Monterey Bay Aquarium Research Institute

An Inertial Current Meter Data Logger for Drift Mitigation Aboard a Coastal Profiling Float

Jonathan Siskind, Stanford University

Mentors: Gene Massion

Summer 2024

Keywords: coastal profiling float, IMU, accelerometer, ocean current

ABSTRACT

Since the early 2000s, the Argo network, consisting of thousands of robotic profiling floats, has monitored the world's oceans. However, these floats face unique challenges near the coasts, as strong ocean currents tend to cause them to drift either onto the beach or too far away from the coastline. Since ocean currents vary in magnitude and direction as the float's depth changes, a proposed solution to this challenge is to program each float to autonomously and independently find ocean currents at specific depths to carry it back to its intended location. To achieve this, floats must have the capability to measure ocean currents. This project consisted of developing a prototype ocean current measuring system using an IMU (inertial measurement unit). This was packaged into a custom data logger device that was designed to mount on to the side of a coastal profiling float. This paper outlines the software, mechanical, and electrical design of the data logger, and describes how its accuracy was successfully validated by a series of tests. Future steps for the integration of a complete drift-mitigation system for coastal profiling floats using this data logger are discussed at the end.

INTRODUCTION

The Argo float network is the primary data collection system that climate, geophysics, and oceanographic scientists have used to measure the world's changing oceans over the past 20 years. The network consists of 3800 free-floating robotic systems scattered across the oceans, each of which contain an array of sensors that measure various chemical and biological markers for ocean health. These sensors are modular, but common data points include temperature, salinity, chlorophyll, backscatter, oxygen, nitrate, and pH (Scripps Institute of Oceanography 2023). Most floats operate on an approximately 10-day cycle. They spend most of that time hovering at a park depth of 1000m. At the 10-day mark, they sink to 2000m before floating up to the surface in a "profile". During these upwards profiles, the onboard sensor suite continuously logs measurements until the float reaches the surface. After surfacing, the floats then relay this telemetry data to scientists via satellite before sinking back down to 1000m. Most floats have enough battery life to repeat these profiling cycles about 200 times before they must be recovered or abandoned.

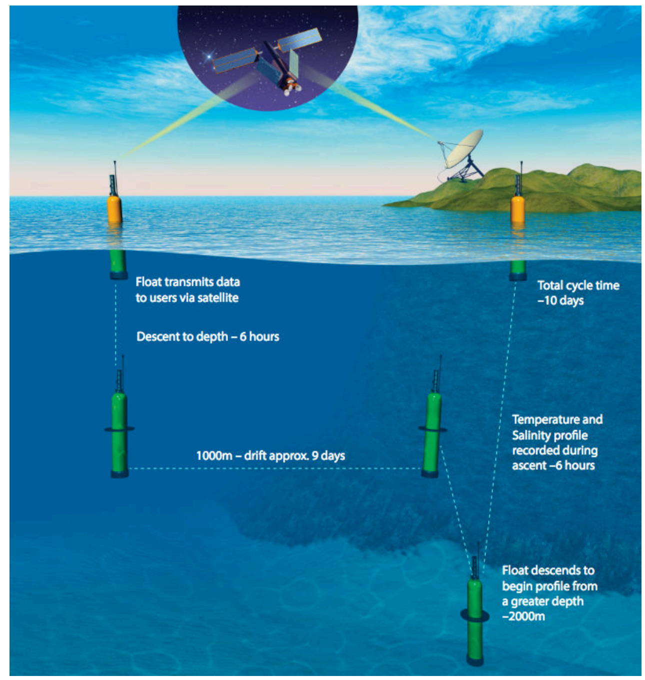

Figure 1. The standard operating cycle of an Argo float. *Scripps Institute of Oceanography.*

These floats control their depth with an active buoyancy-management system. Within each float, there is a hydraulic pump that can fill the external bladder with oil from the internal reservoir, causing the float to expand in volume and therefore become more buoyant. When the float needs to sink, it pumps oil from the bladder back into the internal reservoir, causing the float to shrink in volume and become less buoyant.

