ELEC-4450: Power Electronics

Instructor: Dr. Caniggia Viana

Switched-Mode Power Supply: Project Report

Manveer Aujla	110072428
Alice Seo	110072503

17 April 2025

Buck Converter PCB

The original, unsoldered PCB from the manufacturer is shown in Figure 1. The PCB with all components soldered on is shown in Figure 2.

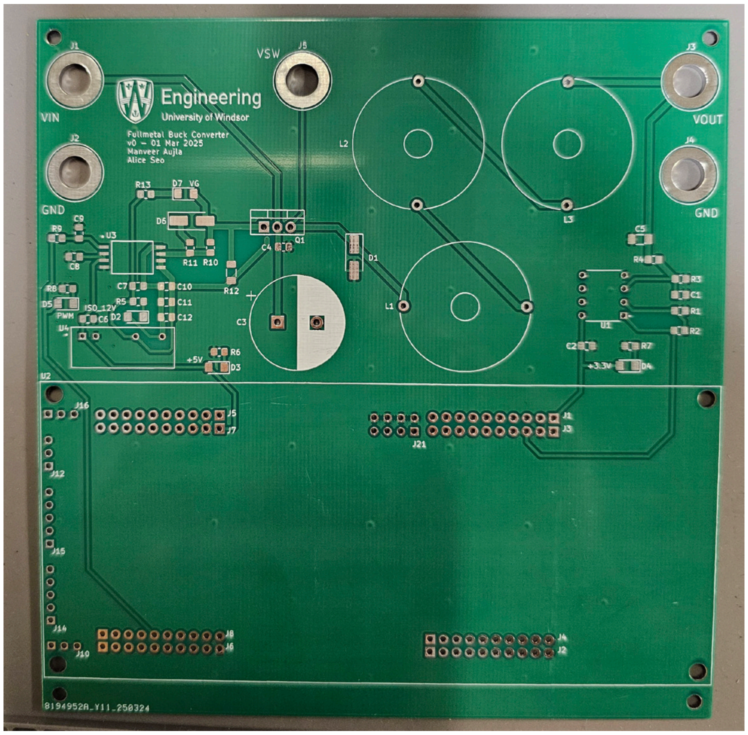

Figure 1: Unsoldered buck converter PCB.

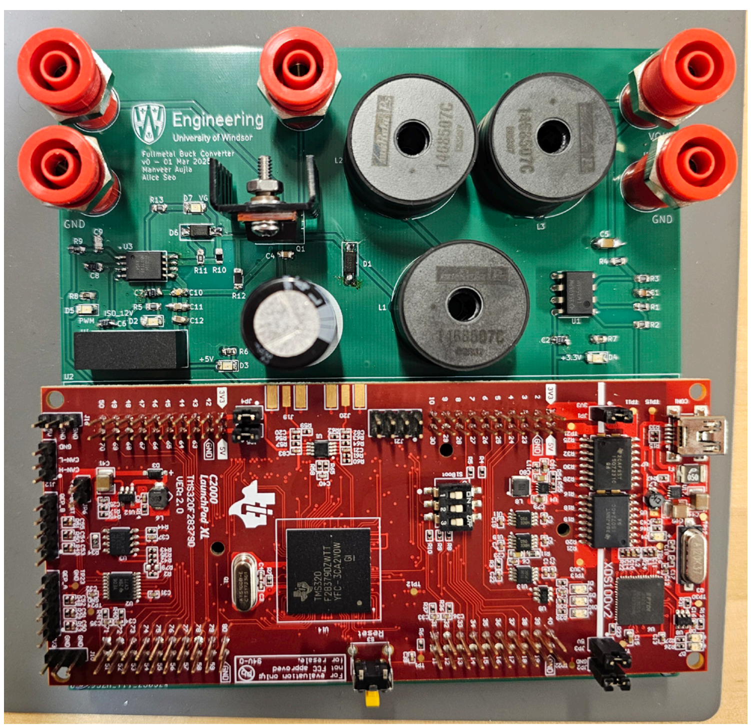

Figure 2: Soldered buck converter PCB.

