ME 103 Experimentation and Measurements
Spring 2026 George Anwar

HW3

Due: Monday, March 9th at 11:59 pm

- This homework will cover the Fourier analysis, filtering, Bode plots, frequency response, and dynamic responses of systems.
- In all of the questions, **show your work**, not just the final answer. Unless we explicitly state otherwise, you may expect full credit only if you explain your work succinctly, but clearly and convincingly. For coding questions, attach a screenshot of your code and output.
- Present your answers with **a suitable number of significant figures** for each question. Show your work, including a mathematical formula or the MATLAB or Python code you wrote, before reaching the result. You may need to install the Signal Processing Toolbox if using MATLAB.
- Throughout this assignment, neglect systematic (bias) errors. Also, assume a normal distribution for the underlying distribution (population) if necessary.
- If you have a confirmed disability that precludes you from complying fully with these instructions or with any other parameter associated with this problem set, please alert us immediately about reasonable accommodations afforded to you by the DSP Office on campus.
- **Start early. Some of the material is prerequisite material not covered in lecture; you are responsible for finding resources to understand it.**

Deliverables

Submit a PDF of your homework to the **Gradescope assignment** entitled "{Your Name} HW3". You **must** typeset your homework in LaTeX (submit PDF format, not .doc/.docx format) . Mac Preview, PDF Expert, and FoxIt PDF Reader, among others, have tools to let you sign a PDF file. We want to make *extra clear* the consequences of cheating.

0 Honor Code

I will adhere to the Berkeley Honor Code: specifically, as a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. Failure to comply with these guidelines can be considered an academic integrity violation. Please email Professor Anwar ganwar@berkeley.edu if you have any questions!

- **List all collaborators. If you worked alone, then you must explicitly state so. Read the following statement and sign below if you agree**:

  "I certify that all solutions in this document are entirely my own and that I have not looked at anyone else's solution. I have given credit to all external sources I consulted."

  Signature : ______________________ *Date* : ______________________

  While discussions are encouraged, *everything* in your solution must be your (and only your) creation. Furthermore, all external material (i.e., *anything* outside lectures and assigned readings, including figures and pictures) should be cited properly. We wish to remind you that consequences of academic misconduct are *particularly severe*!

- Violation of the Code of Conduct will result in a **zero** on this assignment and may also result in disciplinary action.

1 Fourier meets Coding [28 pts]

The purpose of this assignment is to familiarize you with concepts related to Fourier analysis, sometimes referred to as spectral analysis. Fourier analysis has wide applications from electrical systems and vibratory systems to signal processing and even fluid mechanics.

The Continuous Time Fourier Transform (CTFT) Synthesis Equation is defined as follows:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt \tag{1}$$

Here, again, ω is in units of rad/s, not Hz. Note how similar this definition is to the definition of the Laplace Transform, which you may be more familiar with. It is important to recognize that no information is gained or lost when evaluating the Fourier transform. One can transform back and forth between the time and frequency domains without losing any information. Similarly, no information is created. These properties are retained when the Fourier transform is discretized.

The Discrete Fourier Transform (DFT) is analogous to the continuous formulation:

$$F_k = \sum_{m=1}^{N} f_m \cdot e^{-\frac{i2\pi}{N}(k-1)(m-1)}, \quad \forall k \in \mathbb{Z} \tag{2}$$

where N is the total number of discrete points (e.g. data points) in one period T, f_m is the real value of the m^{th} data point, and F_k is k^{th} complex data point of the transformed function. Time corresponds to m; frequency corresponds to k. It should be noted, however, that m and k are simply labels for each data point, not the actual value of time and frequency. For example, the m^{th} data point f_m corresponds to the time $t = \frac{(m-1)T}{N}$. Similarly, the k^{th} transformed data point F_k corresponds to the frequency (in rad/s) $\omega = \frac{(k-1)\omega_0}{N}$.

In practice, the DFT equation above is rarely used. It is cumbersome and calculating every term in the series is unnecessary. Instead, the Fast Fourier Transform (FFT) is widely used. The FFT is a more computationally efficient method for calculating Eq. (2). The FFT is a method to calculate the DFT.