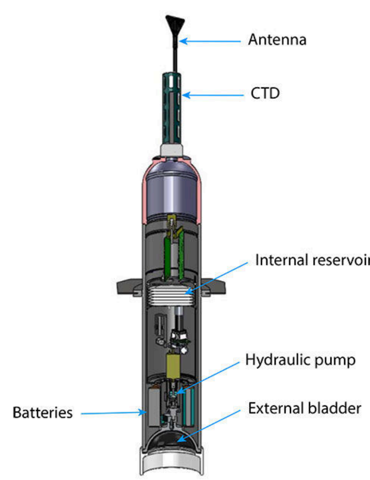

Figure 2. Mechanical representation of an Argo float. *Scripps Institute of Oceanography.*

However, floats do not have any motors, pumps, or thrusters to control lateral movement, making their position vulnerable to currents and waves. This challenge is why floats do not work particularly well around the coasts, as currents and waves will either wash them ashore or push them too far from the coastline.

Data from coastal regions is important to analyze, as the low-level chemical and biological factors near shore affect the abundant marine life up the food chain that also live near the coasts. Thus, engineers and scientists at MBARI have long been interested in developing a unique float design that is optimized to function in coastal environments. Engineers from MBARI's chemical sensors lab have designed a coastal profiling float (CPF) that parks at the seafloor in between profiles instead of hovering at 1000m. This strategy mitigates a substantial portion of the drifting. However, these CPFs are still vulnerable to drifting while profiling upwards and while relaying their sensor data to satellites at the surface.

In order to solve this problem, a secondary drift-mitigation strategy was devised. This strategy relies upon the fact that ocean currents are not constant throughout a water column – if there is a strong surface current going one direction, there is frequently an opposing current somewhere beneath it. This phenomenon, caused by the Coriolis Effect, is called an Ekman Spiral. (University of Hawaii 2024). If a float is able to find the depth of this current, it can park at that depth by changing its buoyancy and "ride" the deeper current back to its original position.

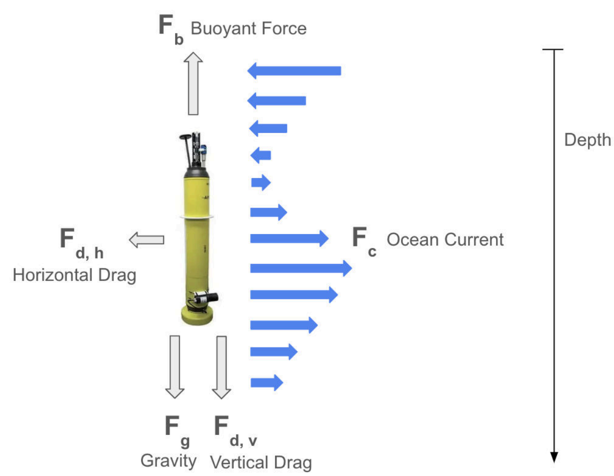

Figure 3. Free body diagram of a profiling float. The blue arrows represent the changing current throughout a water column.

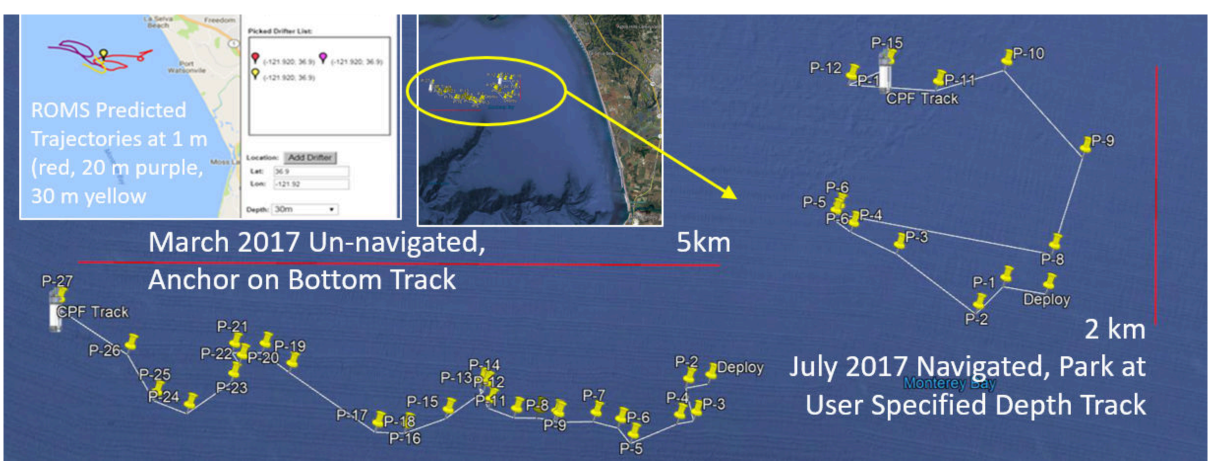

Figure 4. The left float was subject to only ocean drift, and the right float minimized drift by periodically changing depths according to ROMS.

