ECE 445

Spring 2024

Senior Design Design Document

Automatic Ice Fishing Rod

Team 60

Luke Boelke
James Niewiarowski
Andrew Osepek

TA: Zicheng Ma

February 22, 2024

Table of Contents

1 Introduction

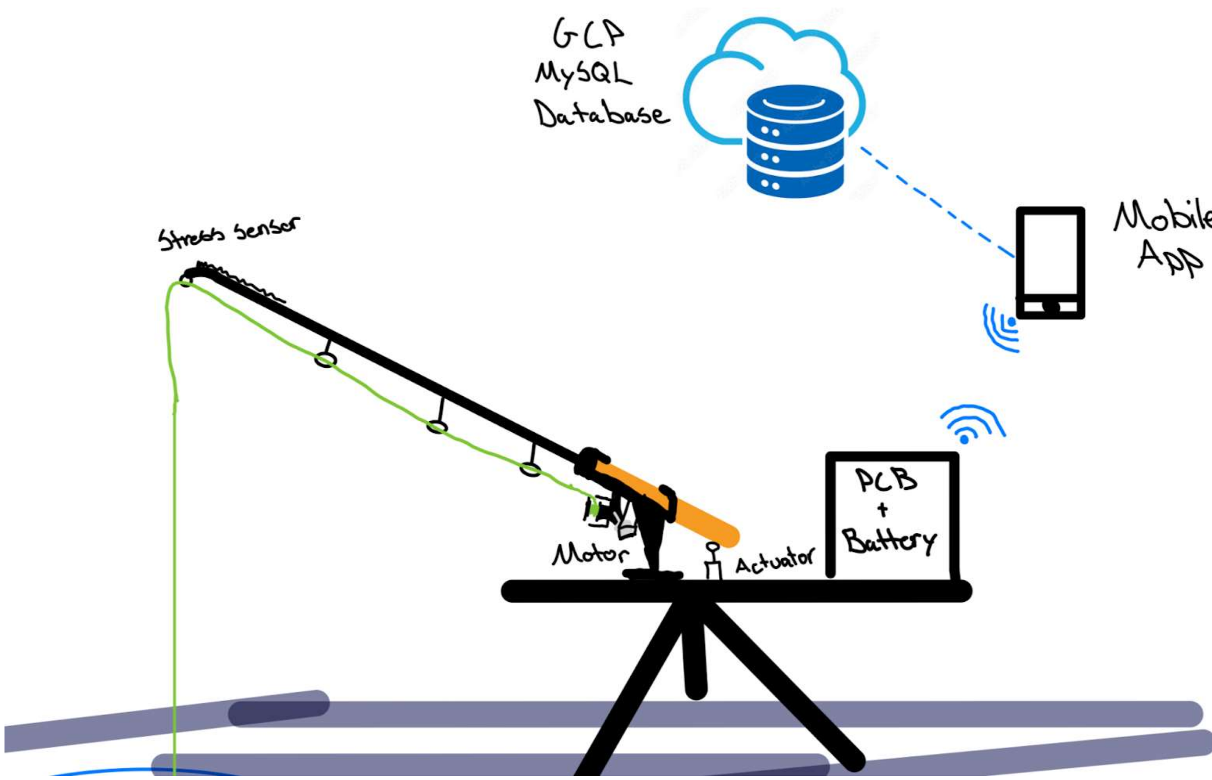
1.1 Problem

Ice fishing can be a very tedious and labor-intensive process. While it is being performed, the fisherman must dedicate all of their attention to the task at hand, constantly jigging the rod, making multitasking impossible. It must be done in a very cold environment as well, which gets uncomfortable after long periods of time. Additionally, there can be long stretches with little to no bites. If the fisherman did not have to constantly attend to the rod, these stretches of no activity would be perfect for taking a break to warm up, eat a meal, etc., but the nature of ice fishing makes this impossible.

1.2 Solution

Our project aims to create an automated ice fishing rod that eases the challenges associated with ice fishing. The user will have the ability to spool any lb-test line onto the device as with any lure when fishing. The fisherman can set the length of the line out from the reel. The fishing rod will have the ability to jig the attached lure in hopes of attracting fish. When a tug occurs at the line, the user will be alerted through an alarm and notification. A mobile app will allow the user to set preferences to the depth of the line and jigging.

