Marking Sheet for MecE 301 Reports

Student Name: Alex Diep

Marking Category	Inadequate (0-2 points)	Adequate (3-4 points)	Superior Work (5 points)	Grade (0-5)	Weight Factor	Total (%)
Title Page	Missing information.	All required information is given.	All required information is given. Appropriate title.		x1	
Introduction	Poorly defined problem or lacking motivation.	Explains the problem and why the work was done.	“Adequate” + motivates interest.		x1	
Procedure	Lists equipment used.	Describes procedure used so that reader could repeat experiments.	“Adequate” + organized in a logical fashion.		x2	
Discussion	Results were simply stated or information was missing.	Results were stated and interpreted.	Discussion shows in-depth interpretation of the results.		x3	
Conclusions/ Recommendations	Summarizes results.	Discusses objectives and how they were met.	Conclusions and recommendations provide insightful solutions.		x2	
Appendices	Missing data and calculations are not explained.	Data is given, calculations are explained.	Data and calculations are put into context.		x2	
Engineering Analysis	Major errors in calculations and analysis.	Minor errors in calculations and analysis.	No errors in calculations and analysis.		x4	
Data Presentation (Figures/Tables /Symbols)	Several errors in data presentation.	A few minor errors in data presentation.	Figures, tables, and symbols are presented to course standard.		x3	
Grammar, spelling, layout.	Several errors.	A few minor errors.	No grammar/spelling errors. Logical layout.		x2	
					Total:	

MEC E 301
Lab 2: Digital Measurement Techniques

by: Alex Diep

Date: October 3, 2023

CCID: abdiep
Student ID: 1664334
Section: D21

Contents

1 Introduction

The primary objective of this lab is to investigate and familiarize elements of digital measurements such as word length (number of bits), quantization, resolution, conversion rates, sampling, accuracy, and signal conditioning through the use of an Arduino Uno. These concepts are important in signal processing and digital communications and familiarity with them is important for future development.

2 Procedure

2.1 Equipment

The following equipment was used to perform the experiment: Arduino Uno, MEC E 301 Shield, computer, oscilloscope, and jumper wires.

2.2 Calibration

For extensive details on the calibration procedure, refer to the precis provided by the MEC E 301 course All measurements were taken from the serial monitor of the Arduino IDE. During calibration, different circuit components and reference voltages were used to measure the voltage output of the PCB.

First, measuring voltages with a 5V reference voltage was performed. As seen in Appendix A, Figure A.1, the `5V` and `GND` pins on the Arduino Uno were connected to the `5V_VIN` and `GND` pins on the PCB. The `A0` pin on the Arduino Uno was connected to the output pin, `2.500V`. After uploading the sketch, ten values were recorded from the serial monitor. The `2.500V` output pin was then swapped to `1.800V`, `1.024V`, and `0.102V` and ten values were recorded for each.

Next, measuring voltages with a 3.3V reference voltage and various circuit components was performed. As shown in Appendix A, Figure A.2, the AREF jumper on the MEC E 301 Shield was inserted, connecting the 3.3V reference voltage to the AREF pin on the Arduino Uno. First measurements of the voltages `2.500`, `1.800`, `1.024`, and `0.102` were performed with the 3.3V reference voltage. Then the output pins of the PCB were connected to the MEC E 301 Shield input-output pins to measure the various circuit components. The input-output pairs of [`D10_I`, `D10_O`], [`+/-10_I`, `+/-10_O`], and [`X10_I`, `X10_O`] measured voltages with a voltage divider, [-10, 10]V range, and amplifier respectively. Again, the voltages of `2.500`, `1.800`, `1.024`, and `0.102` were measured and recorded. Note, the amplifier only measured `0.102` because the other voltages were out of range.

2.3 Time Varying Voltage

The `SINE_OP` pin on the PCB was connected to the `A0` pin on the Arduino Uno as shown in Appendix A, Figure A.2 The code was modified to output a timestamp and voltage value and the baud rate was set to 115200. The serial monitor was opened and the voltage was recorded for 255 values. The values were copied into an Excel spreadsheet for analysis.