Chemical Sensors Lab, MBARI

This strategy was tested aboard two floats, the results of which are seen in Figure 4. The float on the left did not exercise this strategy, and it drifted 5 km in three days. The float on the right was programmed manually in real-time to change depths based on the ROMS current model (Arango and Shchepetkin n.d.), and it only drifted 2 km in that same timespan.

However, for a system that is both more accurate and does not require manual programming of each float at all times, this system needs to be autonomous. Each float must be able to find depths with desired currents independently without human intervention. The first step in this solution is developing a measurement device aboard each float that can constantly measure the ocean currents that each float is experiencing. Since ocean currents induce small accelerations on floats as they drift, an IMU (inertial measurement unit) can collect those acceleration values during each upward profile so that the float may decide what depth to park at.

DESIGN

Since the CPF design is already so mechanically and electrically complex, an external, independent, IMU-driven data logging device was constructed for this project instead of integrating a logging system directly into the CPF. I will split the engineering design of this external data logger into three different parts: software, mechanical and electrical.

SOFTWARE

There are two primary devices that compose the core functionality of this data logger: a VectorNav VN-100 IMU Dev Board and an STM32H7A3 Nucleo Board with an SD card shield. The former is the accelerometer that was tested and validated by several years of previous interns. The latter is a small, low-power microcontroller board that runs the code that will be discussed below.

The primary software challenge of this data logger comes from the fact that the code onboard the Nucleo Board needs to be executing two different tasks simultaneously – it must constantly read incoming acceleration data from the VN-100 and write that data onto an SD card for future analysis. Unfortunately, the constant collection of new acceleration data is a blocking operation, so in normal circumstances, the code will miss a window of incoming acceleration data every time that it saves a chunk of previous data to the SD card. This is the gap seen in **Process A** at $\mathbf{t_1}$ on the left of Figure 5.

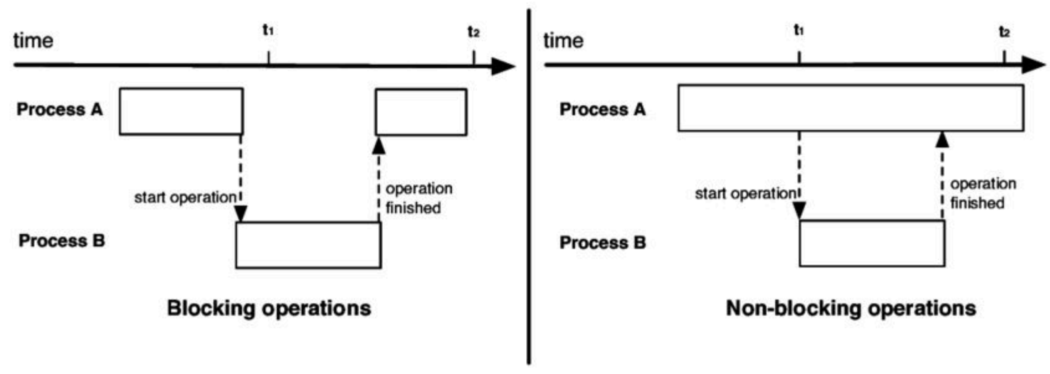

Figure 5. Visual representation of blocking versus nonblocking operations.

Thomas Hartmann, University of Luxembourg

Furthermore, multithreading was deemed unrealistic for this application. Thus, a clever workaround was devised to execute both of these commands simultaneously in a non-blocking manner: interrupt-driven double buffering. This consists of enacting two storage arrays; we can call them Buffer 1 and Buffer 2. Each buffer is a certain length; in this case, it is 512 bytes.

First, Buffer 1 is the designated "read buffer". All the acceleration data incoming from the IMU is saved into Buffer 1.