1.3 Visual Aid

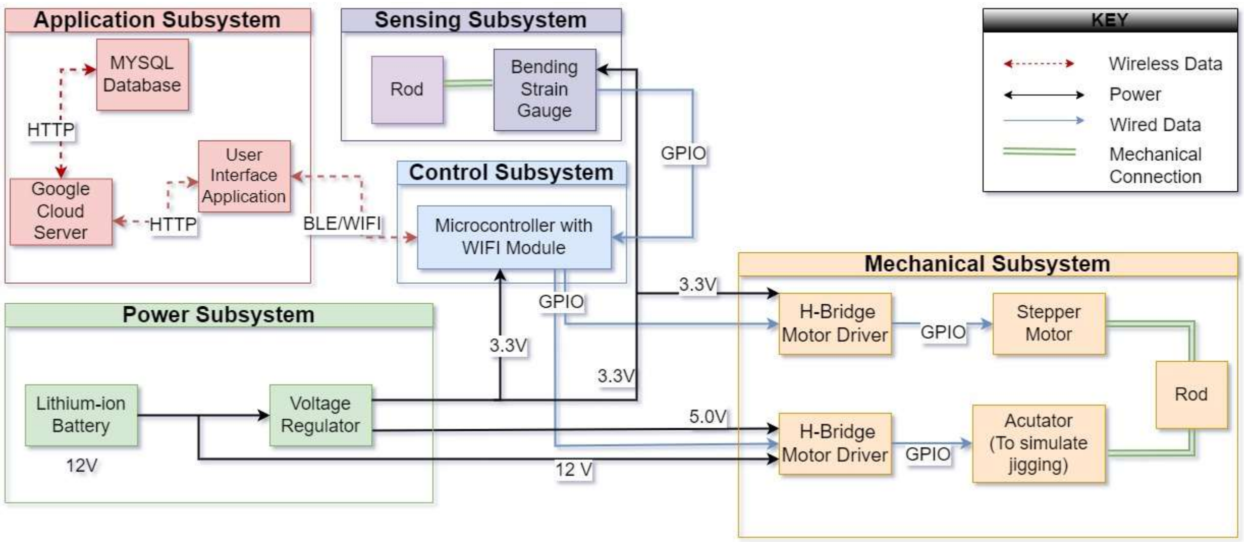


1.4 High-level requirements list

1. The user will be able to set up to 3 different jigging frequencies, a lure depth up to 50 feet (+/- 5 feet) in increments of 1 foot (+/- 0.25 feet), or enable auto reeling.
2. When a bend angle of 30 (+/- 10) degrees is detected, the jigging will halt within 5 seconds of detection, a notification will be sent to the user application, and, if auto reeling is enabled, the line will be reeled in automatically.
3. The user will be able to record their catches in the user application with 7 different data fields. Previous catch information can be viewed in the application.

2 Design

2.1 Block Diagram

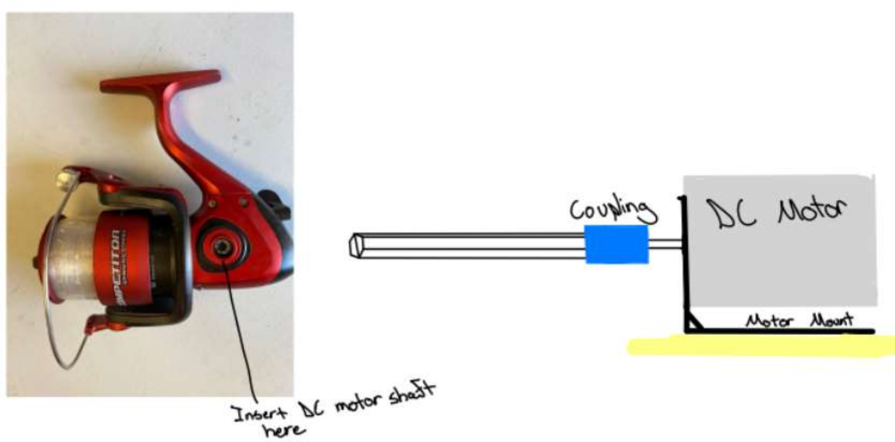


2.2 Physical Design Overview

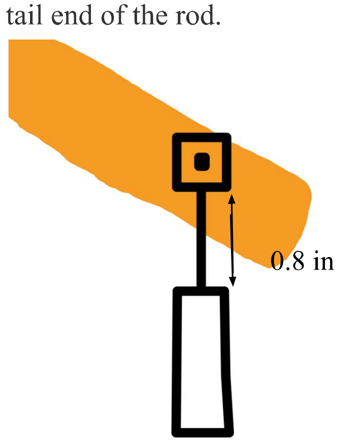
We plan to use a spinning reel rod holder for supporting our fishing rod as shown below. The base of our rod holder will be mounted to a plywood board that will hold our battery and PCB board too. The fishing rod holder has a joint that can be loosened which will allow our rod to move up and down to support our jigging feature.

For the automatic reel, we plan to modify a spinning fishing reel. A spinning reel is driven by a hexagonal shaft as illustrated in the figure below.

To connect our stepper motor to the spinning reel, we will join a hexagonal shaft to our stepper motor's shaft using a coupling. The hexagonal shaft will slide into the reel's slot where the original hand crank shaft was placed. Finally, the stepper motor will be mounted to the fishing reel. A spinning reel can spin in both directions (reeling in/out), so we can control the depth of our line by reversing the direction of the stepper motor to either raise or lower the lure.



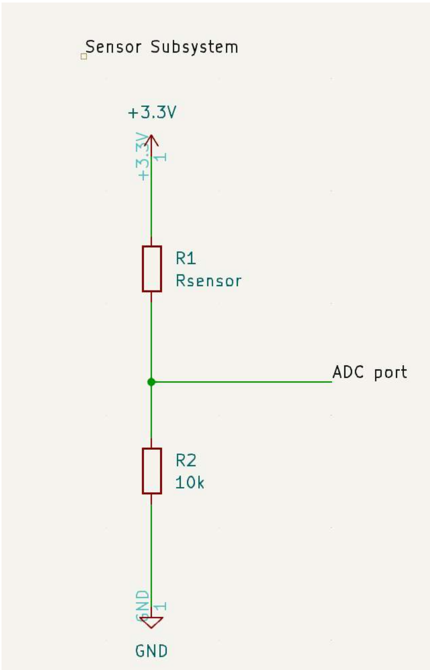
The actuator will be mounted at the tail end of the rod.



Note: We plan to use the machine shop for many of our mechanical builds.