The converter connected to a laptop and the Festo machine, showing the appropriate LEDs lit, is shown in Figure 3.

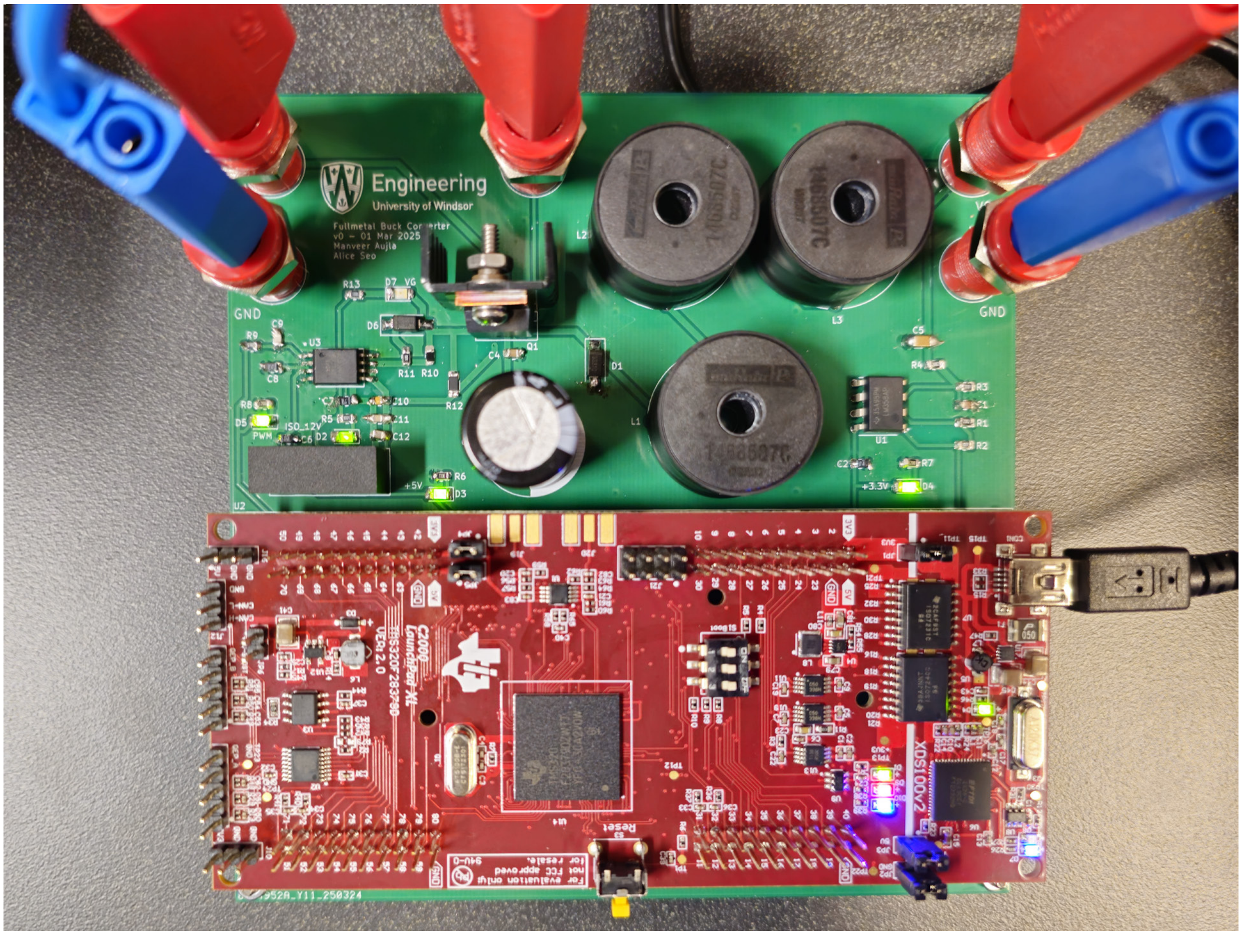

Figure 3: Converter connected to the laptop and Festo machine.

