Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

CONCEPT OF OPERATIONS

REVISION – 1
23 February 2022

Concept of Operations
for
Smart Cricket Bat

Team <22>

Approved by:

____
Project Leader Date

____
Prof. Kalafatis Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/9/2022	Pablo Barron		Draft Release
1	2/23/2022	Pablo Barron		First Revision

Table of Contents

List of Figures

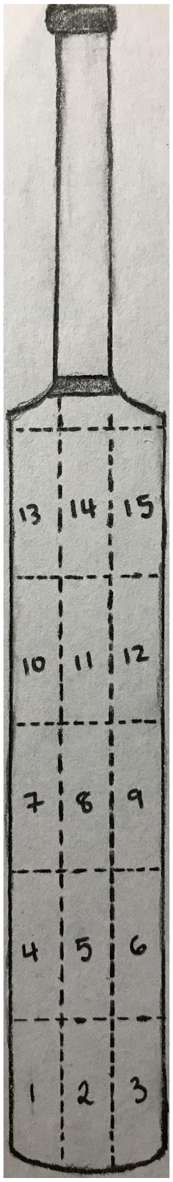


Fig. 1: Cricket Bat Region Subdivision

Fig.2: StanceBeam Device
Charging Dock on the right. Main hardware housing is in yellow. The mounting to attach to the bat below the main hardware housing. Figure above the main housing is to lock the device in place

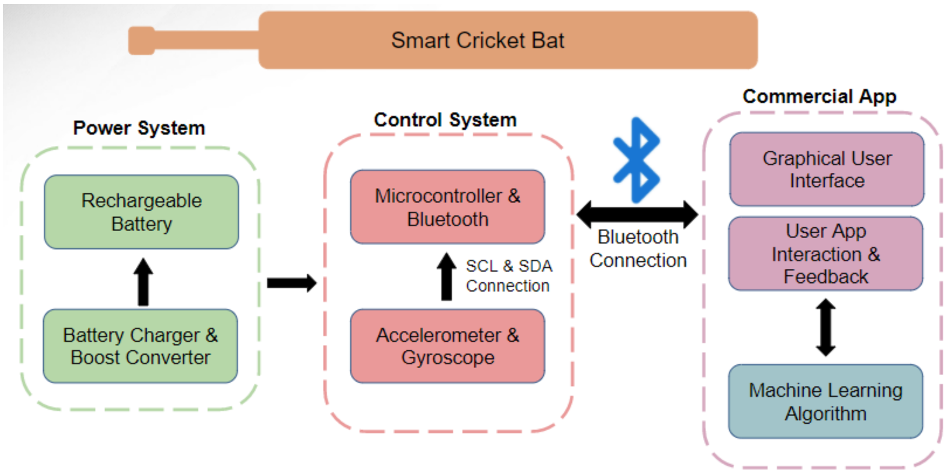


Fig.3: Subsystem Block diagram

1. Executive Summary

While cricket has been around for much longer than its contemporary baseball, there is a stark difference in training equipment available for casual enjoyers, the smart cricket bat plans to fill that space. The smart cricket bat offers an easy at home alternative to coaching lessons in the form of an app and a device you put on your cricket bat of choice. The app will display measurements and calculations from the bat, such as the angle of the swing, the speed of the swing, the force the cricket ball exerts onto the bat, and the approximate position of where the ball hit the bat. The last of which is found via a machine learning algorithm that takes the other measurements and triangulates the impact location. This will help improve both accuracy and precision when swinging for the bats sweet spot.

2. Introduction

Cricket is more of a niche sport than others, so it's no surprise that the community overall is lacking in training and feedback equipment. Most of the training equipment they do have are primarily rudimentary items, such as ball rebounders, batting tees, etc. And recently, one start up, StanceBeam, that is trying to provide easier at home training is a little on the expensive side. The smart cricket bat will fill this gap by providing a device that will attach to the user's cricket bat of choice and acquire various data points on each swing. This device will then connect to the player's phone via bluetooth, process the data acquired from the sensors on the bat, and give real time feedback to help the user.

2.1. Background

In cricket, the goal is to score runs, this is accomplished through completing runs, hitting boundaries, losing wickets, and free runs. Half of these, completing runs and hitting boundaries, involve hitting the ball into a desired area of the field, to accomplish these feats easier the cricket bat is designed to have a "sweet spot" where the bat is thicker than everywhere else, thus having more force behind it. The goal is to have a consistent enough swing to hit the sweet spot as often as possible. StanceBeam is a start-up out of India that is trying to help the average player with this. Their product is a sensor that connects to an app and displays measurements from the bat such as swing angle, swing speed, power generated from swing, and overall swing efficiency. The app also has drills and real-time coaching from real coaches to help improve aspects of their play. In contrast, our app will display where the ball impacted the bat via a histogram, as well as displaying swing angle, swing speed, power generated from swing, and overall swing efficiency. So, while we can't give drills or video analysis, we can help improve swing ball placement and overall consistency in hitting the sweet spot of the bat.

2.2. Overview

The overall goal of the smart cricket bat is to deliver concise, analyzed data from the users swings to the app and give real-time feedback to the user to help improve their overall swing consistency and efficiency. This will be accomplished by a device that will mount to the end of the cricket bats handle and collect both speed and angle of the cricket bat during the user swing. It then sends that data to the smartphone app via bluetooth to process the data, through a machine learning algorithm, and output real time feedback to the user.