2.3 Subsystem Descriptions, Requirements, and Verifications

Sensor Subsystem



Configuration for Sensor Subsystem

The sensor subsystem will consist of a flex bend sensor placed along the middle of the fishing rod. Flex bend sensors act as resistors, in which the resistance value is directly related to the bend angle of the sensor. We will use the voltage divider rule to allow the microcontroller to calculate the resistance value of the sensor, and then use this resistance to calculate the corresponding bend angle. One pin of the sensor will be connected to a 3.3V supply line, and the other will be connected to the STM32 microcontroller's analog to digital converter port, as well as an additional known resistance value, which will then be connected to ground. The microcontroller will poll the ADC port to continually read the voltage value immediately after the sensor. This means that the voltage across the sensor will be 3.3V minus the polled value. Using the voltage divider rule, the microcontroller can then use this voltage to calculate the resistance of the flex sensor, and then use the linear relationship to calculate the corresponding bend angle. If the bend angle of the flex bend sensor is determined to be greater than the threshold angle of 30 (+/-10) degrees, the microcontroller will alert the fisherman via the user application, halt the jigging function of the stepper motor, and, if the auto-reeling setting is enabled, the signal the stepper motor to reel in the line.

Requirements	**Verifications**
• The Sensor Subsystem is able to read and accurately reflect the bend angle of the rod	• Keep rod still and take note of the resistance value • Add attachment and begin applying pressure to slowly bend rod • Continue measuring resistance value • Use a protractor to measure rod angle and

	compare with measured resistance using our formula
• When auto-reel is enabled the line is reeled in when the max bend angle is exceeded	• Begin applying force to rod to bend at least 30 (+/-10) degrees • Ensure that the auto reel setting has been enabled in the application setting • Make sure the stepper motor is activated and reels in the line fully

Application Subsystem

The application subsystem consists of three units: the user application, the Google Cloud Server, and the MySQL database. The user application will force the user to create an account before using the application. Once an account has been created, the user will be able to login to their account, and control the configuration of their automatic ice fishing rod. The application will be presented in a user-friendly mode that easily allows the user to modify the settings of the fishing device, including the depth of their lure, the frequency of the jigging motion, whether to turn on/off jigging, and whether to enable auto-reeling. The user will also be able to insert/remove catch information from their account through a server hosted in GCP. For example, when they make a catch, they can type in the time caught, location, depth of lure, type of fish, etc. into the app, where it will then be uploaded to a GCP database. This information can then be viewed within the app for future reference. Two tables would exist in this MySQL relational database: a user table that contains user information and a catch table that contains catch information. Attributes of the user table would include a unique username, password, first name, and last name. Attributes of the catch table would include time caught, location, depth of lure, type of fish, length of fish, weight of fish, and other information. The application will directly talk to the microcontroller via the STM32's WIFI/Bluetooth capabilities to relay the desired depth, jigging, and auto-reeling settings to the fishing rod. If the microcontroller detects a fish on the line through data received by the sensor, the microcontroller will signal the user application via WIFI/Bluetooth to notify the fisherman.

Requirements	**Verifications**
• The user is able to create an account and login to the system	• Turn on the DB instance and application server • Have the user create an account in the system • Check the database to see that the users' first name, last name, username, and password have been stored
• The user is able to store catch information while logged in	• Ensure the instance is running the same as above • Have the user logged in already • Input a catch information entry into the

	app • Query DB to see successful store for user • Log out of the application • Log back in and go to view catch information to see that the user can still view ONLY their catch information
• The application can set desired depth and jigging settings	• Navigate to the rod controls in application • Input depth into field and option to enable jigging • Print packet sent from application to microcontroller and ensure data passed matches input • Ensure that microcontroller has received the packet/request from the mobile application