Design Changes

Since the Preliminary Design, only a few changes were made to the PCB design:

- Trace widths were increased to a minimum of 0.5 mm
- Surface-mounted LM358 op amp was replaced with a through-hole variant
- Mounting holes were added to the PCB layout
- GA-recommended MCU headers and standoffs were added to the BOM

Changes to the buck converter PCB that were made when debugging and testing are discussed thoroughly in the subsequent sections.

PCB Bring Up: Assembly

In the Preliminary Design, we formulated a PCB bring-up plan. This section describes the assembly and soldering process.

Out-of-Box Visual Inspection

All components ordered from Digikey were accounted for and the PCB from JLCPCB had no visual issues.

Build & Solder the Board

We have relatively little soldering experience, so soldering all of the components onto the board, particularly the surface-mounted gate driver IC, was a challenging but rewarding learning experience. We made some mistakes, such as soldering the Schottky diode in the wrong direction. However, we were able to remedy all mistakes without issue. With each component we soldered onto the board, we used a multimeter to check the continuity between adjacent components/pads to ensure that it matched the layout on KiCad.

Post-Solder Inspection

We made sure that all components were secured to the board as expected, mechanically and electrically. Using a multimeter, we performed a final sanity check on the board to ensure all connections were as expected.

PCB Bring Up: Testing & Debugging

This section describes the testing and debugging process of our PCB.

IC Power and Sanity Test

The MCU was supplied with $V_{\text{CC}} = 5\ \text{V}$ from the Micro-USB connection to a laptop. We noted that the +5V and +3.3V LEDs were on as expected. The ISO_12 LED, although on, was dimmer than the other LEDs. We had chosen resistors in series with each LED to provide roughly 10 mA of current for each specific voltage; since the LED was dim, it meant the output voltage from the isolated power supply was less than expected. The isolated power supply should take a 5 V input from the MCU and boost it to 12 V references to its isolated ground so that the gate driver can route it to bias the high-side switched MOSFET. We measured the output of the isolated power supply $V_{\text{ISO12}} = 3.5\ \text{V}$ (measuring ISO_12 with respect to ISO_GND). Apparently, the isolated power supply was not doing its job.

We measured the resistance between ISO_12 to ISO_GND to be about $R_{\text{ISO-out}} = 3\ \text{k}\Omega$, meaning there was not a massive load (e.g., a DC short) demanding tons of current at the output side of the power supply. After investigation, our theory was that the isolated power supply IC must be defective. To fix this, we borrowed an IC from another group and desoldered the old, seemingly dysfunctional, IC. However, after desoldering the 4-pin IC (using three pairs of hands), solder remained on the through holes. It was too difficult to solder on the new IC because we would need to heat each pin simultaneously and push it into the PCB. So, we engineered a solution: using some jumper wires, electrical tape, and alligator clips, we attached the new isolated power supply to the board. To our dismay, repeating the IC power test showed ISO_12 was still measuring about 3.5 V. Thus, our hypothesis about the IC being broken was likely incorrect.

Next, we removed the isolated power supply from the PCB and applied an external 9-V battery at the ISO_12 and ISO_GND terminals of the PCB. To our surprise, this still did not work, and the voltage somehow became limited to 3.5 V or so. Naively, we put another battery in series and tried again—causing the board to smoke. We measured $R_{\text{ISO-out}} = 20\ \Omega$ which indicated that one of the V_{DD} capacitors had burned. Checking our BOM, the tantalum capacitor C10 has a voltage rating of 16 V. After removing C10, we checked the resistance $R_{\text{ISO-out}}$ once more and it returned to $3\ \text{k}\Omega$. Therefore, we correctly identified that C10 was burned and consequently internally shorted. With C10 removed, the isolated power supply now outputted 12 V. We later learned from the datasheet that C10 has a polarity, and our theory is that we had connected C10 with the wrong polarity which caused the isolated power supply to be unable to provide 12 V.

