A Wave-Powered Autonomous Met-Ocean Sensor (Process and Electrical)

Ahmad Shah
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, United States
sshah6@stevens.

I. Introduction

A. Project's Aim

B. The Approach

In the event that any of our transmission or data collection devices happen to experience any faults, there must be implemented a way to aid it.

To tackle this issue our team decided to create a circuit to act as the aid to this issue. This circuit would serve a purpose as a fail safe for our device in the event it malfunctions. This fail safe intends to tackle any hardware faults or damage. The intended design is to create 2 separate transmission systems on the Pendulum based wave energy converter. The primary system operates under normal conditions, while the secondary system serves as a backup.

As for the data transmission setup, we have chosen Arduino as our primary processor due to its ease of implementation and the range of sensor compatibility. The means of data transmission we chose was radio transmission due to its cost-effectiveness against other methods such as satellite transmission. It also has a larger range than other methods such as wifi or Bluetooth. Our device initially focuses on measuring water turbidity and temperature, with plans to expand its capabilities to include metrics such as salt concentration and ocean currents.

C. Power Flow Description

Battery	Voltage	Purpose
Lithium Ion Battery	12 V	Arduino, Sensors
Lead Acid Battery	6 V	Circuit, etc

Fig. 1.

The Pendulum based wave energy converter's power system is a hybrid of lead and lithium batteries. The lithium battery is able to provide a constant 12V, powering the arduino and sensors, whilst the lead battery provides 6V and is used to power the circuit and any other systems we may add to the device. We also require a way to charge the batteries to ensure continuous data transmission over long voyages. In order to accomplish this task we plan on using appropriate gear ratios to maximize the energy output of our pendulum device with our generator. The electricity will power two charge controllers, each used to control the charging rate and power to each battery ensuring optimal charge throughout the deployment of our device.

II. Fail-Safe System

In our early development of our failsafe system, we prioritized certain key requirements. The system must reroute power in the event the primary module either fails or becomes unresponsive due to some issue with the program or the module. Additionally, it's crucial for the circuit to be cost-effective and to operate using minimal power, given the constraints of our battery source.

Originally there exists an internal fail-safe system within the Arduino, known as a "Watchdog Timer." This feature can be used in the event the Arduino is ever frozen executing a specific task or some error has been thrown by some part of the program. However, this feature can only serve its purpose to solve any software related issues. While our device does include the use of various sensors, it is unlikely that the Arduino would run into issues in sampling data. Rather a more existent threat is physical damage such as water damage. Thus we must construct a system that ensures the hardware is working as expected, not the software.

To effectively manage power consumption while ensuring reliable operation, we devised a "Heartbeat Circuit." This circuit is designed to interpret heartbeat signals from the primary Arduino, which traditionally signifies that a processor is functioning properly. Typically, interpreting these signals would need an additional processor, but in an effort to minimize power usage, we aimed to avoid a setup that relies on three separate Arduinos.

In a conventional three-Arduino configuration, the primary Arduino would transmit a heartbeat to a secondary Arduino, which in turn would reroute power to a tertiary Arduino in the absence of the heartbeat signal. However, this approach presented two significant challenges. First, it required constant power supply to two Arduinos, which was not aligned with our goal of power efficiency. Second, it introduced a vulnerability: if the Arduino designed to receive and interpret the heartbeat malfunctioned, the entire system could fail. Therefore, our

Heartbeat Circuit, which is tailored for low power consumption, emerged as a more suitable solution for these constraints.

Our Heartbeat Circuit solution uses low-power modules and integrated circuits (ICs) to replicate the functionality of an Arduino programmed to interpret heartbeat signals. In our initial design, as depicted in Figure 1, we incorporated a combination of relays and a timer module.

First Iteration

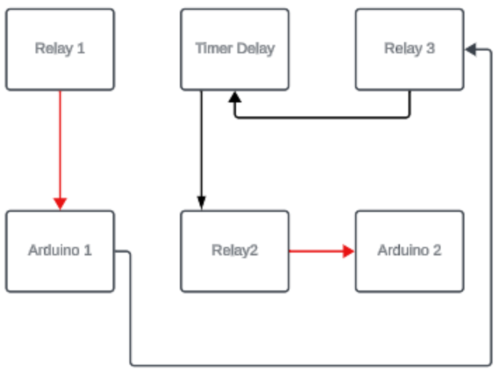

Fig. 2.