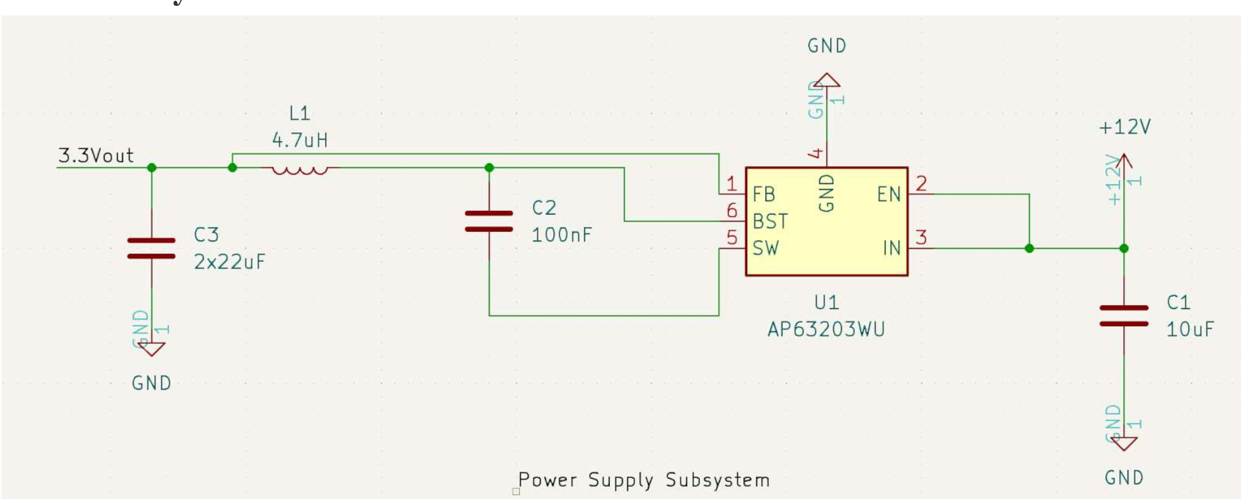
Control Subsystem

The control subsystem will consist of an STM32 microcontroller which will be responsible for all processing data and executing instructions. This subsystem will receive, process, and send data to the other components of the device. It will signal the stepper motor how much to reel the line in/out, and the frequency at which the solenoid should jig the rod. It will receive data from the flex bend sensor and determine whether the threshold angle is met, causing the microcontroller to signal the mechanical subsystem to halt the jigging of the rod, and, if auto-reeling is detected, reel the line in. The microcontroller will communicate with the mechanical subsystem's stepper motor and solenoid via the GPIO protocol, and will communicate with the flex bend sensor via its analog to digital converter port. The microcontroller will also communicate with the user application through the STM32's Bluetooth/WIFI capabilities, receiving the data for the user's desired settings and sending a notification when a fish is detected.

Requirements	**Verifications**
• The microcontroller can receive requests from the application	• Send request from user application • Program microcontroller to print out data taken from application • Ensure the data is consistent and that correct values exist in microcontroller functions
• When the sensor exceeds maximum bend angle stop jigging and reel line in	• Turn on jigging and set line depth of rod to a non-zero value • Bend rod until it exceeds the programmed maximum angle • Check that microcontroller has stopped signals to activate jigging actuator

	• Check that microcontroller has outputted a signal to reverse direction and activate the stepper motor
• The control subsystem can take instructions to reel the line in/out and can correctly calculate how to instruct the stepper motor	• Send input to microcontroller to reel out the line • Check that microcontroller receives requests and execute function to translate into signal for stepper motor • Wait until the stepper motor finishes and verify accurate calculation by using a tape measure to check the line length • Send input to reel in the line to microcontroller • Verify signal received and wait until stepper motor finishes • Use tape measure to check that the reeled in line length is consistent with the input given the provided tolerances
• The control subsystem can receive inputs to alter the jigging frequency and communicate these updates to the actuator	• Set a very slow frequency for the actuator • Verify signal received by microcontroller and that rod is jigging slowly • Set a new high frequency jigging instruction • Verify that the signal is received and the rod speeds up the rate it is jigging at • Turn off the jigging system • Verify the actuator is no longer active

Power Subsystem

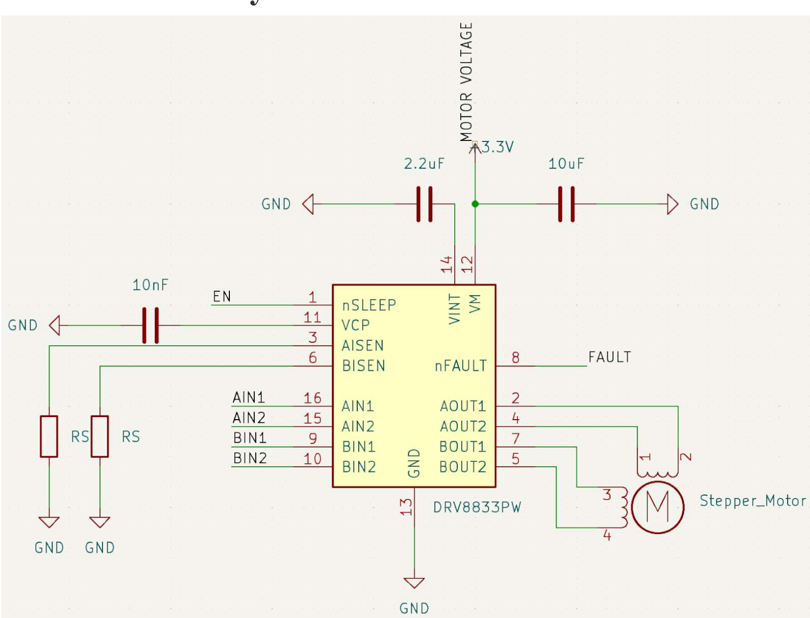


Configuration for Power Supply Subsystem

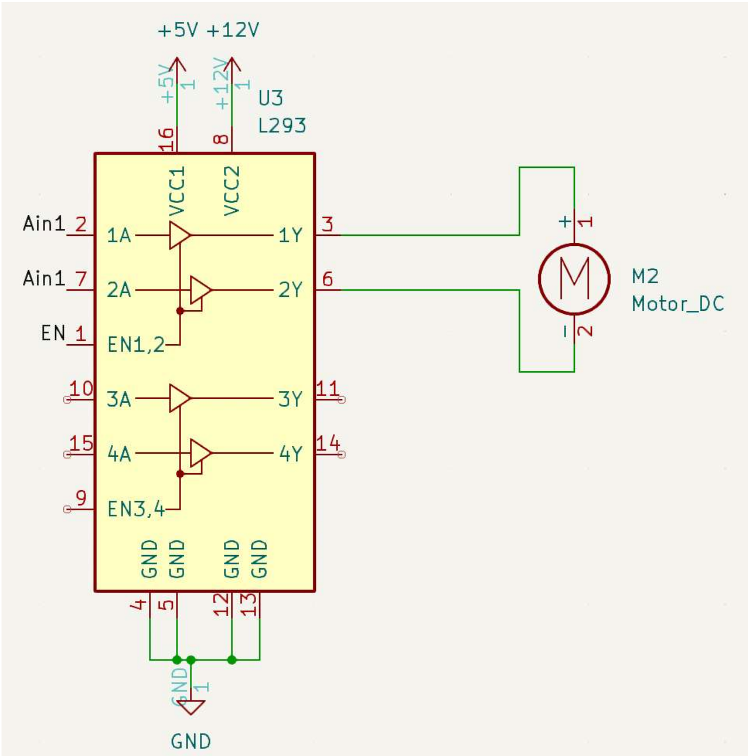
The power subsystem will be responsible for delivering power to the other subsystems of the device. This subsystem consists of lithium-ion batteries connected in parallel to provide a power source with a voltage of 12V and a voltage regulator to drop the voltage to the values required by other subsystems. The power subsystem will need to deliver 3.3V (+/- 0.1V) to the microcontroller, flex bend sensor, and stepper motor 5V (+/- 0.1V) to the motor driver, and 12V (+/- 0.1V) to the solenoid. There will also be a switch on the system to shut off the power supply to prevent the batteries from draining too fast. The power subsystem will be designed to maximize efficiency of electricity used and try to reduce energy loss.