2.3. Referenced Documents and Standards

- Reference Device:
  - https://www.stancebeam.com/
- Standard for the Specification of IMU's:
  - https://standards.ieee.org/ieee/1780/5700/
- Cricket Bat Standard:

- https://www.cricketequipmentusa.com/cricket-bats-specifications-recent-changes-to-the-law-52#:~:text=Length%20and%20width%20of%20the,than%2052%25%20of%20the%20bat.

3. Operating Concept

3.1. Scope

The main functionality of the smart cricket bat will be to calculate where on the bat the ball hits based on data gathered from a gyroscope and an accelerometer. Based on data collected from the gyroscope and accelerometer, the speed of the swing and angle of the swing will be calculated by an offline machine learning algorithm. The data gathered from the IMU sensors will be sent via bluetooth to an app. From the gathered data and ML algorithm results, the app will display to the user with a histogram of where on the bat the ball collided. From the data on the histogram, how efficient the user is hitting the ball will be calculated. The app will also show details of every swing ie. speed of the swing and swing angle.

3.2. Operational Description and Constraints

Our device will be attached on the bottom of the handle of the bat. Once the device is attached, it will need to be calibrated to accurately measure collision location and efficiency. Calibration will include hitting the bat in various regions on the bat and inputting the correct region on the app. The calibration of the device will be the main constraint, since failure to properly calibrate the device will lead to false data being produced. Once calibrated, the user will be able to practice for about two hours on full battery. During the user's practice session or after the session, the user can check the app to view the data gathered for each swing. Once the battery is drained or the user's practice session is finished, the user will need to recharge the device on a separate docking station connected to a wall outlet. Another constraint for our device is size and weight. Our device must be small enough to attach to the bottom of the handle on the cricket bat and light enough to not affect the overall feel of the bat and performance of the user.

3.3. System Description

- Power: The device will be powered by a lithium battery with a battery life of about two hours. The device will have a separate docking station that can be connected to a wall to recharge the battery.
- Sensors and Microcontroller: Our sensors(IMU) will encompass one 3-axis accelerometer and one 3-axis gyroscope. The device will gather the angular velocity and rate of change from the IMU and transfer data via bluetooth to be processed in a machine learning algorithm.
- App: The app will receive data from the MCU and send the data to an offline machine learning algorithm to calculate and predict the location of collision and the efficiency of the user's swing. Once the region of collision and efficiency is

determined, the app will display a histogram of the different collisions that occurred during the user's training session. It will also display the swing speed, swing angle, and efficiency of the user.

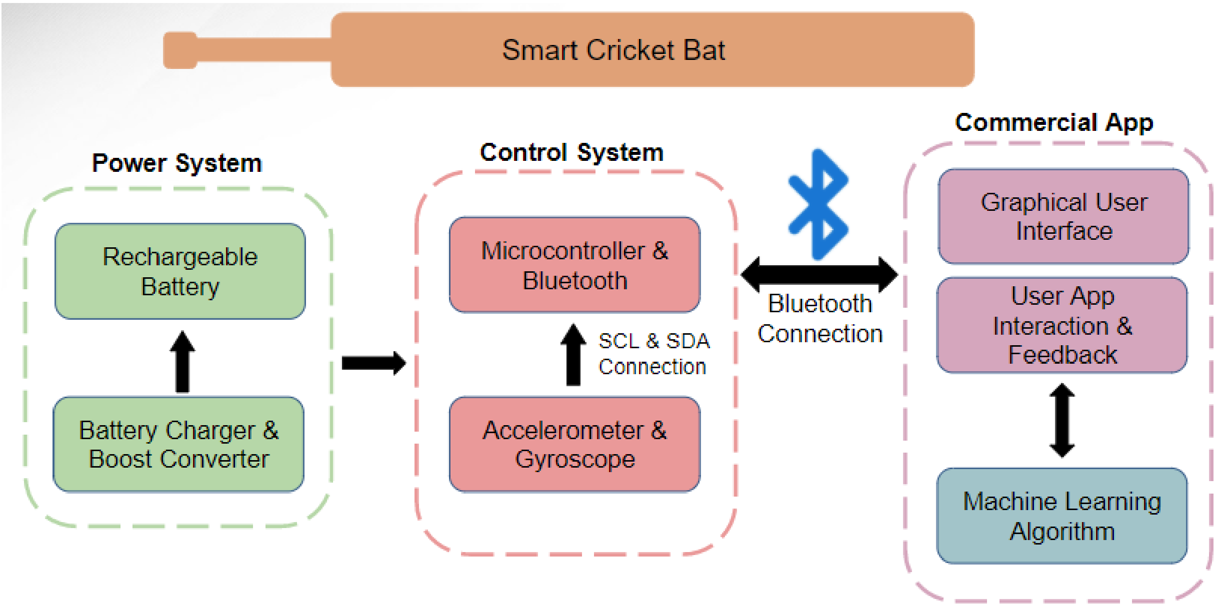


Fig. 2: Subsystem block diagram

3.4. Modes of Operations

The device will have two modes of operation:

- <u>Calibration Mode</u>: Before the user can use the device, the device needs to be calibrated, since cricket bats come in different lengths and weight. The user will hit the bat in different regions while it is stationary and no one is holding it. After each hit the user will input the region they stuck on the app. This action will be repeated once per region on the bat. The bat will be divided into 15 regions. View Fig.1 in List of Figures to view the regions of the cricket bat.
- <u>Practice Mode:</u> Once the device is calibrated, the user is now able to begin their training session. The device will have a two hour battery life. During the training session, the microprocessor will send data gathered from the IMU to the app to be sent to an offline machine learning algorithm to determine the location of collision and the efficiency of the user. The app will then display a histogram to show the collision location of each swing, the speed of the swing, swing angle, and user's efficiency.