Full Power Test

The MCU was programmed using Simulink via the C2000 Microcontroller Blockset using the Formula SAE MATLAB license. The Simulink model to program the firmware is shown in Figure 4. The PID block contains the controller $C(s)$ made in the Preliminary Design.

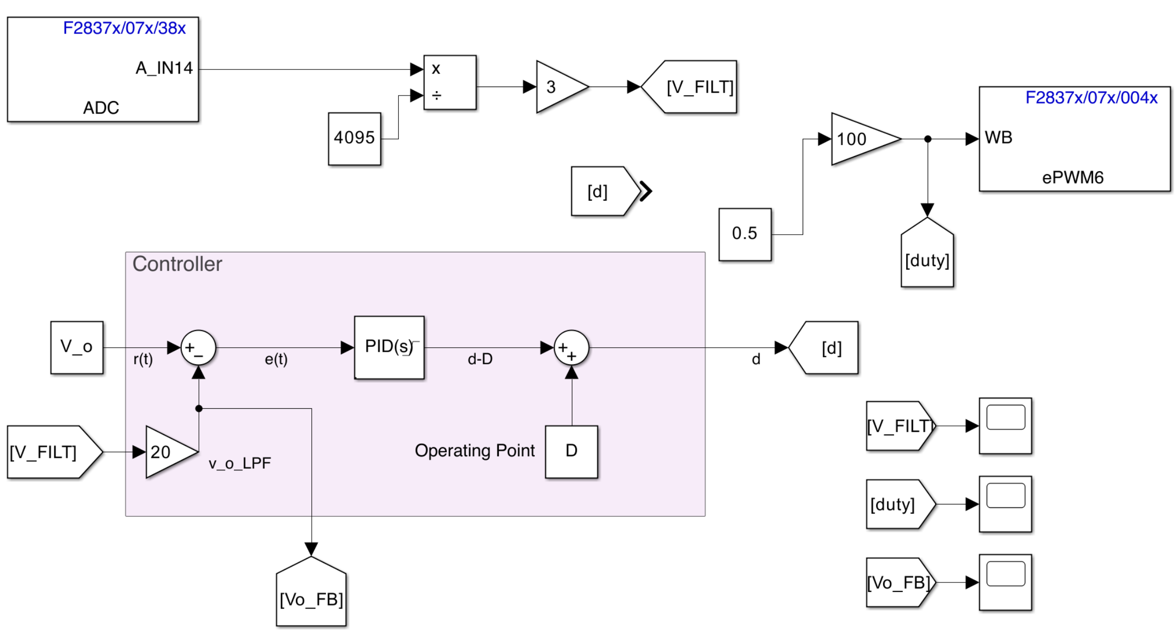

Figure 4: Simulink model using C2000 Microcontroller Blockset to build firmware.

The test setup using the Festo machines is shown in Figure 5.