Requirements	**Verifications**
• The power subsystem can reduce the significant voltage input to the appropriate component voltages	• Connect the power subsystem to the power source • Check that the voltage across the microcontroller terminals is 3.3V (+/- 0.1V) • Check that the voltage across the flex bend sensor is 3.3V (+/- 0.1V) • Check the voltage across the stepper motor terminals is 3.3V (+/- 0.1V) • Check the voltage across the motor driver is 5V (+/- 0.1V) • Check the voltage across the solenoid terminals is between 12V (+/- 0.1V) • These measurements should also allow for the provided tolerances

Mechanical Subsystem



Motor Setup for Stepper Motor Driving Fishing Reel



Motor Setup for Actuator

The mechanical subsystem will consist of an ice fishing rod (short rod length) that is being held by a fishing rod holder (as described in the Physical Design Overview), which is attached to a tripod stand, thus holding the rod upright and dangling the line above the water. A 12V actuator will be attached to the tripod stand, near the tail end of the rod, and will push the rod up and down repeatedly to simulate the jigging motion A H-bridge driver will control the actuator's direction. A 3.3V stepper motor will be attached to the fishing reel to allow for line to be automatically reeled in or out. A H-bridge driver IC will reverse the polarity across the stepper motor to allow the stepper motor to switch directions for reeling in and out. The mechanical subsystem's stepper motor and solenoid will receive data from the microcontroller via the GPIO protocol. The microcontroller will signal the stepper motor how much to rotate in order to reel the line out to the correct depth, what frequency the solenoid should jig the rod (according to the settings in the user application), and when the solenoid should stop the jigging motion. If a fish is detected by the sensor subsystem, the microcontroller will signal the solenoid to halt jigging and, if auto-reeling is enabled, signal the stepper motor to reel the line all the way in.

Requirements	**Verifications**
• The stepper motor can reel the rod in both directions	• When the microcontroller output triggers the stepper motor, the line reels out. • Change the direction of the line • Turn on the stepper motor again and verify the line is being reeled back in
• The actuator can jig the rod up and down at different frequencies	• Turn on the actuator at a low frequency • Verify the rod is moving up and down at a

	slow rate • Increase the actuator's frequency • Verify the rod is jigging at a faster rate • Turn off the actuator • Verify the rod has stopped jigging and is stable

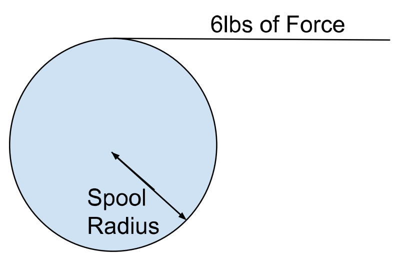
2.4 Tolerance Analysis

Force Exerted by a Fish on the Line

Our automatic ice fishing rig is designed to be used on small to medium inland lakes. Many fishermen prefer to use 4-8 lb test line when selecting a fishing line weight. Line weights in this range can withstand the stress of many species of fish commonly caught in inland lakes. During the Winter, fish are more sluggish and will not pull as hard as in warmer weather; as a result, we can assume up to 6 lbs of force will be applied to our line. When determining the power of our stepper motor, we need to select a stepper motor that can operate under the stress of 6 lbs of force.

$$6\ lb\ of\ Force\ = 26.689\ Newtons$$

Using the force calculated, we can determine the necessary torque that will drive our fishing reel.



Determining the spool radius is challenging, because as the line is reeled-out the radius will decrease and as the line is reeled-in the radius will increase. We believe on average our spool's radius measures between:

$$12.7\ mm\ -\ 19\ mm$$

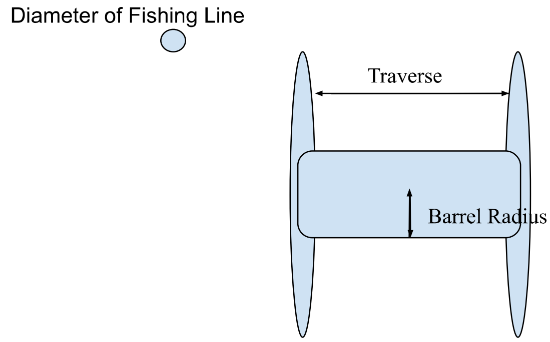
Calculating Torque needed by our stepper motor.

$$Torque\ (Nm)\ = 26.689\ Newtons\ \star\ (0.0127\ m\ -\ 0.0190\ m)\ =\ 0.34\ Nm\ -\ 0.51\ Nm$$

The stepper motor can supply up to 0.65 Nm of torque, so it can support the stress we expect.

