) and H(∞) ? Obtain the poles and zeros? What is the medium frequency gain G_{MF}?

Remember that we have used the virtual shortcircuit method to obtain the poles and zeros. We have not used the one pole operational model so we have missed the high frequency pole it provides. It is, however, at the same frequency as always:

$$p_3 = \frac{GBW}{G_N}$$

Our system has now a total of one zero and three poles.

In order to test a practical case, we will use the same values as before but we will add a 100 nF C1 capacitor:

R1 = 2,2 kΩ   R2 = 1 kΩ   C1 = 100nF   C2 = 10 μF   R3 = R4 = 33 kΩ

	14	Calculate all the poles and zeros values for the indicated values. Calculate also the medium frequency gain G_{MF}.

Now we can measure the circuit.

We can go straight to the bode plot. This time we will test up to 9 kHz. With the SLab system we can see the two first poles but we cannot reach the third one.

```
>>> ac.bodeResponse(0.4,1.2,5,9000)
```

Perhaps you see a discontinuity on the phase response because it jumps to 180º when it falls below -180º.

	15	Mount the circuit and obtain the bode plot. Check the values of p1, p2 and G_{MF}.

Now we can perform some pretty measurements. First we define a log spaced frequency range. We use the default 10 points per decade.

```
>>> f = ac.logRange(5,9000)
```

Then we perform complex gain calculations for those frequencies for all four ADC channels:

```
>>> g1,g2,g3,g4 = ac.freqResponseAll(0.4,0.8,f)
```

We can obtain the original bode plot:

```
>>> ac.plotBode(f,g1)
```

Or we can show it in linear magnitudes instead of dB. Observe that in this case the title is "Frequency Plot" as using linear magnitudes is not valid for a real "Bode" plot.

```
>>> ac.plotBode(f,g1,linear=True)
```

We can also obtain the input impedance. In order to do that, we convert from gains to voltages using the 0.4 V amplitude of the input signal. Then we compute the input current from the voltages on R2 obtained on ADC 3 and ADC 4.
From the current we obtain the impedance as the input voltage is the 0.4 V amplitude (and zero phase) sine generated on DAC 1:

```
>>> ii = 0.4*(g3-g4)/1000
>>> Zi = 0.4/ii
>>> ac.plotBode(f,Zi,linear=True)
```

We should see that the input impedance has a 1 kΩ real value, as expected, for frequencies above the first pole. Below the pole, the absolute value increases and the phase changes so we no longer have resistive impedance.

Last comments

In this project we have seen different circuits based on the inverting amplifier circuit topology.
This time we have seen also that the GBW relates to the noise gain that can be different from the signal gain.
Like in the non inverting amplifier, we see that we can block the DC input voltage.
We have also seen that we can limit the circuit bandwidth just adding a capacitor. You could also do the same in the non inverting amplifier adding a capacitor in parallel with the R1 feedback resistor, but beware that this amplifier cannot attenuate no matter how low is R1 or the combination of R1 and whatever is in parallel, so you are limit to make fall the gain to 1 until you reach the GBW limit.

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/