(a) (i) [6 pts] Write and show your **own** MATLAB code that will evaluate $(F_1, \ldots, F_N)$ directly using Eq. (2) for a **unitless** dataset in `xt1` (in `HW3data.mat`)

Solution: TODO

(ii) [4 pts] Compare the time it takes for your code to run compared to that for the `fft` function in MATLAB. *Hint*: use the `tic` and `toc` functions to measure code execution time.

Solution: TODO

(b) (i) [4 pts] Given $(f_1, \ldots, f_N) \in \mathbb{R}^N$, assume you now evaluate $(F_1, \ldots, F_N)$ by a **slow calculator** that computes one natural exponent e^z or one basic operation (i.e. addition and multiplication) a second. How much time does it take to evaluate $(F_1, \ldots, F_N)$ when naively using Eq. (2)? Ignore data input/output time.

Solution: TODO

(ii) [3 pts] **The first step from DFT to FFT**: Discuss how to reduce the amount of computation time of the naive DFT, regardless of what real values are assigned to $(f_1, \ldots, f_N)$. *Hint*: express Eq.(2) in matrix form and find symmetries.

Solution: TODO

Much like the continuous Fourier transform, the DFT assumes that the function is *periodic*, which may not be the case. Furthermore, the DFT violates one of the original assumptions: that the function is finitely discontinuous. By definition, a periodic discrete function is *not* finitely discontinuous. Violating these assumptions leads to some interesting behavior. For the sake of clarity, the remainder of this document will discuss Fourier analysis in the context of data collection, unless explicitly stated otherwise.

The assumption of periodicity in the time domain requires that the transformed function F_k is also periodic in the frequency domain. As a result, F_k reflects at a certain frequency called the Nyquist frequency. This reflection is the cause of *aliasing*. Aliasing is when a high frequency signal appears to the measurer as a lower frequency signal. The Nyquist frequency occurs at half of the sampling frequency, i.e.,

$$f_{\text{Nyquist}} = \frac{1}{2} f_s = \frac{1}{2} \cdot \frac{N}{T}, \tag{3}$$

where f in Eq.(3) means frequency in Hz, having the relationship with ω as $\omega = 2\pi f$. In other words, any data points beyond the Nyquist frequency are meaningless. They are just a reflection of the data below the Nyquist frequency.

Furthermore, the frequency function can reflect non-positive frequencies like $k = 0, -1, -2, \ldots$. However, we only evaluated the DFT for positive integers for k. Why is that? When doing Fourier analysis, we often plot the magnitude of the function at each frequency. The magnitude contribution of a negative frequency is identical to its positive counterpart (recall the relationship between A_{-n} and A_n obtained from (a). That is, half of the "energy" of the function is in the negative frequency domain. Rather than calculate those magnitudes with Eq.(2), it is easier to double the positive frequency magnitudes to account for the lost "energy." It has become convention to double the magnitude (except at zero and Nyquist frequencies) when processing DFT results.

(c) (i) [3 pts] Calculate and numerically show the Nyquist frequency for the dataset in `xt1`. Use the fact that the data point `xt1(k)` is measured at time `t(k)` in seconds for any `k`.

Solution: TODO

(ii) [3 pts] Modify the code you wrote for (b). First, truncate your DFT vector so that only frequencies below the Nyquist frequency are included. Second, account for the reflection into the negative frequency domain. Show the code you modified.

Solution: TODO

After (c), your code should look similar to the code in MATLAB's `fft()` documentation here (also, see Figure 1). However, the first line in Figure 1 remains unexplained. The first line is a normalization of the data by the number of data points in the set (`L`, in Figure 1). Since the DFT is a summation, more data points generally equate to a larger magnitude. Normalization allows us to compare data sets of different sizes. This form of normalization is one convention; it is not the only form of normalization available.

```
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
```

Figure 1: Part of MATLAB's `fft()` documentation.