3.5. Users

The main users will be cricket players trying to improve their swing. Their experience with cricket will range from grade-school level to professional players. The necessary skill level to operate our device is knowing how to use a phone app and read a histogram.

3.6. Support

The user will be provided with a user's manual to know how to properly use the device. The user's manual will encompass how to calibrate the device to be used in different sized cricket bats, the different functions of the app, ie. efficiency and swing speed, and how to charge the device. If any problems occur with the device, the user can send the device back to be fixed.

4. Scenario(s)

4.1. Indoor Cricket Batting Practice

The first scenario for the smart cricket bat is the user uses the device to practice their swing in an indoor training environment. In this scenario, the only potential damage that the device will encounter is the energy transfer of the ball and the bat colliding and potential sweat from the user. Our device will have to be able to work efficiently under these conditions

4.2. Outdoor Cricket Batting Practice

The second scenario is if the user wants to use the device to practice their swing in an outdoor training environment. In this scenario, many potential damages can occur. The device will need to not only withstand the conditions of the indoor scenario, but also take into account dirt, water, and extreme temperatures, for both cold and heat.

5. Analysis

5.1. Summary of Proposed Improvements

The device will offer a user-friendly app and system to allow them to increase the efficiency of their swing by seeing real time data and details on each swing. This device will be able to work on any size of cricket due to the required calibration mode prior to beginning a practice session.

5.2. Disadvantages and Limitations

- There is a limitation to the size of the system because it should be small enough to fit on to the bottom of the cricket bat handle.
- Unable to record video for each swings and analyze the video
- The device cannot be too heavy to disrupt the feel of a natural cricket bat
- Every user hold the bat differently, which can cause inconsistencies in the data collection and calculations

5.3. Alternatives

- The sensors could be attached to the back of the cricket bat instead of on the handle if size is an issue. Although we opposed from using this design due to a high risk of damaging our sensors due to the energy transfer from the collision of the ball and the cricket bat
- Additional sensors can be added to improve the accuracy of the results.

5.4. Impact

- This project will help cricket players study and improve the user’s batting performance.
- If the device becomes as popular as cricket, it can cause an increase in the need for lithium batteries. This higher demand will then cause the mining of lithium and other raw metals to increase.
- Our device has the potential to reduce the need for cricket trainers.

Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

INTERFACE CONTROL DOCUMENT

REVISION – Draft
23 February 2022

Interface Control Document
for
Smart Cricket Bat

Prepared by:

Author Date

Approved by:

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/23/2022	Gavin Dahl		Draft Release

Table of Contents

List of Figures

1. Overview

In this ICD, the details for physical dimensions and weight will be provided in section 3. The sensing unit will be attached to the end of the bat, the details for attaching will also be provided. This unit does not have a cooling system. Temperature limits will be shown in section 4. This unit shall last for two hours of battery life after fully charged. The electrical details and thresholds for sensors and the whole system are listed in section 5. The details for communication between systems and the user control interface are provided in the last section.

2. References and Definitions

2.1. References

- Reference Device:
  - https://www.stancebeam.com/
- Standard for the Specification of IMU's:
  - https://standards.ieee.org/ieee/1780/5700/
- Cricket Bat Standard:
  - https://www.cricketequipmentusa.com/cricket-bats-specifications-recent-changes-to-the-law-52#:~:text=Length%20and%20width%20of%20the,than%2052%25%20of%20the%20bat.

2.2. Definitions

m	Meter
mm	Millimeter
mA	Milliamp
mAh	Milliamp-hour
mW	Milliwatt
V	Volt
MCU	Microcontroller
IMU	Inertial Measurement Unit (Accelerometer & Gyroscope)
LiPo	Lithium Polymer Battery
BLE	Bluetooth Low Energy
SCL	Serial Clock Line
SDA	Serial Data Line

3. Physical Interface

3.1. Weight

The goal is to keep the weight as low as possible, to not interfere with the players swing. The exact weight of the housing unit that will hold the components is still unknown, but ideally will be around 50 grams. The internal components of the device approximately add up to 40 grams. Weighing the device as a whole, including both the housing unit and the device internals, it will be less than 100 grams.

3.2. Dimensions

The mounting device's interior restriction is limited by the internal components of the device, which will be around 30mm x 35mm x 30mm. The exact dimensions of the housing unit for the device is still unknown, but basing them off the known restrictions for internals of the device, it should be approximately 35mm x 40mm x 35mm, with the goal being to make it a cylindrical mounting device that will have a diameter of about 60mm and height of 35mm.

3.2.1. Dimension of the power system

The power system itself, being composed of 1 Li-Po battery and 1 boost converter/recharging station, makes up approximately 33mm x 20mm x 14mm

3.2.2. Dimension of the control system

The control system is made up of 1 Beetle BLE MCU and 1 6-axis IMU sensor (containing a 3-axis gyroscope and a 3-axis accelerometer), totaling up to approximately 30mm x 35mm x 8mm.