Once Buffer 1 is full, an interrupt is triggered. This interrupt causes Buffer 2 to become the designated read buffer, it continues reading new data from the IMU where Buffer 1 left off. Meanwhile, Buffer 1 is now the "write buffer" and uses the FatFS protocol to write all 512 bytes of data within it to the SD card.

Once Buffer 2 is full, the interrupt is triggered again, so the two buffers switch. This process continues for as long as the microcontroller is powered. Figure 6 below illustrates this process.

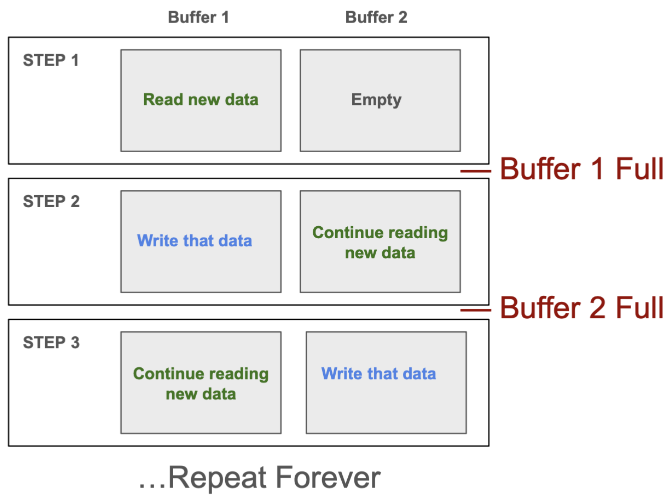

Figure 6. Visual representation of interrupt-driven double buffering procedure for UART data transmission.

The result of this process is that there is never a gap in streamed IMU data. It is always being read into either Buffer 1 or Buffer 2 and is never passed over. There are a handful of considerations to take into account when using this system.

First, the buffer size must be large enough so that all of the data in the "write buffer" is completely saved to the SD card before the two buffers switch. However, a buffer size that is too large will result in the opposite problem; the read buffer might overrun and fill to capacity before the write buffer has completed its write operation. A buffer size of 512 bytes was picked as the "sweet spot" after testing a handful of different sizes.

Secondly, the user has the capability to select the sampling rate of the IMU anywhere from 1 to 800 Hz. For the purposes of determining optimal ocean currents at depths as specific as possible, a higher sampling rate is best. However, a sampling rate that is too high could potentially also cause buffer overrun issues as described above. Thus, a sampling rate of 50 Hz was picked as the "sweet spot" after testing various different sampling rates.

Furthermore, the electrical system was designed without a switch (which was a mistake in hindsight). If a switch was included, the program would call f_sync from FatFS exactly once to finally flush all the written data onto the hardware of the SD card right before ending. However, since the system without a switch could be powered off at any time without notice, the program needs to execute f_sync periodically to ensure that the SD card will retain the data. Unfortunately, f_sync is an expensive command that detracts from the time efficiency of the double-buffering operation explained above. To compromise, f_sync is called once every ten cycles of buffer switching.

MECHANICAL

The data logger was designed to fit inside of a general-purpose MBARI pressure housing tube. After a handful of iterations, a compact structure for mounting all the necessary electronic components was designed in SolidWorks and 3D printed in PLA.

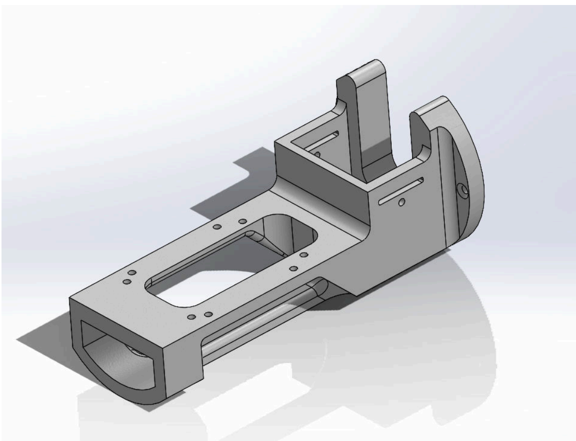

Figure 7. Solidworks model of the mechanical enclosure.