Figure 2 depicts the underlying logic of our design. Initially Arduino 1 would be receiving the main power (indicated by red). The Arduino would also be sending a signal to relay 3, turning its state off.

In the event the Arduino is incapable of sending the signal to the relay, the relay 3's state would switch on and start the timer delay. After a delay of 5 seconds it will signal to relay 2 to send power to the second Arduino and cut all power to Arduino 1.

Final Iteration

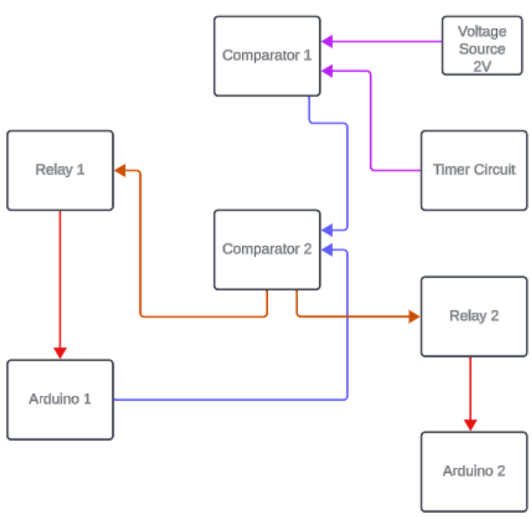

Fig. 3.

After improving on our model several time, we had finalized our design to the model above. Comparator 1 is comparing the signal from a timer circuit to a voltage source of 2 V. This is done to ensure that the timer circuit does not preemptively trip the system as it does produce a LOW voltage of around 1.3 V and a HIGH voltage of 3 V.

Next Comparator 2 compares the voltage from Comparator 1 (which will activate once the timer is activated) to the signal from Arduino 1.

Initially Arduino 1 is powered and sends a signal to Comparator 2. If the signal from the Arduino is higher than that of the timer circuit the Comparator will not activate either of the relays, thus keeping Arduino 1 powered.

In the event Arduino 1 is unable to send power, the Comparator would then send a signal to Relay 1 to switch off, and a signal to Relay 2 to switch on. At this point, Arduino 2 will be powered and Arduino 1 will be turned off.

III. Data Retrieval

A. Temperature:

For our temperature measurement we had chosen the DS18B20 temperature sensor. The DS18B20 is a one-wire digital temperature sensor, meaning it requires only one data line to communicate with the Arduino. The sensor requires a voltage of 3 - 5.5 V, with an operating voltage of -55 °C to 125 °C and an accuracy of +/- 0.5 °C.

Fig. 4.

B. Turbidity:

The turbidity of a liquid is measured in Nephelometric Turbidity Units (NTU). The NTU of clear water is around 1 NTU, and the detection range of the sensor for the level of turbidity is from 0 - 4550 NTU. The sensor utilizes a light emitting diode and a receiver to measure the turbidity of water, the more particulates in between the light emitting diode and the receiver, the weaker the signal. The sensor returns a voltage signal that is read by the Arudino from a range of 0 - 800 units. A reading of 800 indicates clear water, whilst a lower reading indicates the presence of particulate matter obstructing the light emitting diode path.