Furthermore, the first line takes the absolute value of the frequency function (Y, in Figure 1). Before applying the absolute value, the frequency function is complex, containing both magnitude and phase/argument information of each frequency content. The absolute value of the frequency function is a power spectrum of the frequency content; it must be real-valued.

Finally, note that the third line in Figure 1 omits the first index of `P1`, which should correspond to $\omega_n = 0$. When $\omega_n = 0$, the only content is the mean value of the periodic function (e.g., the DC offset in electrical signals). Since the $\omega_n = 0$ value is not reflected into the negative frequency domain, it does not need to be doubled, thus it is omitted in line 3. The same argument holds when ω_n corresponds to the Nyquist frequency (i.e., $\omega_n = 2\pi f_{\text{Nyq}}$). At this point, the DFT data should be fully processed for interpretation.

(d) (i) [3 pts] Plot the single-sided power spectrum from your DFT result. Make sure to have the data points match the correct frequencies (in Hz) and magnitudes (using the normalization in Figure 1).

Solution: TODO

(ii) [1 pt] Report the frequencies and magnitudes of all of the notable peaks seen in the power spectrum.

Solution: TODO

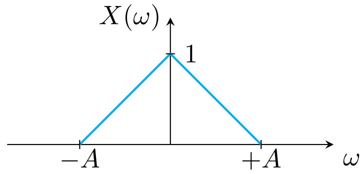
(e) [1 pt] **Validation**: Conduct the same procedure as Problem 4 but use MATLAB's `fft()` instead of your DFT function. Compare the FFT plot with the DFT one and confirm that they are the same.

Solution: TODO

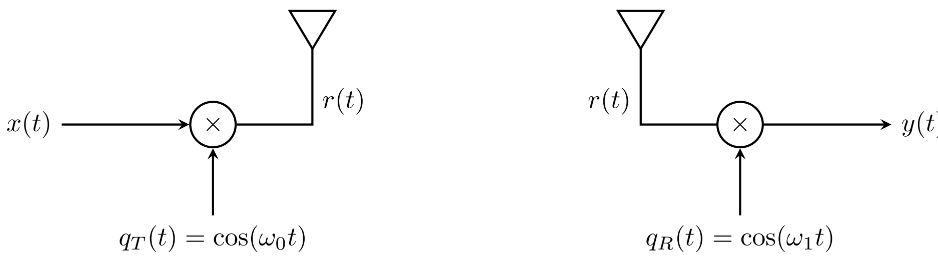
While the results from Problems (d) and (e) are useful, there are several other useful tricks for clarifying the results of DFT or FFT. One of the tricks is windowing. Windows are piecewise functions that are identically zero when outside the data domain but multiply the data by some factor within the data domain. Windows ensure that a set of data meets the requirement of periodicity: the data begins and ends at the same value. When applying a DFT on a finite set of data, a rectangular window is implicitly being applied. A rectangular window multiplies the function by 1 within the domain; multiplies by zero outside the domain. Other common windows include the Hann/Hamming windows and Blackman windows.

2 Heterodyning and Amplitude Modulation [15 pts]

We are given a bandlimited continuous time transmission signal $x(t)$ with the following triangular spectrum shown below.



A spectrum is a graph that shows how much of each frequency is present in the signal. The following diagram shows an (AM) amplitude modulation-demodulation scheme to communicate the signal $x(t)$ to a receiver. *Note*: that the circle with $\times$ means multiplication and the triangle is the antenna that transmits and receives.



There is also a frequency mismatch (ϵ) between the transmitter (LHS) and receiver (RHS) carrier signals q_T and q_R, respectively. In particular, assume that

$$0 < \epsilon \ll A \quad \text{and} \quad A < \omega_0 = \omega_1 + \epsilon$$

(a) [3 pts] Determine reasonably simple expressions for the signals r and y in terms of ω_0. You may or may not find the following trigonometric identity useful:

$$\cos\alpha\cos\beta = \frac{1}{2}[\cos(\alpha+\beta) + \cos(\alpha-\beta)]$$

Solution: TODO

(b) [10 pts] Provide well-labeled plots of $R(\omega)$ and $Y(\omega)$, the spectra of the signals r and y, respectively.