Lastly, measurement using an oscilloscope was performed. A schematic of the system is shown in Appendix A, Figure A.3. The `SINE_OP` pin on the PCB was connected to the oscilloscope. The `Autoset` button was pressed to automatically set the oscilloscope. Occasionally, the `Autoset` needed to be set to a sawtooth wave to get a good reading. Lastly, the `Measure` button was pressed and the oscilloscope was set to measure the peak-to-peak voltage, frequency, and mean voltage.

3 Results and Discussion

3.1 Calibration of the Arduino Uno Results and Discussion

The results can be found in B in Table B.1. The trade-off between resolution and range is that as the range increases, the resolution decreases. This is because the number of bits available to represent the range is fixed. From Appendix C, if the range is multiplied by a factor of k, then the resolution is also multiplied by a factor of k, increasing resolution error.

Repeatability is the maximum deviation between two measurements of the same reference quantity. Accuracy is the maximum deviation between the measured value and the true value. Repeatability is a measure of precision while accuracy is a measure of correctness.

The accuracy for the 5V reference voltage is $\pm 54.00\,\mathrm{mV}$ while the accuracy for the 3.3V reference voltage is $\pm 17.00\,\mathrm{mV}$. The accuracy of the Arduino Uno is improved using the 3.3V reference voltage. Many computers do not supply exactly 5V over the USB cable and the Arduino's A/D converter assumes the voltage is 5V. Supplying a known voltage to the AREF pin from the MEC E 301 Shield allows for more accurate measurements.

From Appendix D, if we use a conservative estimate for the accuracy of the voltage source, which occurs at 2.500V, then the accuracy is $\pm 1.250\,\mathrm{mV}$. ANSI/ISA 51.1 states that the accuracy of the standard can be ignored should the accuracy be one tenth of the instrument tested.

Comparing this to the accuracy of the Arduino Uno, in Table B.1 the accuracy of the voltage source is sufficient for all reference values except the setup with the 3.3V reference voltage and 10x amplifier, which has an experimental accuracy of $\pm 0.000\,\mathrm{mV}$.

Referring to Appendix B, the experimental accuracy was worse than the manufacturer's accuracy for all configurations except the 3.3V reference voltage with a 10x amplifier and the 3.3V reference voltage with a [-10, 10]V range.

The 5V reference voltage has an accuracy of $\pm 54.00\,\mathrm{mV}$ while the manufacturer's accuracy is $\pm 9.766\,\mathrm{mV}$. This may be attributed to the variable reference voltage supplied by the computer

The calculation is shown in Appendix E. The error at 0.102V is $\pm 1.020\,\mathrm{mV}$ and the error at 2.500V is $\pm 25.00\,\mathrm{mV}$. The error at 0.102 is not significant while the error at 2.500V is significant by

ANSI/ISA 51.1.

3.2 Time Varying Voltage Results and Discussion

Graphs of the time varying voltage can be found in Appendix A. Figure A.4 shows the time varying voltage for the 10-bit Arduino Uno. Figure A.5 shows the time varying voltage for the 5-bit Arduino Uno.

From Appendix F, Table F.5, the Arduino measurements of mean voltage, frequency, and peak-to-peak voltage are $1.135\,\mathrm{V}$, $47.266\,\mathrm{Hz}$, and $1.217\,\mathrm{V}$ respectively. The oscilloscope measurements are $1.17\,\mathrm{V}$, $48.54\,\mathrm{Hz}$, and $1.28\,\mathrm{V}$ respectively. The Arduino measurements are close to the oscilloscope measurements.

The 10 and 5 bit frequencies do not overlap with the oscilloscope frequency. The 5-bit peak to peak voltage and mean voltage overlap with the oscilloscope measurements. The 10-bit peak to peak voltage and mean voltage do not overlap with the oscilloscope measurements.

From Appendix F, Table F.8, both the 10-bit and 5-bit Arduino have a precision uncertainty of about $\pm 0.4\,\mathrm{Hz}$, which is about 1% of the measured frequency. The measurement of frequency with the Arduino can be improved by increasing the number of samples taken.

From Appendix F, Table F.11, the 5-bit measurement of peak-to-peak voltage had the highest total uncertainty. For the 5-bit measurements, for both the peak and troughs, the precision and systematic uncertainty were $\pm 0.01031\,\mathrm{V}$ and $\pm 0.05156\,\mathrm{V}$ respectively. The highest uncertainty is caused by systematic uncertainty.