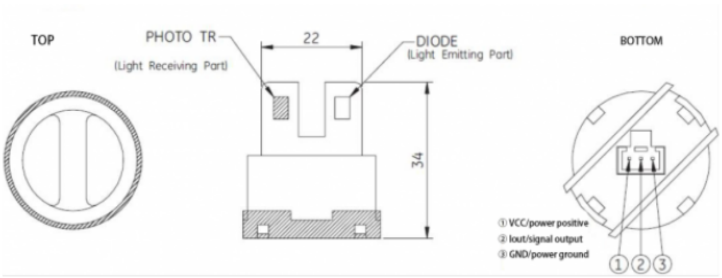

Fig. 5. Turbidity Sensor Dimensions

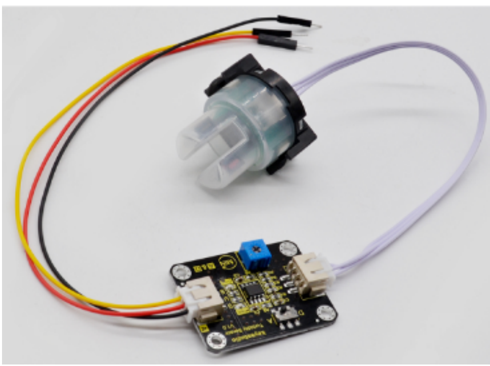

Fig. 6. Turbidity Sensor

IV. Calibration

Temperature sensors often need calibration to effectively measure metrics. The graph below demonstrates our testing for calibration, with an observed high of 101.44 C in boiling water and a low of 0.44 C in ice cold water. In our code we have subtracted a value of 1.44 C from the reading to attain an accurate reading. Testing the lower end of the temperature was done to confirm the accuracy of the temperature sensor under extreme cold.

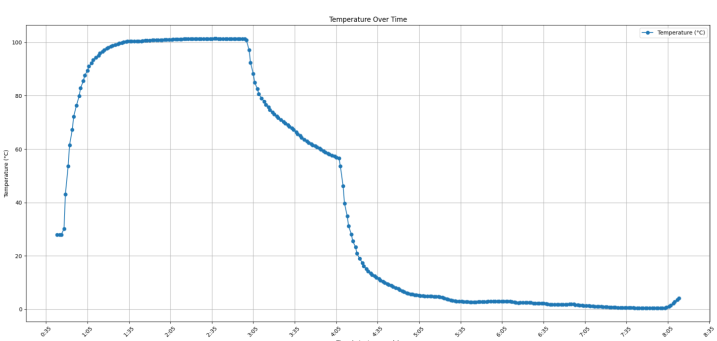

Fig. 7. Temperature Sensor Graph

V. Experimental Data Collection

In our initial testing we set up three distinct tests. First measuring the turbidity and temperature of clear water at room temperature, next we measure the qualities of the salty water, lastly we measure the qualities of warm water. We expect the temperature of the clear and salty water to be around 20 degrees Celsius and the warm water somewhere above 40 degrees Celsius. We also predict the turbidity of the clear water and the warm water to be similar with great water clarity, while the salty water would yield a lower value.

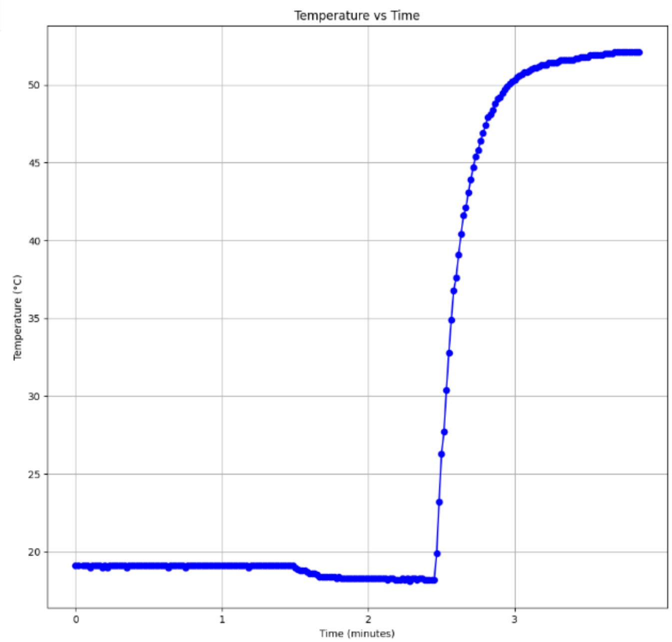

Fig. 8.

The temperature probe measured a temp of around 18 C in the first sample (clear water), and drops slightly more when placed in the second sample (salty water). When the probe is placed in the warm water, the temperature increases significantly achieving a maximum measured temperature of 52.1 C.