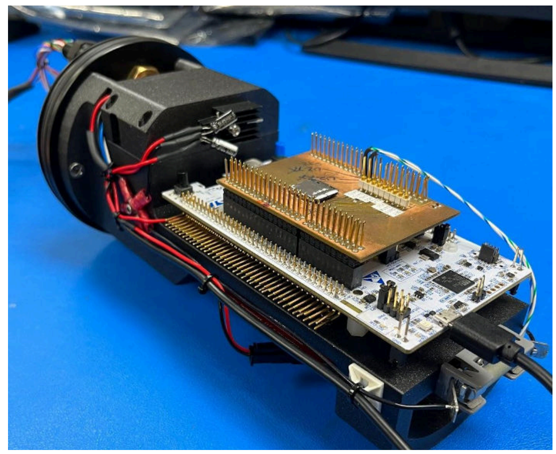

Figure 8. The fully assembled data logger.

ELECTRICAL

As described previously, the two major hardware components of this data logger are the VN-100 IMU and the STM32H7A3 microcontroller Nucleo Board.

First, a battery was needed to power both of these devices. Several factors were weighed in the selection of a battery, including battery chemistry (between Li-Ion and LiFePO4), number of cells, and battery capacity. A high-capacity 10.5Ah 2-cell (7.4V) Li-Ion battery with UN 38.3 certification was selected. Li-Ion chemistry was chosen because of its higher energy density, high capacity was chosen to maximize time for testing, and 7.4V was chosen to minimize the reduction needed to bring the voltage down to the required amount for the IMU and Nucleo Board (5V). To do this reduction from

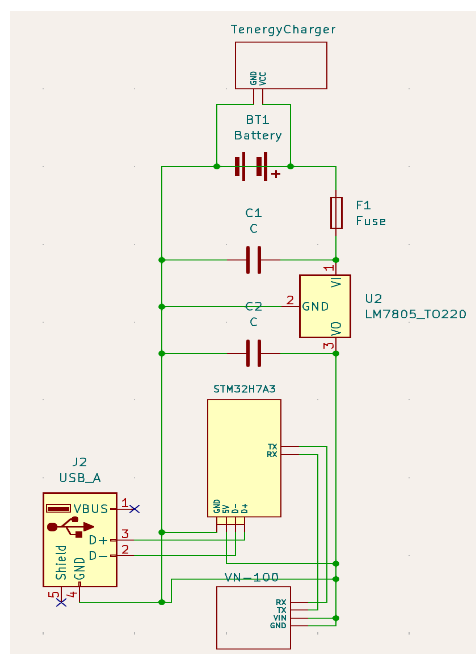

Figure 9. Electrical schematic of the data logger.

7.4V to 5V, a 7805 linear regulator was used. It was attached to a heatsink to prevent overheating, and capacitors were added to stabilize the 5V output that powered both the IMU (via the Vin pin) and the Nucleo Board (via USB through CN1). A 2A replaceable fuse was added for overload protection.

In addition, it was deemed helpful to be able to charge the battery and load new code to the Nucleo Board while the entire system was enclosed. This capability was implemented by creating a wire harness with a USB connector and battery-charging wires through the 10-conductor Subconn MacArtney cable that penetrates the front of the pressure housing's bulkhead.

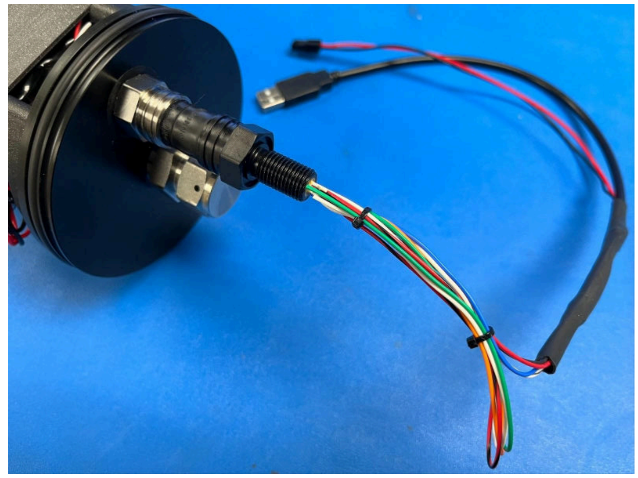

Figure 10. External wire harness for programming and battery charging.