Solution: TODO

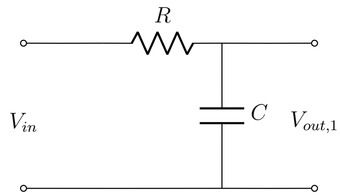
(c) [2 pts] Explain why the information-bearing signal x is irrecoverable, even if we send the signal y through an ideal low-pass filter. *Hint*: recall that ϵ is unknown.

Solution: TODO

3 Frequency Responses of Low-pass Filters [29 pts]

This question is about different types of low-pass filter and how their frequency responses differ.

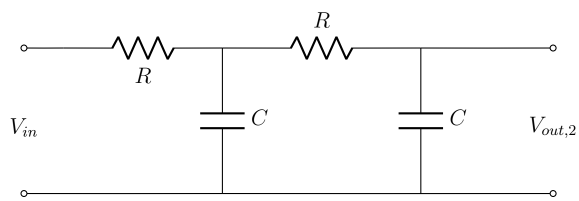
(a) [4 pts] First consider the basic passive, first-order low-pass RC filter:



Confirm that you understand why this circuit functions as a low-pass filter by re-deriving the complex transfer function, $V_{out,1}/V_{in}$. Show your derivation.

Solution: TODO

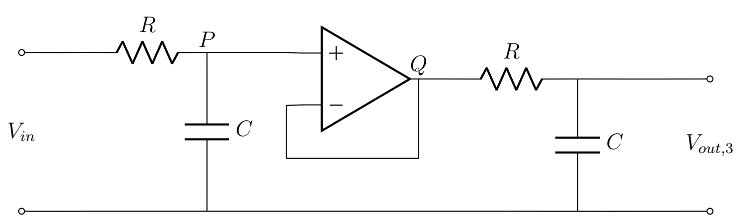
(b) [6 pts] Now consider what would happen if we connected together two RC stages to make a rudimentary second-order filter:



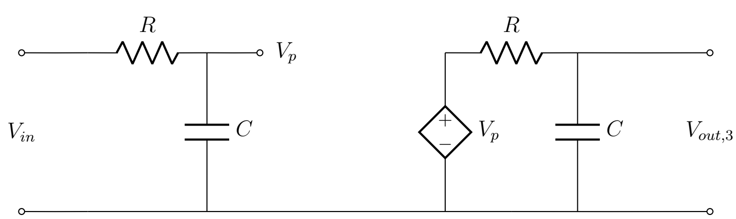
Derive the complex transfer function $V_{out,2}/V_{in}$ for this circuit, using the voltage-divider principle and complex impedances.

Solution: TODO

(c) [4 pts] The impedances of the two RC stages above interact with each other, which made computing the transfer function a bit more complicated than multiplying together the complex transfer functions for two first-order RC filters. Instead, we can design filters where the different stages are separated from each other by a 'buffer', an active circuit whose output voltage equals its input voltage and that has high input impedance and low output impedance. Two RC stages are 'cascaded' together as follows:



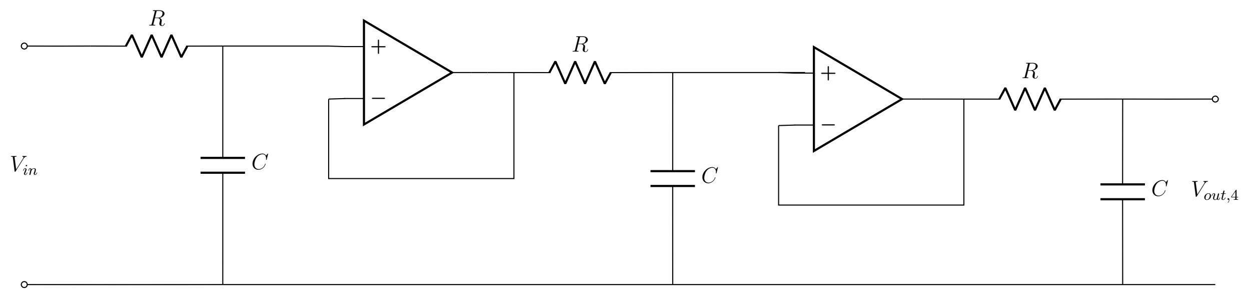
For simplicity, we can model the voltage follower (unit gain buffer op-amp) as the following circuit