The turbidity sensor first measures a value of 745 which indicates clear water, once placed into the salty water the value drops significantly, and when placed into the warm water, it again increases.

VI. Data Transmission

In order to effectively transmit data over large distances accurately various different methods were explored. Firstly, Wifi and bluetooth transmission methods were investigated. This mode of transmission is relatively easy and simple to implement with high data transfer rates, allowing the transfer of large continuous data without needing to be chunked or compressed. However, its range was severely limited to the range capabilities of nearby routers, furthermore it's more vulnerable to unauthorized access and is generally slower and less reliable in data transmission. Satellite Transmission

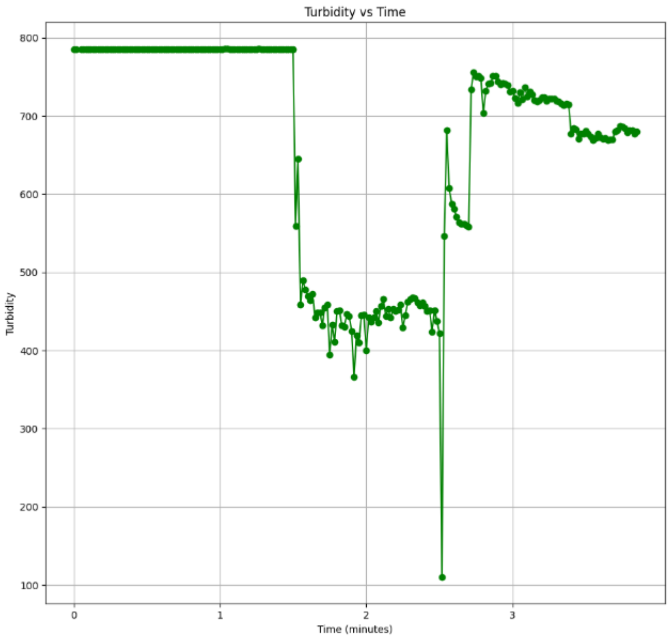

Fig. 9.

is capable of providing coverage in remote areas, and is particularly useful in areas where there is a lack of necessary communication infrastructure. Although Satellite Transmission is the most desirable and will be utilized in final development, its setup and operation costs are high. Lastly, we investigated radio frequency transmission which boasted low-cost deployment due to a plethora of radio modules, more importantly its range is quite impressive. Radio Frequency transmission allows for data transmission up to 100 meters with cost-effective components, and can be easily be used with a power amplifier and a Low-Noise Amplifier to enable a range of up to 2 kilometers.

For our purpose, we decided radio transmission more aligned with our requirements, and had chosen nRF24L01 Radio Module as our transmission device. Furthermore, the model we had selected comes with a RFX2401C chip with an integrated power amplifier and low-noise amplifier, that increases the range of our module from 100 meters to 1000 meters. The module can utilize 125 different channels, with each channel able to have 6 addresses. The 125 different channels allows for 125 different modules to be utilized within close proximity of each other, and each module is able to have 6 addresses allowing for communication with 6 other units at the same time.

From our device we are sending approximately 50 character messages, consisting of the temperature, turbidity and the time. Each character uses 1 byte, and our program is sending 50 bytes per second, accumulating to 180,000 bytes per hour or 0.172 MB every hour. In our testing we received 3600 messages in an hour, resulting in 122,400 bytes being received of the 180,000 bytes sent. Our experimental and testing evaluations were close to our expectations. Furthermore its important to note that in our approximation we used a base 50 character for each message, whilst in actual testing our message size which can further explain this disparity.