From Table F.10, the systematic and precision uncertainty were approximately $\pm 0.01\,\mathrm{V}$ and $\pm 0.02\,\mathrm{V}$ respectively. Increasing the number of bits and increasing the linearity of the Arduino Uno would decrease the systematic uncertainty. Increasing the sample rate would decrease the precision uncertainty since the samples can *catch* the peak and troughs of the sine wave better.

4 Conclusion

The calibration section of the lab highlighted important concepts such as quantization, the trade-off between resolution and range, and the importance of a known reference voltage. The Arduino Uno was used to measure the voltage across different circuit components and reference voltages, and their impact on resolution, accuracy, and range was observed. The time varying voltage section of the lab highlighted the importance of sampling rate and resolution. Hands-on experience with an oscilloscope was obtained, and measurements were compared to the Arduino Uno. The Arduino Uno was able to somewhat agree with the much more expensive oscilloscope. Increasing the sample rate and resolution of the Arduino Uno would allow for better agreement with the oscilloscope.

A Figures

A.1 Schematic of the system

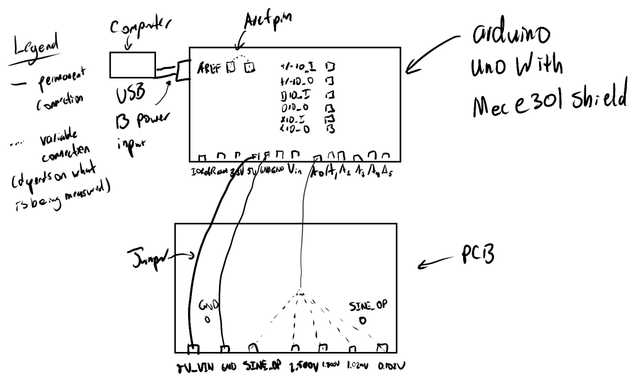

Figure A.1: Schematic of the system without a reference voltage used to measure the voltage across the PCB

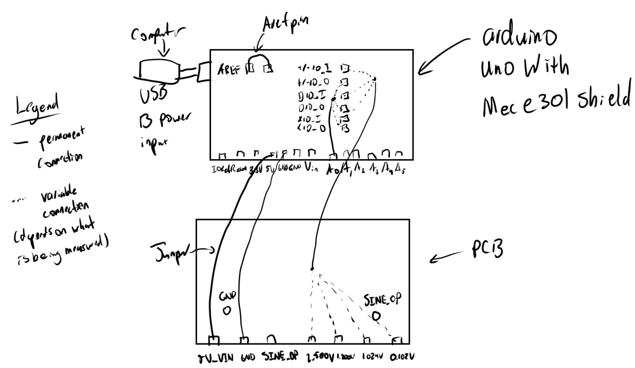

Figure A.2: Schematic of the system with a 3.3V reference voltage used to measure the voltage across the PCB

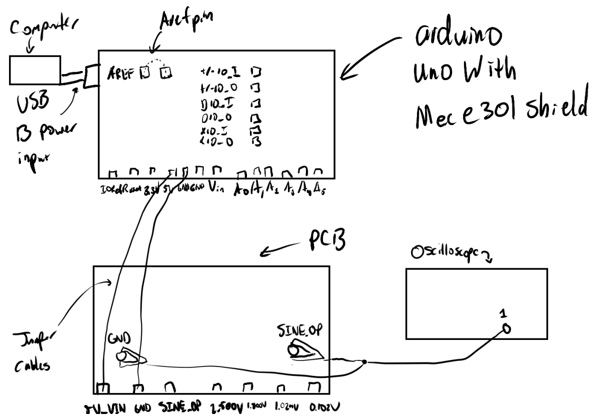

Figure A.3: Schematic of the system used to measure the voltage across the PCB with an oscilloscope

A.2 Plots of Time-Varying Signals

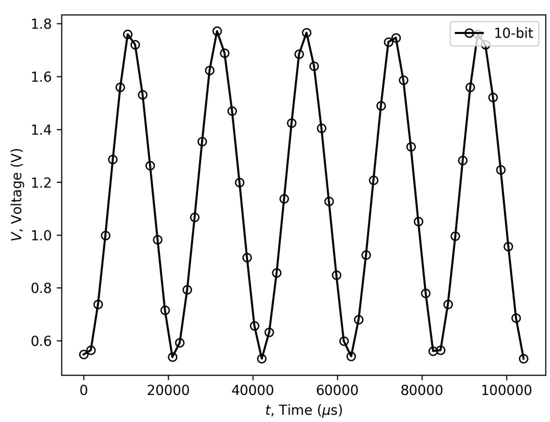

Figure A.4: Time varying signal of PCB 18 Measured with 10-bit ADC of the Arduino Uno