Measuring Line Out

Using our spool's radius, the amount of fishing line on our reel, and the length of our fishing rod, we can determine how much fishing line is out.



Diameter of Fishing Line:
- 8 lb monofilament fishing line: 0.27mm = 0.00027 m

Barrel Radius:
- 0.0127 m

Traverse (length of spool):
- 0.0381 m

Circumference (No line on):
- $2\,\Pi r \;=\; 2\Pi(0.0127\,m)$ = 0.0797 m

Individual threads of line across spool's length:
- 0.0381 m / 0.00027 m = 141.11 ≈ 141 threads

Each layer of line will add 0.00027 m to the spool's radius.

Suppose 100 m of line is spooled onto the reel. Calculate the number of layers, n, of line on the spool:
- $100m \;=\sum_{x=0}^{n-1} (2\,\pi\;(0.0127 \;+\; 0.00027x)\,)\;*\;141$
- $100m \;=\sum_{x=0}^{n-1} 2\,\pi\;(0.0127)\;*\;141 + \sum_{x=0}^{n-1} 2\pi(\,0.00027x)\;\;*\;141$
- $100m \;= 2\,\pi\;(0.0127)\;*\;141\;*\;n + 2\pi(\,0.00027)\;\;*\;141\;*\;n\;*\;(n\;-\;1)/2$
- $n\;\approx 8.25$ layers of line on the spool

Considering there is 100 m of line on the spool. How many rotations of the spool are required to release x meters of line?
- $x \;=\sum_{i=n}^{8.25-1} 2\,\pi\;(0.0127)\;*\;141 + \sum_{i=n}^{8.25-1} 2\pi(\,0.00027i)\;\;*\;141$
- $x \;=\sum_{i=0}^{8.25-1} 2\,\pi\;(0.0127)\;*\;141 - \sum_{i=0}^{n-1} 2\,\pi\;(0.0127)\;*\;141 + \sum_{i=0}^{8.25-1} 2\pi(\,0.00027i)\;\;*\;141$
  $-\sum_{i=0}^{n-1} 2\pi(\,0.00027i)\;\;*\;141$
- $x\;=\;100\;-\;2\,\pi\;(0.0127)\;*\;141\;*\;n\;-\;2\pi(\,0.00027)\;\;*\;141\;*\;n\;*\;(n\;-\;1)/2$

Converting this into a quadratic equation in terms of n, we get

- $0 = 0.000135n^2 + 0.012565n - (100 - x)/(282\pi)$

We can then solve for n using the quadratic formula to get the layer reached after releasing x meters of line. 8.25 - n will then give us the number of layers needed to unreel x meters of line (where 8.25 is the initial number of layers and n is the final number of layers after unreeling), and because each layer contains 141 threads, we need 141 revolutions to unreel a single layer. This means we need 141*(8.25 - n) rotations to release x meters of line.

The previous calculation calculates the number of rotations *of the reel* needed to unreel x meters of line. The number of rotations *of the stepper motor* needed for this, however, will be different, due to the gear ratio of the fishing reel. The gear ratio is the number of times the spool rotates for every one rotation of the hand crank. Our reel has a gear ratio of 5.2:1, meaning that every time the stepper motor makes one full rotation, the reel will make 5.2 rotations. So, the stepper motor will need to rotate 141*(8.25 - n)/5.2 times to unreel x meters of line.

The maximum length of line will be 50 ft (15.24 m). Using the above equation, to unreel 15.24 meters of line, we will need 165.493 rotations of the reel. If we ignore the change in radius due to added layers, holding the radius at a constant 0.0127 m, 165.493 rotations will unreel 13.21 m of line (43.34 ft). This means that the difference between accounting for and ignoring the change in radius has a maximum value of 50 - 43.34 = 6.66 ft, which is greater than our tolerance of 5 ft for the depth of the line. This means that we *will* have to account for the change in radius when calculating line depth, using the summation equations shown above.

Microcontroller Data Reading

We will poll the STM32 microcontroller ADC port to read data from the flex bend sensor. In order to ensure that the microcontroller has enough time to convert the voltage value read by the ADC port into the estimated bend angle, and determine if it is above the trigger threshold, we will set the ADC's polling frequency to be much slower than the STM32's clock frequency. We will set the frequency of the ADC peripheral to 32kHz using the internal low-power 32kHz RC, meaning that the sensor's resistance value will be read 32 thousand times per second. If a fish is on the line, this high polling frequency will ensure that the change in bend angle is detected almost immediately. The microcontroller's CPU clock (32 MHz crystal oscillator with integrated trimming capacitors) operates at 32MHz, much faster than the polling frequency of the ADC port.

Instructions/second = 32,000,000
ADC polling frequency = 32kHz
Number of instructions performed between polls = ((32,000,000 instructions)/(1 second))/((32,000 polls)/(1 second)) = 1,000 instructions/poll

This means that 1,000 instructions can be executed by the microcontroller before the next poll from the ADC port. Even though some lines of code will take more than one clock cycle to complete, the analogRead, the conversion from voltage to bend angle, and the check for meeting threshold angle will take less than 1,000 instructions. This means that the polled voltage value will not change while calculations are being performed, giving us accurate results.