VII. Battery Discharge Graphs

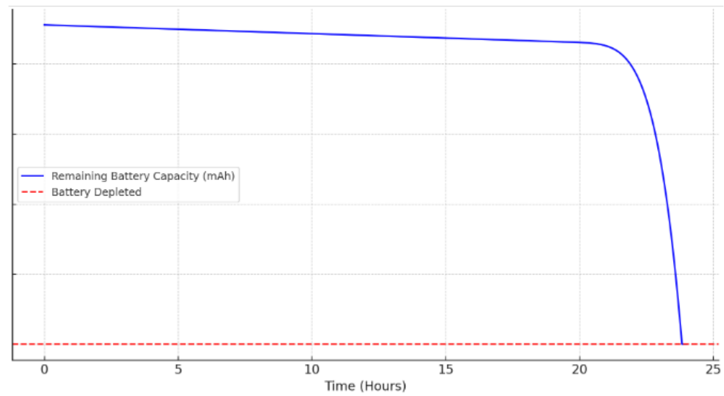

Fig. 10. Simulated Discharge Graph of Lithium Battery

Before testing the discharge rate of our lithium battery, we modeled the expected rate to verify the performance of our Arduino and its sensors. This was done to ensure that our power consumption was predictable and precise, enabling us to accurately forecast the impact of adding more sensors and devices on the longevity of our device. Figure 10 displays our predictions.

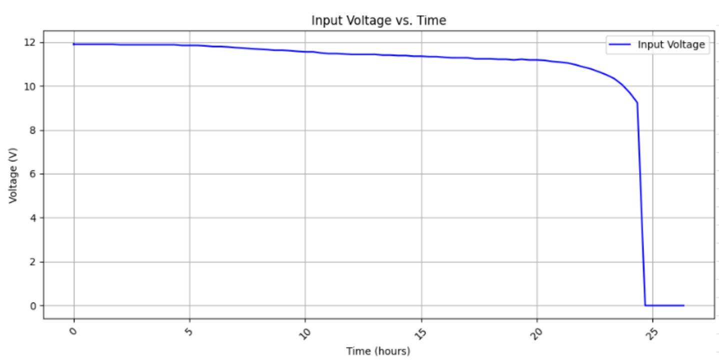

Fig. 11. Observed Discharge Graph of Lithium Battery

Figure 11 displays that our simulation graph is similar to our experimental graph. In our simulation we had predicted that there would be no remaining power around the 24 hour mark, and as seen in our observed study, the battery was depleted at 24.5 hours. This indicates that our calculations were accurate and that the method used to create the simulation, is sufficient to predict battery life with future additions of additional sensors. The graph can be split into four notable stages:

- **Stage 1: Initial Drop:** The curve starts with a slight drop in capacity.
- **Stage 2: Mid Section:** The curve remains mostly flat, where the battery maintains a stable output.

- **Stage 3: Sharp Decline:** As the battery approaches depletion, the curve drops sharply, indicating it is nearing the end of its discharge cycle.
- **Stage 4: Depleted Battery:** At this stage, the battery is incapable of supplying any power.

VIII. Energy Production

We tested two different generators to harvest the rotational energy of our Pendulum based wave energy converter. The first generator we tested is a DC generator, whilst our second generator is bi-directional and uses a high gearbox ratio, and generates AC power.

We tested the max voltage production of each generator, and their relative ability to keep a stable output.

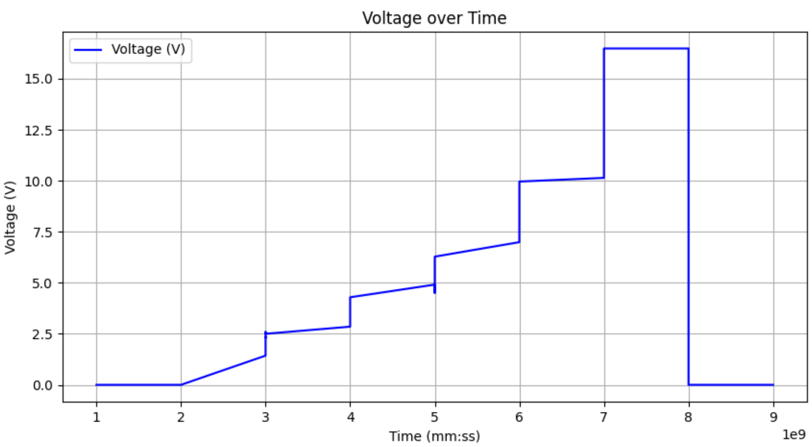

Fig. 12. DC Generator Output

The graph above depicts the DC generator energy production, and the time is in units of tens of seconds. The DC generator was able to reach a maximum voltage of 16 V, with stable and incremental jumps in voltage in correlation with the rpm of the rotating shaft.