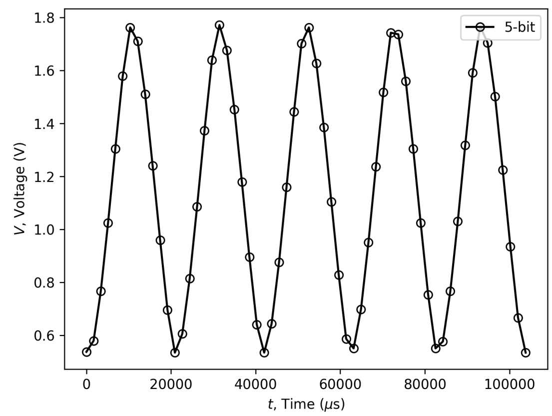

Figure A.5: Time varying signal of PCB 18 Measured with 5-bit ADC of the Arduino Uno

B Appendix: Arduino Uno Accuracy

Table B.1 summarizes the range, resolution, repeatability, accuracy, and manufacturer's accuracy for various ranges of the Arduino Uno. Sample calculations for the 5V reference voltage are shown below. Note, the manufacturer's accuracy is $\pm$ 2 LSBs.

$$\begin{aligned}
\text{Resolution} &= \frac{V_{\text{ru}} - V_{\text{rl}}}{2^n} \\
&= \frac{5.000 - 0.000}{2^{10}} \\
&= \boxed{4.883\,\text{mV/LSB}} \\
\text{Repeatability} &= \max(\text{Max Deviation}) \\
&= \max(\langle 5.00, 0.00, 9.00, 44.00 \rangle) \\
&= \boxed{44.00\,\text{mV}} \\
\text{Accuracy} &= \max(\text{Deviation}) \\
&= \max\left(\begin{bmatrix} -0.009 & -0.009 & -0.009 & -0.009 & -0.009 & -0.009 & -0.009 & -0.009 & -0.004 & -0.009 \\ 0.001 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 \\ 0.016 & 0.021 & 0.012 & 0.016 & 0.012 & 0.012 & 0.016 & 0.016 & 0.012 & 0.016 \\ 0.02 & 0.024 & 0.02 & 0.024 & 0.01 & 0.02 & \mathbf{0.054} & 0.024 & 0.034 & 0.02 \end{bmatrix}\right) \\
&= \boxed{54.00\,\text{mV}} \\
\text{Manuf. Acc.} &= 2\,\text{LSB} \times \text{Resolution} \\
&= \boxed{9.766\,\text{mV}}
\end{aligned}$$

For significant figures, since the range is given to 3 decimal places, the resolution is given to 3 decimal places, more often than not, the number of significant figures is 4. This is because addition and subtraction do not take into account the number of significant figures but rather the number of decimal places.

Table B.1: Range, Resolution, Repeatability, Accuracy, and Manufacturer's Accuracy for Various Ranges of the Arduino Uno

Arduino Config.	Range (V)	Resolution (mV/LSB)	Repeatability (mV)	Acc. (mV)	Manuf. Acc. (mV)
5V Ref.	0.000 - 5.000	4.883	44.00	54.00	9.766
3.3V Ref.	0.000 - 3.300	3.223	4.000	17.00	6.445
3.3V Ref., 10x VDiv	0.00 - 33.00	32.23	32.00	83.00	64.45
3.3V Ref., [-10, 10]V	-10.00 - 10.00	19.53	0.000	24.00	39.06
3.3V Ref., 10x Amp.	0.000 - 0.330	0.3223	0.000	0.000	0.6445

C Appendix: Derivation of Resolution

The trade-off between resolution and range is that as the range increases, the resolution decreases. This is because the number of bits available to represent the range is fixed. For example, the Arduino Uno has 10 bits to represent the range of voltages. This means that the Arduino Uno can represent $2^{10} = 1024$ different voltages. If the range is 0 to 5V, then the resolution is 4.883mV/LSB. If the range is 0 to 10V, then the resolution is 9.766mV/LSB.