TESTING METHODS AND RESULTS

First, a rudimentary bench test was conducted to assess the functionality of the data logger. The data logger was placed upright on a tabletop, and was quickly moved left and right with roughly 5 second intervals in between. The results for this test are seen in Figure 11.

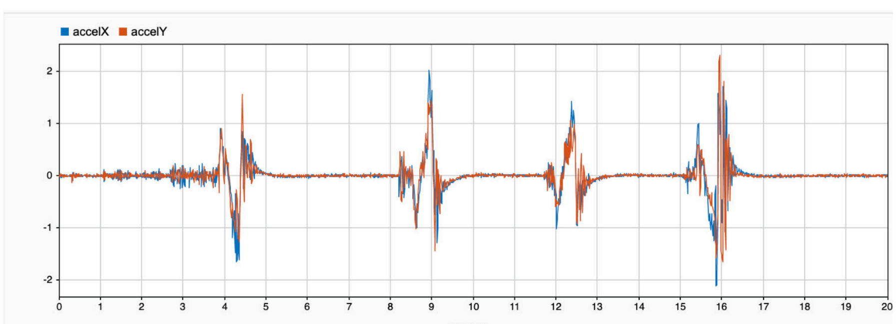

Figure 11. Unfiltered bench acceleration test signal.

The resulting MATLAB plot demonstrates that the data logger does indeed function properly, as the spikes in acceleration data coincide with the movements recorded during the bench test. A lowpass filter designed in MATLAB was applied to this data to smooth the signal and eliminate most of the noise. The result is seen in Figure 12.

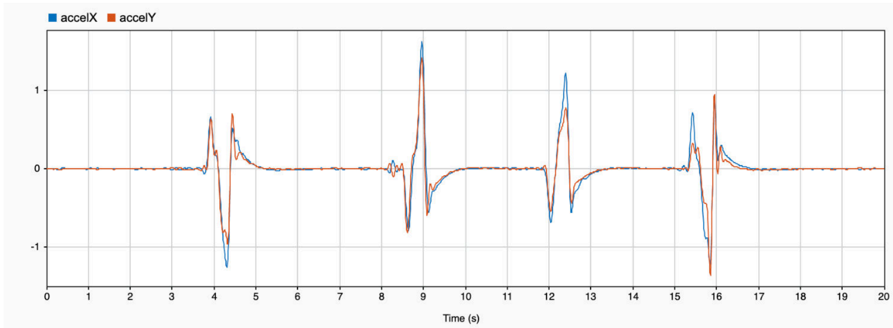

Figure 12. Low pass filter applied to bench acceleration test.

Next, a test was conducted to validate the data logger's accuracy against a real world physical phenomenon. It is known that the tangential acceleration curve of a swinging pendulum is a sinusoid, so the data logger was attached to a large pendulum to determine if its logged function matched the theoretically calculated sinusoid.

In this experiment, the period of the acceleration data's sinusoid collected by the data logger was shown to have a period nearly identical to that of the calculated period from physical analysis. Thus, this experiment verifies that the acceleration data collected from the logger is trustworthy, at least to a certain extent.

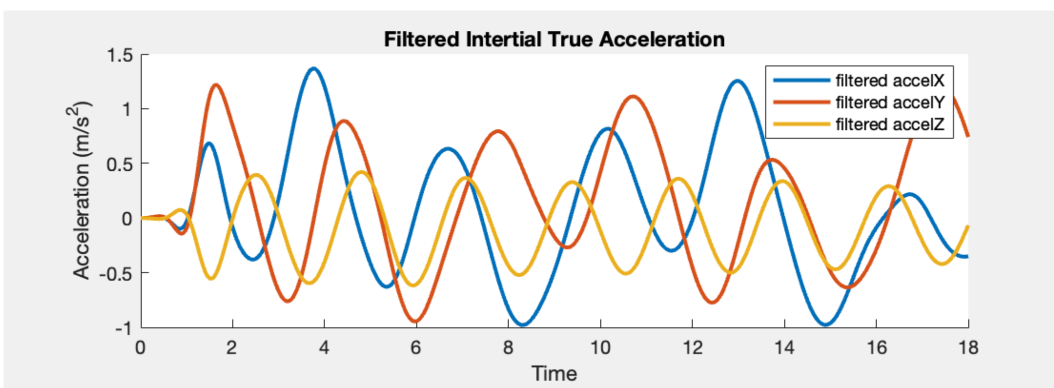

Figure 13. Filtered accelerations in 3 axes for the data logger attached to a swinging pendulum.