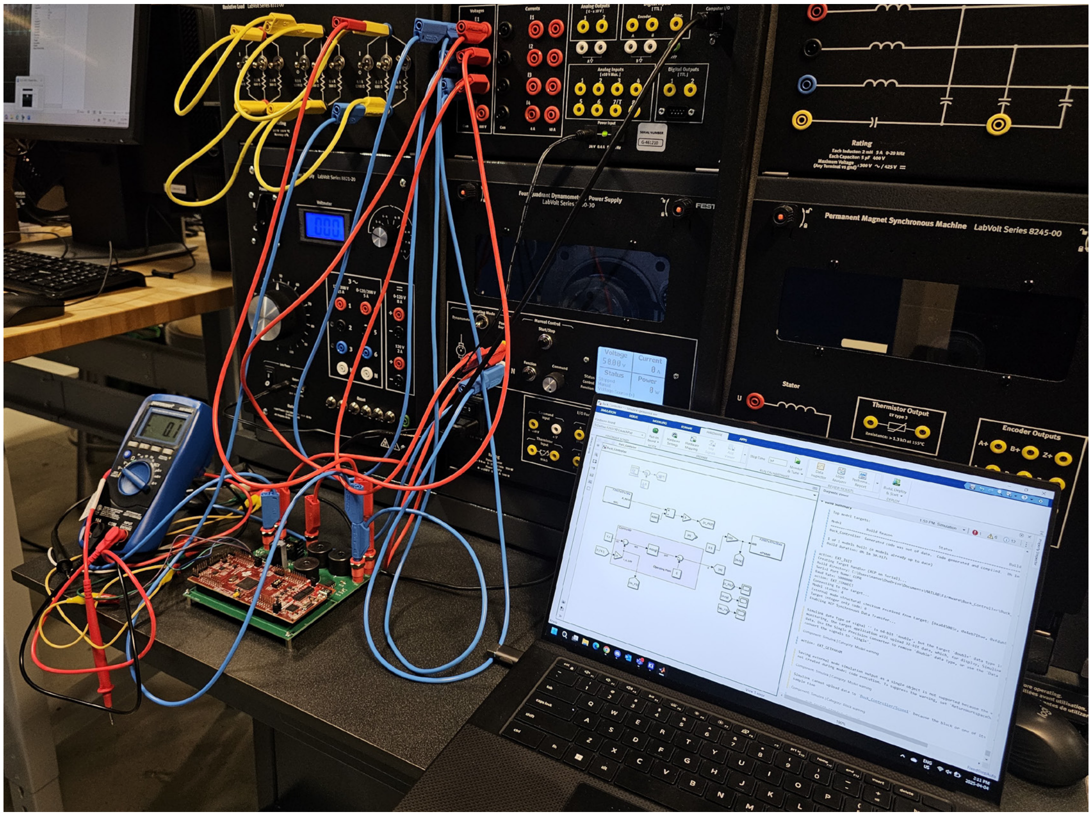

Figure 5: Test setup.

An input voltage of $V_i = 10$ V was applied to VIN from the Festo machine to the converter. The MCU was programmed for a voltage reference of $V_o^* = 5$ V. Therefore, we would expect the duty cycle $d(t)$ would adjust according to the controller such that the output voltage V_o tracks the reference. Unfortunately, this did *not* work; when the input voltage was switched on, the output voltage rose to nearly the input voltage before rapidly decaying to about 0–0.4 V and the duty cycle converged to a value of about 3.5%.

To debug, we needed to figure out whether this was a hardware or software problem. The MCU was programmed to manually set the duty cycle to 50%. In this attempt, with the input voltage of 10 V, the average output voltage went to 4.46 V, which roughly follows the conversion ratio for a buck converter. These results are shown in the next section. We then probed the hardware to compare to the ADC measurements and found that the ADC was measuring the correct voltage. However, we noted that the LPF was not behaving as expected. Specifically, we noted that the V_+ and V_- terminals of the op amp measured a voltage difference of about 2 V or so. For an ideal op amp, $V_+ = V_-$ and so we would expect this voltage difference to be nearly zero!

Our current hypothesis is the LPF stage hardware is compromised such that the ADC is getting a voltage reading larger than the actual normalized output voltage. In other words, since the ADC voltage determines the next iteration's duty cycle, $d(t)$ converging to 3.5% (when we expect it should be at least greater than 50%) indicates that the controller is receiving information that we are *at* the reference voltage when $d(t) = 3.5\%$, despite this being false. Problems notwithstanding, we are happy that our PCB was able to, at the very least, operate as a buck converter for a given duty cycle. For true controlled buck operation, we will investigate and fix the LPF stage to achieve reference tracking for load transients.