This can be shown mathematically. Then the resolution is given by:

$$\text{Resolution} = \frac{V_{\text{ru}} - V_{\text{rl}}}{2^n}$$

Let us assume the upper range V_{ru} and lower range V_{rl} are both multiplied by a factor of k.

$$\begin{aligned}\text{Resolution'} &= \frac{kV_{\text{ru}} - kV_{\text{rl}}}{2^n} \\ &= \frac{k(V_{\text{ru}} - V_{\text{rl}})}{2^n} \\ &= k\text{Resolution}\end{aligned}$$

D Appendix: Voltage Source Accuracy

The accuracy of the voltage source is 0.05%. A table deviation from the true value is shown below.

Table D.2: Deviation from True Value for Voltage Source with 0.05% Accuracy

Voltage (V)	Deviation from True Value (mV)
0.102	0.051
1.024	0.512
1.800	0.900
2.500	**1.250**

As one can see, the voltage source deviation is 10x smaller than the accuracy of the Arduino Uno. This means that the voltage source is a suitable reference voltage for the Arduino Uno.

Table D.3: Accuracy of Voltage Source Compared to Arduino Uno

Configuration	Accuracy (mV)
Voltage Source	1.250
5V Ref.	54.00
3.3V Ref.	17.00
3.3V Ref., 10x VD	83.00
3.3V Ref., [-10, 10]V	24.00
3.3V Ref., 10x Amp.	0.000

E Appendix: Circuit Component Accuracy

Below is a table of the additional error from the voltage divider, voltage scaler, and amplifier. The error is calculated by multiplying the voltage by 1%.

Table E.4: Additional Error from Voltage Divider, Voltage Scaler, and Amplifier

Voltage (V)	Error ($\pm$mV)
0.102	1.020
2.500	25.00

F Time-Varying Voltage Measurements

Below in Table F.5 are the nominal measurements for the time-varying voltage for the 10-bit and 5-bit ADCs.

Table F.5: Nominal Measurements of Time-Varying Voltage

	Mean Measurements of Time-Varying Voltage		
	Frequency (Hz)	Peak to Peak (V)	Mean Voltage (V)
10-bit	47.266	1.217	1.135
5-bit	47.372	1.2194	1.144
Oscilloscope	48.54	1.28	1.17

F.1 Mean Sample Calculation

The mean voltage for the 10-bit and 5-bit ADCs was calculated using Excel using the `=AVERAGE()` function across 5 cycles, which was approximately 60 data points.

F.2 Time-Varying Voltage Period Measurements

Below is a table of the time-varying period measurements for the 10-bit and 5-bit ADCs.

Table F.6: Time-Varying Period Measurements for 10-bit and 5-bit ADCs

	Period (μs)				
	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
10-bit	21000	21132	21132	21124	21400
5-bit	20912	21104	21136	21096	21304

Obtaining the nominal value (mean), standard deviation, and T-distribution inverse was done through Excel. The mean was calculated using the `=AVERAGE()` function across 5 cycles, which was approximately 60 data points. The standard deviation was calculated using the `=STDEV.S()` function. The T-distribution inverse was calculated using the `=T.INV()` function, where $\alpha = 0.05$ and $n = 5$. The results are shown in Table F.7.

Sample calculations for the **10-bit** ADC are shown. To calculate the random uncertainty, the

Table F.7: Time-Varying Period Uncertainty Measurements for 10-bit and 5-bit ADCs

	Nominal Value (μs)	STDEV (μs)	T-Inv	P_x ($\pm\mu$s)
10-bit	21158	146.66	2.7764	182.10
5-bit	21110	139.42	2.7764	173.11

following equation was used:

$$\begin{aligned} P_x &= t_{\alpha/2,n-1}\frac{\sigma}{\sqrt{n}} \\ &= 2.7764\frac{146.66\,\mu\text{s}}{\sqrt{5}} \\ &= \boxed{182.10\,\mu\text{s}} \end{aligned}$$

F.3 Time-Varying Voltage Frequency Calculations

Table F.8: Time-Varying Frequency Results for 10-bit and 5-bit ADCs

	Frequency (Hz) Nominal Value	(±Hz) Uncertainty
10-bit	47.264	0.40680
5-bit	47.370	0.38844