3.3. Mounting Locations

The mounting location will be at the end of the cricket bat’s handle. It will use a kind of sleeve that is put over a little more handle and will securely lock together with the housing device for the control and power systems. If there are any complications with trying to mount the device in this location, the back up spot for mounting will be on the back of the bat at the other end of the handle.

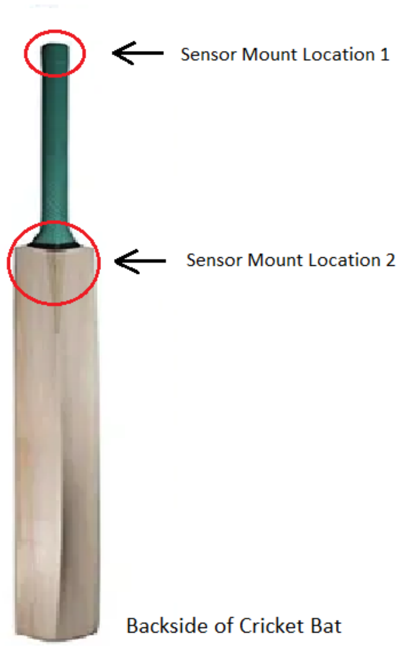


Figure 1: Mounting Locations on Bat

4. Thermal Interface

Since our device will be small enough to be attached to the bottom of a cricket bat, there will not be a need for fans or heat sinks for cooling. The only thermal interface our device will require is having vents or perforations on the main housing of the MCU and IMU's to allow airflow, as the MCU is the limiting factor and cannot run properly over 85°C (185°F). This will prevent overheating in case the user is using our device outdoors on a summer's day.

5. Electrical Interface

5.1. Primary Input Power

Power will be supplied by a rechargeable 3.7V lithium ion polymer battery with 150 mAh capacity. The voltage will be regulated and the boost converter will convert the voltage to 5V, which is within the acceptable range of input voltage levels for the MCU and the sensors.

5.2. Signal Interfaces

The signal interfaces are the input to the MCU from the IMU sensor containing a 3-axis gyroscope and a 3-axis accelerometer. The IMU sensor module uses SCL/SDA for communicating with the microcontroller.

5.3. User Control Interface

The interface with the user will be an android app which will display the data collected by the IMU sensor as well as the data for the location of collision with the ball and the efficiency of the user’s swing predicted by a machine learning algorithm.

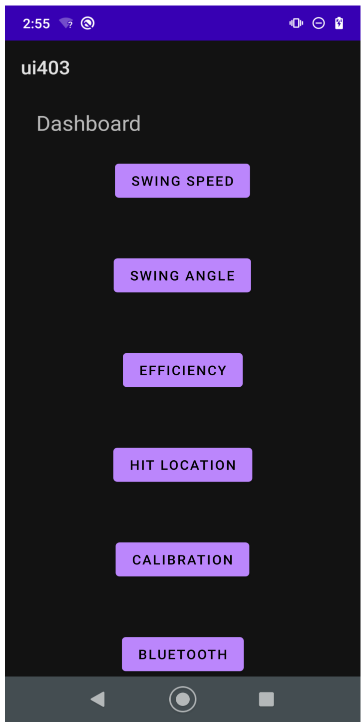
6. Communications / Device Interface Protocols

6.1. Wireless Communications (Bluetooth & Internet)

Our device will communicate with our app via bluetooth, which will have a connectivity range of about 50m. The app itself will also connect to our machine learning algorithm, that's stored in the cloud, via an internet connection.

6.2. GUI

All user interactions with our device will be controlled by an android app. These interactions will include calibration of our device and data output. The user will be able to navigate through different pages of the app via designated buttons on the home screen. A screen show of the buttons on the dashboard is shown below.



6.3. Device Peripheral Interface

The MCU and IMU will communicate with each other via a SCL and SDA connection. This connection allows the SCL to be the synchronized clock signal between the MCU and IMU for data transfer over the IC2 bus, and SDA is the data line that holds the actual data.

Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – Draft
23 February 2022

Functional System Requirements
FOR
Smart Cricket Bat

Prepared by:

Author Date

Approved by:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/23/2022	Gavin Dahl		Draft Release
1	4/22/2022	Gavin Dahl		Corrections

Table of Contents

List of Figures

1. Introduction

1.1. Purpose and Scope

Our purpose is to help users improve their cricket skill through a training session using our system. The sensing unit will be attached to the end of the bat. This unit shall last for two hours of battery life after fully charged. The sensors will receive data and transfer them via bluetooth to an android phone app. The ML algorithm shall calculate the necessary datas to the ones that need to be shown on the phone app like the hit location, swing angle and speed. The users shall be able to access these data through the phone app. The users will be able to know whether their hit was on the "sweet spot" or not as well as how efficient their swing was.