Experimental Results

As mentioned in the previous section, our buck converter could only operate for a given duty cycle hardcoded in the firmware. That is, we were unable to use feedback control and track a reference due to hardware issues with the LPF stage. Waveforms for $V_i = 5$ V and $d(t) = 100\%$ are shown in Figure 6.

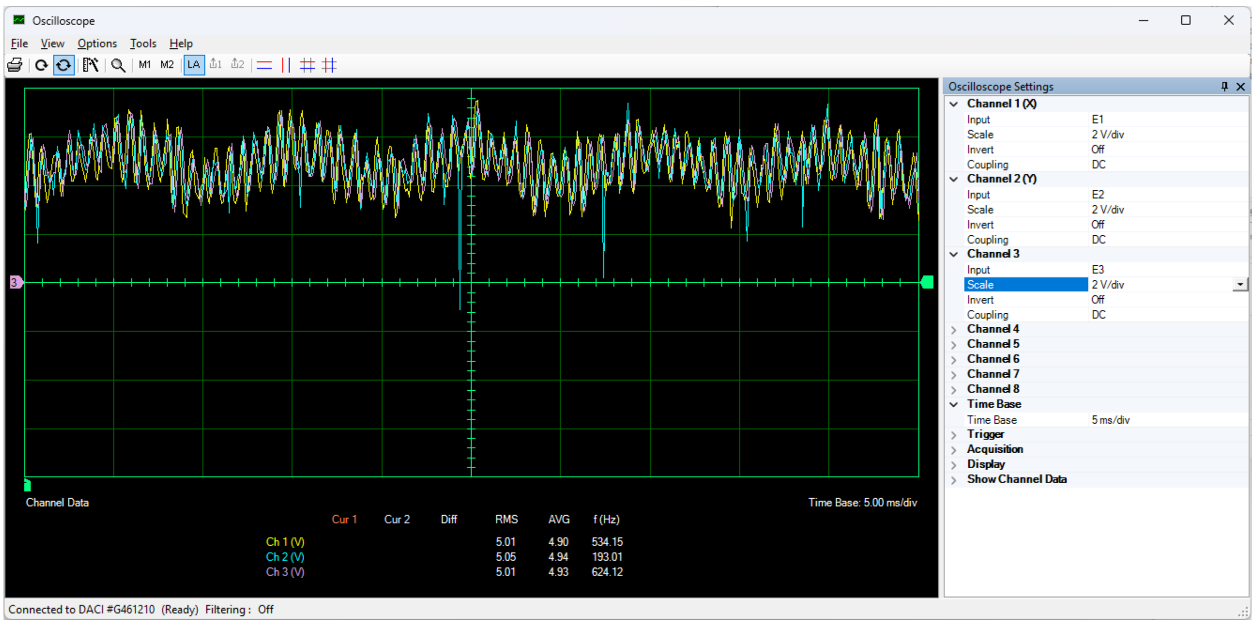

Figure 6: Waveforms of V_o (Ch1), V_{sw} (Ch2), and V_i (Ch3) for $V_i = 5$ V and $d(t) = 100\%$.

The input voltage essentially passes to the output, as expected with a duty cycle of 100%. That is, $V_i \approx (100\%)V_o$. Waveforms for $V_i = 10$ V and $d(t) = 50\%$ are shown in Figure 7.