Calculating Bend Angle

Because the flex bend sensor's resistance changes based on the bend angle, the STM32 microcontroller will need to convert the resistance to the corresponding angle, so it can then be converted to the corresponding force value. To obtain this resistance value, we will use the voltage divider rule, where one of the resistors is the flex bend sensor. By connecting one side of the flex bend sensor to a constant voltage value (3.3V) and the other side to the STM32's ADC port, we can obtain the voltage drop across the sensor with 3.3V minus the value read by the ADC port.

Variables:

- v: voltage drop across sensor
- y: voltage read by the ADC port
- X: sensor resistance
- R: known resistance value
- X_i: sensor resistance when straight
- X_f: sensor resistance when bent at a 180 degree angle
- a: bend angle of sensor

Calculating voltage across the sensor:

$$v = 3.3 - y$$

Using voltage divider rule:

$$v = 3.3 * X/(X + R)$$

Isolating X to solve for the resistance of the sensor:

$$X = v * R/(3.3 - v)$$

Using this calculated sensor resistance, we can estimate the bend angle with the knowledge that there is a relatively linear relationship.

$$a = 180 * (X - X_u)/(X_f - X_i)$$

Because resistors typically have tolerances between 2% and 5%, our 3.3V voltage line (according to high level requirements) has a tolerance of 0.1V, and the flex bend sensor we are using has a tolerance of 30%, we must account for this when calculating the bend angle.

$$v = (3.3 \pm 0.1) - y$$

$$X = (3.3 \pm 0.1 - y) * (R \pm R * 0.05)/(3.3 \pm 0.1 - (3.3 \pm 0.1 - y))$$

Substituting in our value for R (10kΩ), and canceling out the supply voltages (3.3 ± 0.1) in the denominator because they are the same line and so will share the same tolerance:

$$X = (3.3 \pm 0.1 - y) * (10 \pm 0.5)/y$$

Substituting this value into the equation for bend angle:

$$a = 180 * ((3.3 \pm 0.1 - y) * (10 \pm 0.5)/y - X_i)/(X_f - X_i)$$

Then substituting in our values for the maximum and minimum resistance of the sensor, X_i = 10kΩ ± 3kΩ and X_f = 20kΩ ± 6kΩ (including corresponding tolerance of 30%):

$$a = 180 * ((3.3 \pm 0.1 - y) * (10 \pm 0.5)/y - (10 \pm 3))/((20 \pm 6) - (10 \pm 3))$$

We can then calculate the minimum and maximum values of a using these tolerances, to get a range of possible values:

Minimum value:

$$180 * ((3.3 - 0.1 - ADC) * (10 - 0.5)/y - (10 + 3))/((20 + 6) - (10 - 3))$$
$$= 180 * ((3.2 - y) * 9.5/y - 13)/19$$
$$= 288/y - 213.158$$

Maximum value:

$$180 * ((3.3 + 0.1 - ADC) * (10 + 0.5)/y - (10 - 3))/((20 - 6) - (10 + 3))$$
$$= 180 * ((3.4 - y) * 10.5/y - 7)$$
$$= 6426/y - 3150$$

Expected value:

$$180 * ((3.3 - y) * 10/y - 10)/(20 - 10)$$
$$= 594/y - 360$$

As can be seen from the above equation, the tolerances have a massive impact on the range of possible values when calculating the bend angle. For example, if sensor is bent to 90 degrees, the expected resistance value will be 15kΩ, meaning the voltage read by the ADC (y-variable) port should be 1.32V. Accounting for the tolerances, however, if we read 1.32V, the calculated bend angle will range between 5.0238 degrees to 1718.18 degrees. If our calculated angles are off by this much, our trigger for whether or not a fish is on the line will almost certainly fail, preventing our system from functioning properly. In order to fix this issue, we can simply measure the actual resistance values of our resistors and sensor (both when straight and when bent at 180 degrees), as well as measure the actual voltage of the 3.3V line. This will be one of our first tasks after receiving our parts and building our power subsystem, so that we can significantly decrease these tolerances (especially with regards to the flex bend sensor, as a 30% tolerance is an incredibly large window), making them near negligible when calculating the bend angle.

3 Cost & Schedule

3.1 Cost Analysis

We expect to spend 15 hours/week working on this project for the next 9 weeks. Therefore, we look to spend 135 hours in total per person for completing this project. A comparable industry salary for this type of position would expect $45/hour. Each team member will cost $45/hour x 2.5 x 135 hours = $15,187.50 for the semester. The total team, consisting of three people, will cost $45,562.50 in labor costs.

We expect to spend approximately $166.75 in parts. However, we look to minimize our parts costs by using fishing gear that we have in our garages. The parts cost listed does not take into account this factor.