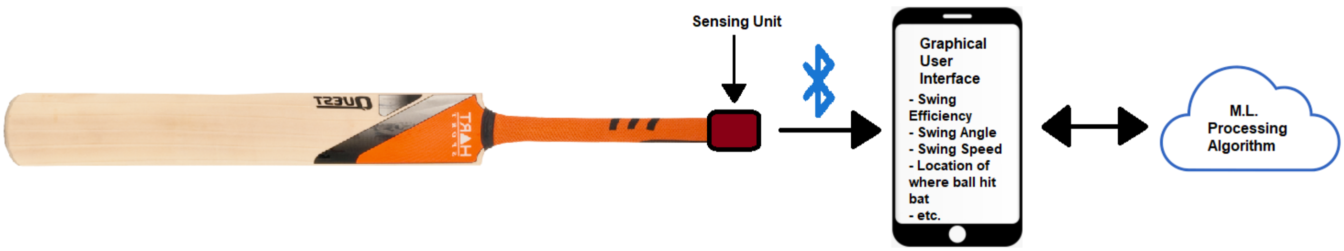


Figure 1. The Smart Cricket Bat Conceptual Image

1.2. Responsibility and Change Authority

Gavin Dahl has the responsibilities of making sure the team is sticking to the execution plan, completing milestones, and producing deliverables. He is also responsible for making the team confined to deadlines. The team's sponsor, Pranav Dhulipala, is the only one with the authority to make changes to our end goal requirements. If there is an unforeseen element that causes us to be unable to meet a requirement, Gavin and Pranav will discuss what will need to be done to change the requirement.

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
IEEE 1780-2022	2022-02-09	Standard for the Specification of Inertial Measurement Units
10.1109/ICIAFS.2008.4783997	2009-02-13	Design and Implementation of a Bluetooth based General Purpose Controlling Module
IEEE 802.15.1	2005-02-14	Standard for Telecommunications and Information Exchange Between Systems
IEEE 802.15.4	2020-05-06	Standard for Low-Rate Wireless Networks

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Link	Document Title
https://www.cricketequipmentusa.com/cricket-bats-specifications-recent-changes-to-the-law-52#:~:text=Length%20and%20width%20of%20the,than%2052%25%20of%20the%20bat	CRICKET BATS SPECIFICATIONS - RECENT CHANGES TO THE LAW
https://www.stancebeam.com/	Stance Beam

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

3.1. System Definition

In cricket, ball placement on the bat is very crucial to maximize the points you receive. To better improve a player's ball placement, we will design a device that can be attached to any cricket bat and predict the location of the collision between the bat and the ball. Our device will be powered by a rechargeable lithium battery with a battery life of 2 hours. Data will then be gathered via a 3-axis accelerometer and gyroscope. A MCU will then send the data via bluetooth to an Android app to send to an offline machine learning algorithm to predict the location of the collision. All results will then be displayed on the app.

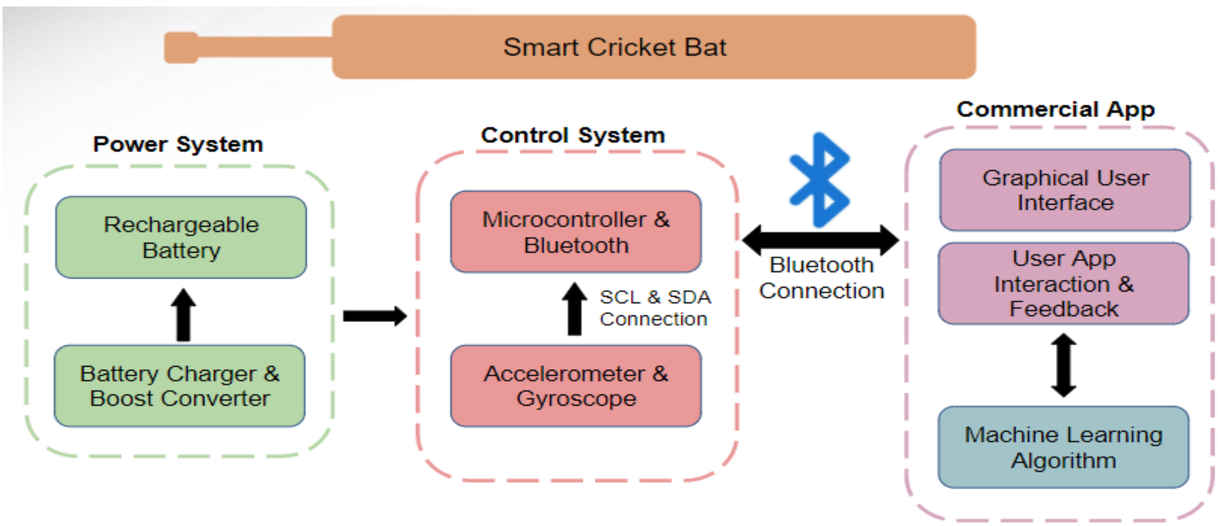


Figure 2. Block Diagram of System

The block diagram above shows the individual subsystems of our device. The subsystems include: Power System, Control System, and Commercial App. Our device will be powered by a lithium battery connected to a boost converter to deliver the necessary power requirement for the control system. We will design a separate docking station connected to the wall to recharge the battery. The control system encompasses a 3-axis accelerometer, a 3-axis gyroscope and a DFR0339 microprocessor. The angular momentum and rate of change data we measured with the gyroscope and accelerometer will then connect and be sent via bluetooth to the commercial app. The commercial app will then send the data to a machine learning algorithm via an internet connection to calculate collision location and efficiency. The results of our device will then be displayed on the easy to use app. Each output variable will have its own designated page to be viewed.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Calibration

Our device must be able to be calibrated to properly calculate results. The process of calibration will encompass hitting each region on the bat at least once, and telling the app which section you struck each time.

Rationale: Our device must be able to be used on any cricket bat, and each cricket bat has a different length, weight, and "sweet spot". Calibrating our device before usage will result in a higher certainty percentage.

3.2.1.2. Collision Location Prediction

The main function of our device will be able to predict the location of the collision between the cricket bat and the ball based on angular velocity and rate of change data measured by the IMU sensor. The certainty must be at least 90%. This will require a machine learning (ML) algorithm.