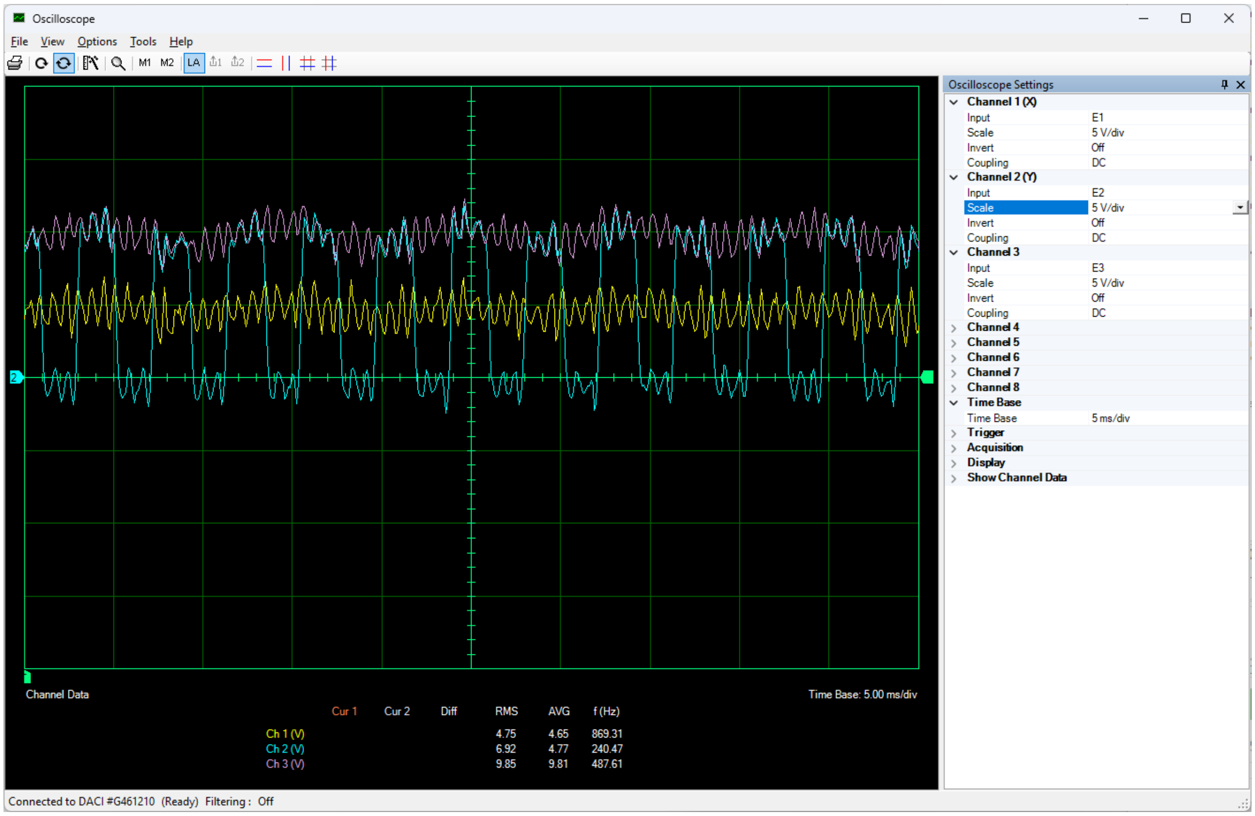

Figure 7: Waveforms of V_o (Ch1), V_{sw} (Ch2), and V_i (Ch3) for $V_i = 10$ V and $d(t) = 50\%$.

For a duty cycle of 50%, an ideal buck would output $V_o = 50\% V_i = 5$ V. However, we have losses that are unaccounted for; this is the purpose of the controller with feedback. Despite us not designing for the losses, the controller with feedback allows us to track references despite not using a robust model for the plant (i.e., the buck converter). Since we did not achieve feedback control, the best we could do was manually set a duty cycle. We can see that the buck converter is behaving as expected for its conversion ratio.

Individual Contribution

Manveer Aujla

My main contributions to the project include designing the PCB, designing and simulating the controller, selecting components based on the design constraints and making the BOM, soldering PCB components, debugging and reworking the PCB to make it functional as a buck converter, programming the firmware on the MCU, writing the bulk of the Preliminary Design, writing the ethical implications essay, and writing the final report.

Alice Seo

My contribution to the project includes checking over the PCB layout, components, and BOM, measuring the spacings between the components, soldering some PCB components and continuity testing, and contributing to writing the Preliminary Design and the ethical implications essay.