The equation relating frequency and period is shown below. Sample calculation for the 10-bit ADC is shown below.

$$\begin{aligned} f &= \frac{1}{T} \\ &= \frac{1}{21158\,\mu\text{s}} \\ &= \boxed{47.264\,\text{Hz}} \end{aligned}$$

Table F.9: Time-Varying Peak and Trough Voltage Measurements for 10-bit and 5-bit ADCs

	Peak and Trough Voltage (V)				
	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
10-bit Trough	1.212	1.234	1.234	1.206	1.199
5-bit Trough	1.225	1.237	1.228	1.192	1.215
10-bit Peak	1.763	1.772	1.763	1.743	1.766
5-bit Peak	1.76	1.772	1.766	1.747	1.76

Utilizing error propagation, the random uncertainty in frequency is:

$$\begin{aligned} P_{x'} &= \sqrt{\left(\frac{\partial f}{\partial T} P_x\right)^2} \\ &= \left|\frac{\partial f}{\partial T} P_x\right| \\ &= \left|-\frac{1}{T^2} P_x\right| \\ &= \frac{P_x}{T^2} \\ &= \frac{182.10\,\mu\text{s}}{21158\,\mu\text{s}^2} \\ &= \boxed{0.40680\,\text{Hz}} \end{aligned}$$

F.4 Time-Varying Voltage Peak to Peak Voltage Calculations

Below are the resulting peaks and troughs for the 10-bit and 5-bit ADCs for 5 cycles.
The nominal values for troughs and peaks were calculated using the `=AVERAGE()` function in Excel, the standard deviation was calculated using the `=STDEV.S()` function, and the T-distribution inverse was calculated using the `=T.INV()` function, where $\alpha = 0.05$ and $n = 5$. The systematic uncertainty was taken to be the accuracy of the 3.3V reference voltage for 10-bit ADC. and half the resolution for 5-bit ADC.

Table F.10: Time-Varying Peak and Trough Voltage Uncertainty Measurements for 10-bit and 5-bit ADCs

	Nominal Value (V)	STDEV (V)	T-Inv	P_x ($\pm$V)	B_x ($\pm$V)	U_x ($\pm$V)
10-bit Trough	0.544	0.01111	2.7764	0.01380	0.01700	0.0219
5-bit Trough	0.542	0.008307	2.7764	0.01031	0.05156	0.0526
10-bit Peak	1.761	0.009274	2.7764	0.01151	0.01700	0.0205
5-bit Peak	1.761	0.01092	2.7764	0.01356	0.05156	0.0533

The total uncertainty, U_x, was calculated using the root sum square (RSS) method. An example calculation for the 10-bit trough is shown below:

$$\begin{aligned} U_x &= \sqrt{P_x^2 + B_x^2} \\ &= \sqrt{0.01380^2 + 0.01700^2} \\ &= \boxed{0.0219\,\mathrm{V}} \end{aligned}$$

Obtaining peak to peak voltage was done by subtracting the trough from the peak. The results are shown in Table F.11. A sample calculation for 10-bit ADC and equation for peak to peak voltage is shown below:

$$\begin{aligned} V_{\text{p-p}} &= V_{\text{ru}} - V_{\text{rl}} \\ &= 1.761\,\mathrm{V} - 0.544\,\mathrm{V} \\ &= \boxed{1.217\,\mathrm{V}} \end{aligned}$$

Table F.11: Time-Varying Peak to Peak Voltage Results for 10-bit and 5-bit ADCs

	Nominal Value (V)	U_x ($\pm$V)
10-bit	1.217	0.0381
5-bit	1.219	0.0981

Error propagation was used to calculate the uncertainty in peak to peak voltage. The equation for peak to peak voltage is shown below with sample calculations for the 10-bit ADC.

$$
\begin{aligned}
U_{x'} &= \sqrt{\left(\frac{\partial V_{\text{p-p}}}{\partial V_{\text{ru}}} U_{x,ru}\right)^2 + \left(\frac{\partial V_{\text{p-p}}}{\partial V_{\text{rl}}} U_{x,rl}\right)^2} \\
&= \sqrt{1^2 0.0219^2 + (-1)^2 0.0205^2} \\
&= \boxed{0.0381\,\text{V}}
\end{aligned}
$$