Rationale: Cricket players will use our device to improve their swing efficiency. The location of the collision will dictate how far and in which direction the ball will go. Depending on where the ball lands on the field is how many turns (points) a player can achieve. Therefore, knowing where the ball collides with the bat is crucial for the player to know.

3.2.1.3. Efficiency Calculation

Our device will be able to calculate how efficient the user is swinging the bat based on gathered data from previous swings. Efficiency is determined by how close the collision was to the "sweet spot" of the bat. The sweet spot of a cricket bat is located on the thickest section of the bat and is different for each cricket bat.

Rationale: Based on the efficiency the user will be able to tell how far off from the sweet spot their swing is.

3.2.1.4. Bluetooth Range

Our device will be able to communicate with the app within a range of 50m.

Rationale: The Beetle BLE MCU specifies it's bluetooth range is 50m. The user will be in one place during a practice swing session, therefore it does not need to cover the wide area of a cricket field

3.2.1.5. Easy to Use App/GUI

Our app will be easy navigable and easy to read the outputs

Rationale: The user might not have programming or data analysis skills so having an app that anyone can use and read is crucial for our device to be successful in the cricket market.

3.2.1.6. Battery Life

Our device will be powered by a lithium battery that must have a battery life of 2 hours and must be rechargeable.

Rationale: Battery life and rechargeability were requirements set by the customer.

3.2.2. Physical Characteristics

3.2.2.1. Mass

The mass of the sensing unit that mounts to the handle of the cricket bat should be no more than 100g.

Rationale: This is a requirement specified by our sponsor due to wanting our device to be as unobtrusive to the users swing as possible.

3.2.2.2. Volume Envelope

The volume envelope of the smart cricket bat's sensor unit shall be less than or equal to 35mm in height, 60mm in outer diameter, and 56mm in inner diameter,

Rationale: This is a requirement specified by our sponsor due to wanting our device to be as unobtrusive to the users swing as possible.

3.2.2.3. Mounting

The mounting device will attach to the bottom of the cricket bat's handle and will have a locking mechanism with the sensor unit to give it a secure fit.

Rationale: Swinging the bat and hitting a ball vibrates the bat a lot, we want to avoid a scenario where the sensing unit can easily disconnect from the bat.

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

a. The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not damage the smart cricket bat, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

b. No sequence of command shall damage the smart cricket bat, reduce its life expectancy, or cause any malfunction.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.1.1 Power Consumption

The maximum peak power of the system shall not exceed 300 mW.

Rationale: This requirement is to ensure that enough power is being supplied to each of the components for at least 2 hours when the battery is fully charged.

3.2.3.1.2 Input Voltage Level

The input voltage level for the MCU and the IMU shall be +5 V.

Rationale: The microcontroller's operating voltage is 5V and the IMU input voltage can be between 3.3V to 5V.

3.2.3.1.3 External Commands

The smart cricket bat shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low level electrical to the high-level packet format.

3.2.3.2. Outputs

3.2.3.2.1 Data Output via App

Our app will be able to output data gathered from the accelerometer and gyroscope and the calculated results from the ML algorithm. The outputs will include: swing speed, swing angle, collision location, and efficiency. Each output will have its own page the user can navigate to via buttons on the home screen.

Rationale: By having an app that can constantly be gathering and outputting data in an easy to read manner, it eliminates the need for the user to have tech. and data analysis skills. Without the app, users will have to download the data onto their computer to view their results.

3.2.3.3. Connectors

The smart cricket bat will use a microUSB to charge the sensor unit, and will also be able to get the sensor data via a microUSB in the case of bluetooth not working.

Rationale: Most efficient way to make the device is to make it rechargeable, so that it can be as small and light as possible.

3.2.3.4. Wiring

The smart cricket bat will use a PCB to connect all the internal components (MCU, IMU, Li-Po Battery, Boost Convertor/Charger) of the sensor unit.

Rationale: Normal wiring will most likely not be secure or robust enough for the amount of shock that the device will encounter.

3.2.4. Environmental Requirements

Our device will be adequate for indoor and outdoor use.

Rationale: Although our device will not be used during a cricket match, a user might be holding their practice session indoors or outdoors.

3.2.4.1. Pressure (Altitude)

Our device will be designed to operate in 1 atm pressure.

3.2.4.2. Temperature

Our device will be able to operate between temperature ranges of 0°C to 85°C.

Rationale: Since our device can be used outdoors, temperatures during the summer in Texas are between 97℉ - 105℉.

3.2.4.3. External Contamination

Our device will be housed in a casing to protect the components from any dirt or foreign objects.

Rationale: Since our device can be used outdoors, it must be protected from any materials that can cause damage to the components.

3.2.4.4. Rain

Our device will not be able to be used during the rain, but it should be waterproof enough to protect the components from the sweat of the user. The device will operate as intended within the humidity range of 0 to 80%, and should have the possibility of correct operation through 90 to 100%.

3.2.5. Failure Propagation

The smart cricket bat shall not allow propagation of faults beyond the app's interface.

3.2.5.1. Failure Detection, Isolation, and Recovery (FDIR)

3.2.5.1.1 Communication Test

The Smart Cricket Bat System shall provide users a notification if the user is using the system out of range or the internet/bluetooth connection is unstable at this moment.