Here, the high input and low output impedances of the buffer mean that the RC stages' impedances do not interact, and the overall transfer function is the product of the transfer functions of the two RC stages. Thus, write down the complex transfer function $V_{out,3}/V_{in}$ for this circuit.

Solution: TODO

(d) [3 pt] Similarly, a third-order cascaded RC filter could be constructed as follows:



Write down the complex transfer function $V_{out,4}/V_{in}$ for this circuit.

Solution: TODO

(e) [7 pts] Create a Bode plot, with magnitude and phase on separate sets of axes, on which you superimpose the frequency responses of each of the four filters considered above. Assume RC = (1 second) for all filters. Also add the frequency response of the third-order passive Butterworth filter:

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^3 + 2s^2 + 2s + 1}, \quad \text{where } s = j\omega$$

You will need the Control System Toolbox installed to use the `bode()` and `tf()` functions. This toolbox is freely available to registered students.

Solution: TODO

(f) [3 points] Comment on differences between the frequency responses of the rudimentary 2-stage RC filter, found in part (b), and the cascaded filter with buffer, found in part (c). What are some potential pros and cons of the two approaches?

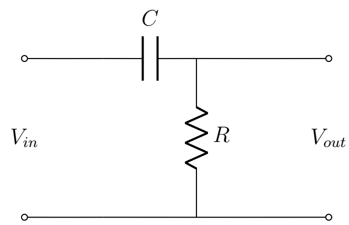
Solution: TODO

(g) [2 points] Comment on differences between the frequency responses of the 3-stage cascaded RC filter, found in part (d), and the third-order Butterworth filter. What are some potential pros and cons of the two approaches?

Solution: TODO

4 Frequency Response of Other Filters [9 pts]

Consider the following filter (note that the resistor and capacitor are switched compared to the low-pass filters in the previous question):



(a) [3 pts] Derive the complex transfer function, and expressions for the phase and magnitude of the frequency response.

Solution: TODO

(b) [1 pt] In a few words, describe the behavior of this filter.

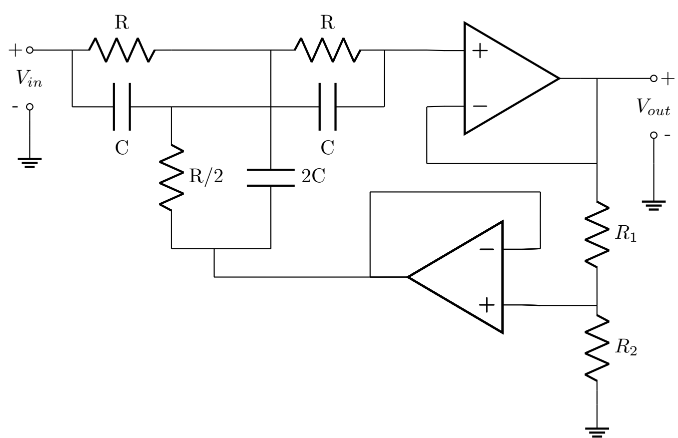
Solution: TODO

(c) [1 pt] Discuss any potential purpose/application(s) that you can envisage for a filter of this kind when making measurements.

Solution: TODO

Now, consider the following transfer function and circuit for a filter, which is called a 'notch' filter:

$$\frac{s^2 + \omega_0^2}{s^2 + Q^{-1}s + \omega_0^2}, \quad \text{where } s = j\omega$$



(d) [2 pts] Plot a Bode plot of this filter for a central frequency of $\omega_0 = 120\pi$ rad/s, and a quality factor, $Q = 10$.