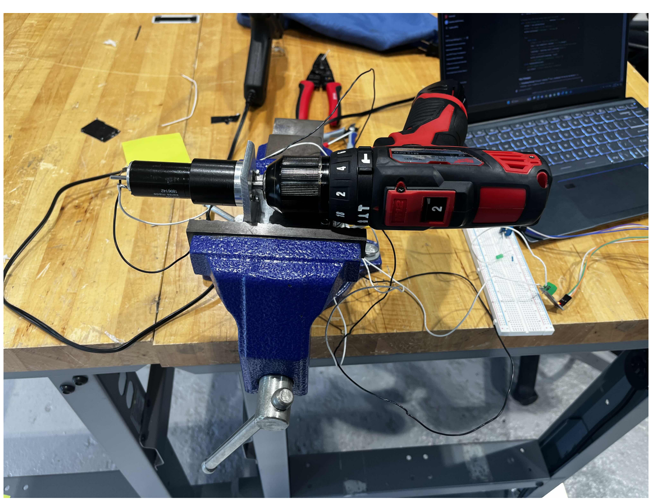

Fig. 13. DC Generator Testing

To rotate the generator's shaft, a drill was employed, supplemented with various adapter pieces to ensure a compatible connection. The DC motor achieved 16 V at 1500 RPM, however it was incapable of producing any voltage beyond 16.48 V, its likely a lower RPM is sufficient to produce this voltage.

Fig. 14. AC Generator Testing

When testing the maximum voltage production of the AC generator, we found that the voltage production exceeded that of the voltage sensor we were utilizing. Thus we used a Voltmeter to measure the absolute max voltage, which was observed to be 54.5 V at ¡1500 RPM.

As the AC generator was capable of producing a higher voltage and made use of an internal gearbox, we had selected the AC generator for future implementation.

Fig. 15. AC Generator and Pendulum Device

Above depicts our current setup, with our generator being mounted vertically on the shaft of the pendulum. As our devices rely on DC power, we first include a small rectifier circuit to convert the AC to DC power.

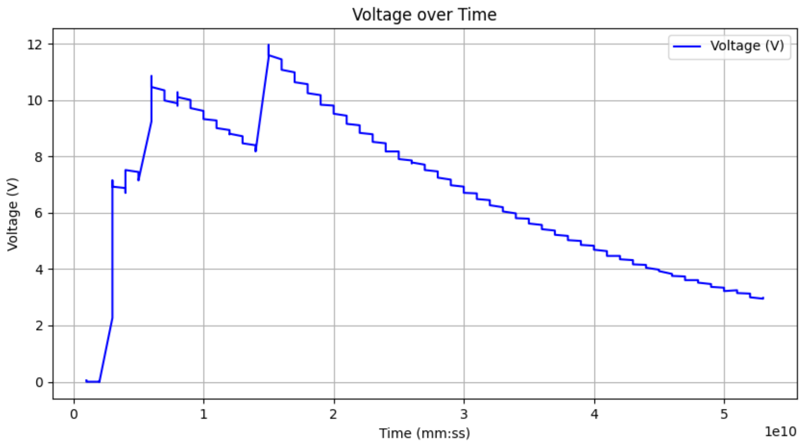

Fig. 16. AC Generator Testing

When we tested the experimental energy production of the Pendulum based wave energy converter, we obtained the following graph. In addition to the rectifier circuit, we added a capacitor to serve as a smoothing filter to lessen the impact of immediate changes in voltages.

Currently we are experiencing issues with generating a constant expected voltage in order to power the charge controllers that will recharge the batteries being used. Our generator produces a varying voltage depending on the speed of the pendulum, utilizing a boost converter would solve allow us to generate higher voltages. However, if we implement a boost converter catered towards calm waters which would produce 4 V and boost it to 12 V, in the event that the pendulum produces more energy due to disturbed waters, the same boost converter would further boost 12 V to a higher value that would likely cause damage to our rectifier, smoothing capacitor, and charge controllers.