In total, we expect our project to cost: $45,562.50 + $166.75 = $45,729.25

Description	**Manufacturer**	**Quantity**	**Extended Price ($)**	**Link**
Flex Sensor 55mm Male Pins	Spectra Symbol	1	15.67	LINK
Nema17 Stepper Motor 3.3 V 95oz-in 60mm 1.8A,Hybrid 2-Phase 0.65NM 4-Wire 17HS4218	CNCTOPBAOS	1	17.50	LINK
L293 Quadruple Half-H Drivers	Texas Instruments	1	3.083	LINK
DRV8833PW Dual H-bridge motor driver	Texas Instruments	1	2.016	LINK
Micro Electric Linear Actuators 12V, Stroke 0.8", Force 19.2N/4.32lbs, Mini Waterproof Motion	Poweka	1	29	LINK
AP63203WU-7 Buck Switching Regulator IC Positive Fixed 3.3V 1 Output 2A	Diodes Incorporated	10	7.61	LINK
AP63205WU-7 Buck Switching Regulator IC Positive Fixed 5V 1 Output 2A	Diodes Incorporated	10	7.61	LINK
STM32WB55RGV6	STMicroelectronics	1	8.24	LINK

LiCB A23 23A 12V Alkaline Battery (5-Pack)	LiCB	1	5.39	LINK
360 Degrees MiniPort Fishing Rod Holder	METER STAR	1	14.97	LINK
Ugly Stik Complete Spincast Reel and Fishing Rod Kit	Ugly Stik	1	32.37	LINK
8lb Fishing Line	Zebco	3	3.29	LINK
Plywood 3/8 in		1	5	N/A
Miscellaneous Circuit Elements (Resistors, Etc.)	N/A	N/A	10	N/A

3.2 Schedule

Week	Task	Person
February 19th - February 26th	Finalize and submit design document	James
	Revise and resubmit project proposal	Andrew
	Review schematic for design	Luke
	Prepare for design review	Everyone
February 26th - March 4th	Complete design review	Everyone
	Order parts	Andrew
	Design PCB	Luke
	Order PCB	James
March 4th - March 11th	Measure actual resistance values of flex bend sensor and all other resistors	Everyone
	Solder printed PCB	James
	Begin writing code for microcontroller	Andrew

	Assemble/test mechanical system	Luke
March 11th - March 18th	Begin building power subsystem	Everyone
	Finalize software and program the microcontroller	James
	Test the finished PCB	Andrew
	Connect microcontroller to sensor and MCU calculates correct bend angle	Luke
March 18th - March 25th	Connect power subsystem to components and begin testing	James
	Build user application database instance and structure	Andrew
	Build front end and core application with connection to db	Luke
March 25th - April 1st	Test that the application can send requests to the microcontroller	Everyone
	Test Sensor Subsystem	James
	Test Control Subsystem	Andrew
	Test Application Subsystem	Luke
April 1st - April 8th	Continue to test system as a whole and make necessary changes	James
	Continue to test system as a whole and make necessary changes	Andrew
	Continue to test system as a whole and make necessary changes	Luke
April 8th - April 15th	Fix minor existing bugs	James
	Fix minor existing bugs	Andrew
	Fix minor existing bugs	Luke

April 15th - April 22nd	Demo	James
	Demo	Andrew
	Demo	Luke

4 Ethics and Safety

When developing an automatic ice fishing rod, it's essential to consider various ethical and safety issues, both during the development process and in terms of potential misuse. The IEEE and ACM (Association for Computing Machinery) Codes of Ethics provide general guidelines that can be applied to our project. The issues listed below ensure our project upholds to the highest standards established:

Privacy Concerns (ACM Code 1.6)

Any time personal data is being collected and stored, there is a risk of a breach of privacy. Our application will allow users to upload data to a GCP database, which will contain two tables: one for personal information and one for catch information. The main privacy concern deals with the former, as this table will contain the user's first and last name. In the application's privacy policies, we will clearly state that the information they are submitting will be stored in a database, and ask that they only proceed if they give us consent to store their data. In the event that this database is hacked, this information could be used for malicious purposes that could potentially harm the user. In order to minimize these consequences in the event of an attack, this risk will be clearly stated in the privacy policies, and will inform the user that they may use an alias when submitting this information if they wish to take extra precautions.

Transparency and Honesty (ACM Code 1.3)

When developing a new product, it is important to fully document the entire design and implementation processes. Data should not be tampered with and no values should be altered to ensure honesty and transparency. The claimed capabilities of our design must accurately reflect its actual capabilities, so users receive the quality they expect. In order to make sure we are transparent throughout our development, we will thoroughly document the process in our lab notebooks. We will also make our code open sourced so that users can understand our system and how it works. The claimed values will be realistic and accurate, and will be reported with tolerance values (e.g., +/- 5 N) to account for slight differences in conditions, equipment, etc.

Mechanical and Electrical Safety (IEEE 1.1)

These codes involve safety issues related to mechanical and electrical failures in the rod or its deployment mechanisms.To mitigate these issues we will employ the following precautions. The battery and PCB will be enclosed in an element-proof box to minimize the risk of electric shock and mechanical/electrical damage to the system. The system components will also be properly grounded. The automatic reeling system will operate at a frequency that is safe for the user as the fisherman's line will not be reeled in excessively fast. We will rigorously test the mechanical stability of the rod to ensure that it can withstand the stress of catching multiple fish of varying sizes.

User Training and Guidelines (ACM Code 1.2)

This code relates to safety issues surrounding Injuries due to improper use or lack of understanding of the equipment. In our user manual we will have instructions for using the rod. To mitigate this issue we will provide clear user manuals, safety guidelines, and potentially implement features like emergency stop mechanisms. These instructions will include how to turn on and off the system. There will also be an analog off switch on the power delivery circuit to cut power in cases of extreme malfunction. The user manual will also have safety guidelines such as how far to stand from the rod and advise against putting your hands near the hook. Our user manual will also advise against tampering and have information about how to handle component breakdown such as battery corrosion or leakage.

References

Code of Ethics - Association for Computing Machinery, www.acm.org/code-of-ethics. Accessed 7 Feb. 2024.

"IEEE Code of Ethics." *IEEE*, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 7 Feb. 2024.