Solution: TODO

(e) [1 pt] What frequency (in Hz) does the filter suppress most strongly? What could be an application of this filter in measurement?

Solution: TODO

(f) [1 pt] What would you expect to happen if you made the quality factor smaller? Try plotting the transfer function for $Q = 1$. Describe what changes and what any implications might be for measurement.

Solution: TODO

5 DC Motor Filtering [10 pts]

The speed of a spindle is measured using a small DC motor acting as a generator. The generated voltage is converted into a digital form using an Analogue-to-Digital converter. Commutation in the motor causes a high frequency ripple in the generated voltage, with a peak-to-peak amplitude of 1 V and a frequency which is 100 times that of the rotating spindle.

It is required to provide an **inverting** first order filter that will amplify the generated voltage signal by a factor of 10; it is also required to attenuate the ripple by 60 dB when the spindle speed is 200 Hz.

(a) [4 pts] Sketch the Bode magnitude and phase shift diagrams of the overall filter circuit by hand, indicating all frequencies and gains of interest.

Solution: TODO

(b) [3 pts] Draw a circuit diagram of the required filter arrangement, including all relevant parameters and circuit elements.

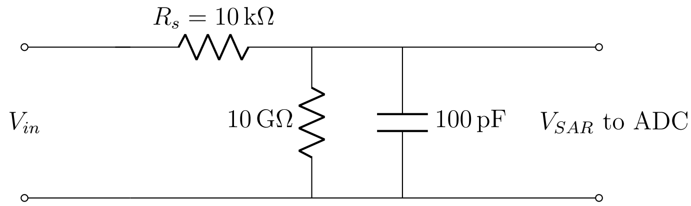
Solution: TODO

(c) [3 pts] If only 1μF capacitors are available, determine values for all resistors in the circuit.

Solution: TODO

6 Settling Error and Time [9 pts]

Consider the following apparatus connected to the input of a successive approximation register (SAR) ADC. The source resistance (inherent in the apparatus generating the signal) is $R_s = 10\ k\Omega$ and the input impedance of the ADC interface is represented by a 10 $G\Omega$ resistor in parallel with a 100 pF capacitor C_{in}.



The voltage seen by the ADC is v_{SAR}. Initially, $v_{\mathrm{SAR}} = v_{\mathrm{in}} = 0$. Then v_{in} undergoes a step change to $v_{\mathrm{in}} = 1$ V at time $t = 0$. The circuit will charge up as follows:

$$v_{\mathrm{SAR}}(t) = (1\ \mathrm{V})\left(1 - \exp\left(-\frac{t}{R_s C_{\mathrm{in}}}\right)\right).$$

Notes: We ignore the input 10 GΩ resistance as it is so high, and assume the input impedance of the ADC circuit itself (the circuit that lies to the right of the 10 GΩ resistor and 100 pF capacitance) is large.

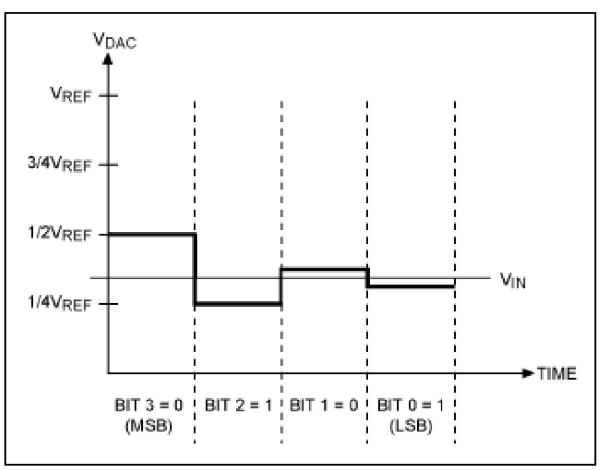
(a) [2 pts] Plot the percentage error between v_{SAR} and v_{in} for time from 0 to 100 μs. Use your judgement to decide whether to use logarithmic, linear, or some other type of scale for the graph.