3.2.5.1.1.1 BIT Critical Fault Detection

The BIT shall be able to detect a critical fault in the smart cricket bat's sensing unit, machine learning algorithm, or consumer app, 95 percent of the time.

Rationale: This is to provide the best user experience we can, letting the user know that we know there is an error and the system is working to fix it.

3.2.5.1.1.2 BIT False Alarms

The BIT shall have a false alarm rate of less than 5 percent.

Rationale: Due to some of the possible erroneous uses of the cricket bat, it might interpret an action of the user as an error, when in reality it is not.

3.2.5.1.1.3 BIT Log

The BIT shall save the results of each test to a log that shall be stored in the machine learning system for retrieval and will be used to improve common errors and overall quality of life of the user experience.

Rationale: Our device is very user focused, so providing the best user experience is of high priority.

3.2.5.1.2 Isolation and Recovery

The smart cricket bat should provide for fault isolation and recovery by enabling subsystems to be disabled based upon the result of the BIT.

Rationale: If there are any errors, we want to turn the device off and let the user know to stop using it so they aren't wasting energy on swings that will not be analyzed.

4. Support Requirements

The user will be provided with a user’s manual to know how to properly use the device. The user’s manual will encompass how to calibrate the device to be used in different sized cricket bats, the different functions of the app, ie. efficiency and swing speed, and how to charge the device. If any problems occur with the device, the user can send the device back to be fixed.

Appendix A: Acronyms and Abbreviations

m	Meter
mm	Millimeter
mA	Milliamp
mAh	Milliamp-hour
mW	Milliwatt
V	Volt
MCU	Microcontroller
IMU	Inertial Measurement Unit (Accelerometer & Gyroscope)
PCB	Printed Circuit Board
LiPo	Lithium Polymer Battery
BLE	Bluetooth Low Energy
GUI	Graphical User Interface
BIT	Built-in-test

Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

Execution Plan

Execution Plan for the Smart Cricket Bat

	1/24/2022	1/31/2022	2/7/2022	2/14/2022	2/21/2022	2/28/2022	3/7/2022	3/14/2022	3/21/2022	3/28/2022	4/4/2022	4/11/2022	4/18/2022	4/25/2022	4/30/2022
Midterm Presetation															
IMU and MCU Selection															
IMU and MCU Integration															
Battery and Booster/Charger Parts selection															
Charging Circuit															
Control/Power Integration															
GUI design/outline															
GUI button function integration															
G.U.I/Control System Itegration via Bluetooth															
ML algorithm researching and implementing															
Status Update															
Perf Board Creation															
Sensing Housing Unit Design															
Sending Data from Android Device															
Battery and Boost Integration															
Boost Converter PCB Design															
Machine Learning Training															
Sensing Housing Unit Manufacturing															
Sensing Unit Handle Mount Design															
Data Output															
Sensing Unit Handle Mount Manufacturing															
MCU/IMU PCB Design															
System Validation															
Final Design Presentation															
Final System Demo															
Final Report															

Legend

- Not Started
- In Progress
- Completed
- Behind Schedule

- Pablo Barron
- Jiakai Hu
- Gavin Dahl
- Nolapat Pipitvitayakul

Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

VALIDATION PLAN

Validation Plan for the Smart Cricket Bat

Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
Sending Data via Android device	The app should be able to send data (input for ML Algorithm)	Upload the app to a simulated android phone and test by outputting data to a localized device to test data sending	IN PROGRESS: App is able to send data to a device via an input stream	Pablo Barron
ML Algorithm Precision	The ML algorithm provides precise results of output data within acceptable error range.	Use all data gathered to test the training and testing accuracy.	TESTED: Training accuracy around 98% and Testing accuracy around 13%	Jiakai Hu
Communication Range	Communication between the sensing unit and app stays active for a distance of up to 100ft.	Test normal functionality of the smart cricket bat's operations at 5ft intervals ranging from 0ft to 100ft.	IN PROGRESS: Bluetooth successfully connected in a radius of 10 feet	Gavin Dahl Pablo Barron
System Latency	User swing analysis deliver to user via the app in no greater than 10s after the user's bat has struck the ball.	Take 20 practice swings with the bat, timing the time from the ball's impact on the cricket bat till the analyzed swing data is available to user on app.	UNTESTED	Full Team
Wireless Connection Stability	Connection between sensing unit and smartphone app does not drop.	Sensing unit connected to smartphone app via bluetooth, set to default mode, and left to run for 1 hour. Connection is monitored via smart phone app.	IN PROGRESS	Gavin Dahl Pablo Barron
Full Range of Motion	Sensing unit can measure the angle of the bat at a full 360°.	Sensing unit is attached to pivoting arm on a protracter, the angle is tracked on both a piece of paper and in a text file, and then compared.	TESTED: Sensing unit tracks accurate degree of turn for all 3 axes	Gavin Dahl
Easy to Use GUI	The app is easily navigable to allow any person, regardless of technical skills, to use our device	Upload the app to a simulated android phone to view the clarity of the app	TESTED: App is able to run on physical device and does not crash	Pablo Barron