DISCUSSION

The limited time of the 10 week internship elapsed before this data logger could be tested further. However, the tests described above verify the data logger's potential to be used in the test tank and eventually the ocean.

If future contributors to this project were to continue developing the hardware aboard the data logger, my main suggestion would be to implement a switch that is accessible from outside the pressure housing. This way, the logger could be turned on and off without needing to remove it from the pressure housing (which is a time-consuming process).

The immediate next steps for the development of the drift-mitigation portion of the CPF involve more extensive and rigorous testing using the data logger. First, I would advise developing a tank test to assess the logger's functionality in water. The logger should be entirely sealed within a pressure housing, and the watertight seals should be validated with a helium leak test before inserting the logger in MBARI's test tank.

Thereafter, the data logger should be mounted to the side of the CPF and then deposited into the ocean. A test can be conducted to compare periodic before-and-after GPS coordinates of the float with the acceleration plots collected by the data logger. For example, if GPS indicates that the float drifted 1km south over the course of one day, the data collected by the logger should primarily feature accelerations and velocities that create net southwards movement.

Once this test has been completed, the long term goal is to incorporate the mechanical, electrical, and software elements of the data logger entirely within the infrastructure of the CPF. Then, adjust the CPF's depth-control program to use the data collected by the logger in order to mitigate drift.

CONCLUSIONS

The ocean current data logger shown in this paper is functional, reliable, accurate, and robust, and will prove to be a valuable tool in the development of MBARI's coastal profiling floats. Continued iterative and rigorous testing of this data logger will result in a system that enables floats near the coasts to mitigate lateral drift from ocean currents.

ACKNOWLEDGEMENTS

I have many thanks to give for my wonderful summer experience at MBARI. First of all, my mentor, Gene Massion, imparted immense practical engineering knowledge to me over the course of my internship; because of him, I learned more engineering during those ten weeks than any other time in my life. Furthermore, the electronics technicians at MBARI – James McClure, Cris Beebe, Jim Montgomery, and Jose Rosal each kindly lended me their expertise during the construction and assembly of the data logger. I also must thank the members of the Chemical Sensors Lab for their kindness during my ten weeks working in their lab. The 2024 class of MBARI interns also provided me with great company during my summer; in particular, I owe thanks to Forrest Milner for technical assistance on certain elements of my project. Last but not least, I owe thanks to my research mentor at Stanford, Dr. Oussama Khatib, for writing the recommendation letter that earned me this fantastic opportunity at MBARI.

References:

Arango, Hernan, and Alexander Shchepetkin. n.d. "Regional Ocean Modelling System (ROMS)." https://www.myroms.org/.

"Argo Float Program." n.d. Argo. University of San Diego, Scripps Institute of Oceanography. https://argo.ucsd.edu/.

Coetsee, Renee. n.d. "Signal Processing Onramp | Self-Paced Online Courses - MATLAB & Simulink." Matlabacademy.mathworks.com. https://matlabacademy.mathworks.com/details/signal-processing-onramp/signalprocessing.

Hartmann, Thomas. 2016. "Enabling Model-Driven Live Analytics for Cyber-Physical Systems: The Case of Smart Grids," August 11, 2016. https://orbilu.uni.lu/bitstream/10993/28924/2/phdthesis-final-21112016-compressed.pdf

Massion, Gene, Ken Johnson, and Brent Jones. 2020. "New Coastal Profiling Floats for Diagnosing Ocean Health." MBARI. MBARI. February 6, 2020. https://www.mbari.org/news/new-coastal-profiling-floats-for-diagnosing-ocean-health/.

"Ocean Surface Currents." 2024. University of Hawaii. 2024. https://manoa.hawaii.edu/exploringourfluidearth/physical/atmospheric-effects/ocean-surface-currents#:~:text=As%20current%20moves%20down%20the.

"STM32H7 DevBoard User Manual." n.d. ST Microelectronics.

"VectorNav DevBoard User Manual." n.d. VectorNav.

"VectorNav's VN-100 IMU/AHRS." n.d. VectorNav. https://www.vectornav.com/products/detail/vn-100.