Solution: TODO

(b) [1 pt] Calculate how long v_{SAR} takes to fall to an error of 1% after v_{in} changes.

Solution: TODO

(c) [2 pts] The ADC has 16 bits, and the clock frequency of the ADC, f_{clock}, is 10 MHz. In lecture, we described a naïve configuration for an ADC in which the register counted up from 0 and incremented one LSB (least significant bit) at a time, once per clock pulse, until the output of the register matched the input voltage. In a real SAR ADC, the control logic is more sophisticated and faster. The register is initialized in the middle of the measurement range (for an N-bit ADC, at 2^{N-1}). Upon the first clock cycle, the most significant bit of the register is set to the correct value to represent the analog input, and at each subsequent clock cycle the next most significant bit is updated, until the least significant bit is reached and all bits of the register have been updated to represent the input signal as accurately as possible. The digital output matches the analog input. The sampling time of the ADC operated in this way can be approximated as N/f_{clock}. What is the maximum sampling frequency that this ADC can reasonably be operated at?

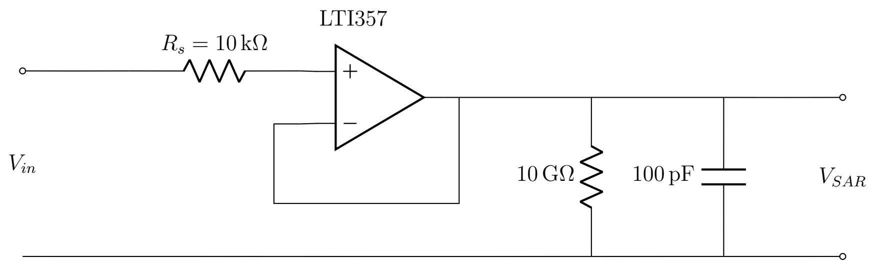


Solution: TODO

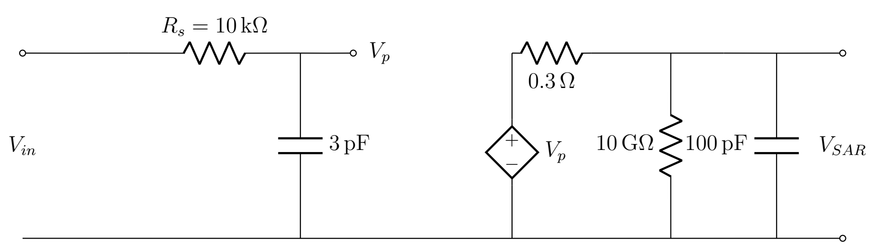
(d) [2 pts] Taking all these factors into account, what would be a conservative settling time to quote, to guarantee an error below 1% when recording this signal? Explain your reasoning (well-reasoned answers receive complete credit even if they aren't the answer we came up with).

Solution: TODO

(e) [1 pt] To try to reduce settling time, a buffer circuit is introduced between the apparatus and the ADC, consisting of an LT1357 operational amplifier configured as a voltage follower. The op-amp functions as a voltage-controlled voltage source that isolates the relatively high source impedance of the apparatus from the ADC's input capacitance. The LT1357 can be assumed to have an input capacitance of 3 pF and an output impedance of 0.3 Ohms, as shown below. For this part of the question, assume the slew rate of the LT1357 is unlimited.



The above buffer circuit can be modeled as



Update your estimate for the time taken for the V_{SAR} to fall to an error of 1%. Explain your reasoning briefly.

Solution: TODO

(f) [1 pt] Now consider that the specified slew rate of the LT1357 is in fact 600 $V/\mu s$. (The circuit model for the LT1357 shown above is approximate, so there are some other internal imperfections that determine the slew rate.) Is slew rate a significant limiter for the overall response speed of the measurement system? How does considering the slew rate affect your estimate of the settling time of the overall measurement, if at all? What if you had used a basic LM741 op amp, with a slew rate of 0.5 $V/\mu s$?

Solution: TODO