Operation Time	System operates continuously on battery power for a minimum of 2 hours.	Sensing unit is turned on, set to default mode, and left to run for 2 hours. Power is monitored via a digital multimeter.	TESTED: Fully charged battery connected to boost converter with load current of 50 mA can run for 2:15 hours	Nolapat Pipitvitayakul
Detection Range	Sensing unit can detect vibrations from at least 38in away when mounted on a bat.	Mount sensing unit on end of the cricket bat handle and hit the top of the bat 10 times to ensure full range.	TESTED	Gavin Dahl
Detection Accuracy	Sensing unit is able to detect a collision between ball and bat on any area of the cricket bat.	Mount sensing unit on cricket bat, measure data from hits in a variety of areas on the bat (at least 15) until it is confirmed there are no dead areas.	TESTED	Gavin Dahl
Detection Sensitivity	Sensors are able to detect degrees of motion within 1deg of change and is able to give changes in speed to 1 decimal places.	Mount sensing unit to cricket bat, connect to PC via microUSB, use protractor to accurately gauge all 3 axes for IMU and compare, drop IMU from predetermined heights and compare values gotten with expected calculated results.	TESTED	Gavin Dahl
Ease of Use	System is easily attached to end of handle of the cricket bat, is easily connected to the app via bluetooth, and is easy to calibrated during first time start up calibrations. Whole process should take no more than 5 minutes.	Use stopwatch to measure how long it takes te user to mount device, pair to phone, and do the calibration setup.	IN PROGRESS	Full Team
Mass	Mass of combined control system (MCU and IMU sensors), power system (Li-Po Battery and boost converter), and our housing unit, will weigh no more than 100g.	Use digital scale to measure weight of combined unit.	UNTESTED	Nolapat Pipitvitayakul Gavin Dahl
Volume Envelope	The housing unit for the sensing device should a cylindrical shape, be no more than 60mm in	Measure inner and outer diameters and height of created housing unit for sensors.	UNTESTED	Nolapat Pipitvitayakul Gavin Dahl

	diameter, with an interior diameter of 56mm, and a height of 35mm.			
Mounting	The sensing unit is able to be mounted to the handle of any cricket bat and can lock securely into place.	Use the developed mounting device to mount the sensing unit onto end of the handle of the bat, and do various shack and shock test to confirm secure fit.	UNTESTED	Nolapat Pipitvitayakul Gavin Dahl
Input Voltage (MCU)	The input voltage for our Beetle BLE board shall be between 5V - 8V.	Use multimeter to validate input voltage level.	TESTED: Boost Converter Output is 5V +- 0.1	Nolapat Pipitvitayakul
Battery Charging Voltage and Current	The sensing unit has a 3.7V 150mAh Lo-Pi batteries as its power supply. These can be charged through a microUSB cable that can supply maximum 150mA of charge current with a 4.2V charge voltage.	Use multimeter to validate voltage levels and charge current levels.	TESTED: Charge voltage is 4.2 V and maximum charge current is 150 mA	Nolapat Pipitvitayakul
App Data Gathering via Bluetooth	The Android device should be able to receive data from our MCU via bluetooth	Upload the app to a simulated android phone and input different "dummy" data to see if the android device recieved the data	TESTED: Random Reading errors, but we are able to read data sent from the mcu	Pablo Barron
Thermal Resistance	The system should be able to operate in environments with temperatures ranging from 0°C to 85°C.	Use heating mechanism to raise temperature to 85°C and test systems functionality. Place system in cooling mechanism to lower temperature to 0°C and test systems.	UNTESTED	Gavin Dahl
Shock Tolerance	The IMU should be able to handle g shocks up to a max of 10,000g.	Test dropping IMU at differing heights and then use systems normal functionality to try to validate that IMU will still function after taking shocks more than 10,000	IN PROGRESS	Gavin Dahl

Performance on Execution Plan

The execution plan was executed completely. Although some tasks were actually performed somewhat out of order to accommodate the schedules of both teammates and the people operating the equipment used for the validation of the sensors, each task that was set forth in the execution plan was completed on time. The project was completed in full, with all of the objectives that were outlined at the beginning of the project demonstrated as required by the initial proposed solution.

Performance on Validation Plan

The validation plan is not complete yet, as we will need to validate a few more extremities once we move onto the next phase of creation. Data was collected on each of the different subsystems and their performance, in order to ensure that they will properly fulfill their individual function when integrated with the full system next semester. Although some difficulties were encountered, most notably the Bluetooth communication between the mobile device and the microcontroller, the different parts of the systems were tested to be reliable in the end, and the data acquired confirms this.

Smart Cricket Bat

Gavin Dahl, Jiakai Hu, Pablo Barron, Nolapat Pipitvitayakul

SUBSYSTEM REPORTS

REVISION – Original
20 April 2022

SUBSYSTEMS REPORT
FOR
Smart Cricket Bat

PREPARED BY:

____

Author Date

APPROVED BY:

____

Project Leader Date

____

John Lusher II, P.E. Date

____

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	4/20/2022	Gavin Dahl		Draft Release

Table of Contents

List of Tables

List of Figures

1. Introduction

The smart cricket bat will acquire data during the user's swing and give feedback to the user on how to further improve their swing. The system gathers data through the inertial measurement unit mounted at the handle of the bat, where it is transmitted to the consumer app via a microcontroller with a Bluetooth module. The data transferred to the app is then uploaded to the machine learning algorithm to be processed and then returns to the user the characteristics of the swing and what can be done to improve. The system is broken down into the power, control, app, and machine learning subsystems, each of which was designed and rigorously tested. Since each subsystem was validated to be working correctly and fulfilling all requirements, there is a clear path to integration for these subsystems into the full system specified in the Conops, FSR, and ICD.

2. Power Subsystem Report

2.1. *Subsystem Introduction*

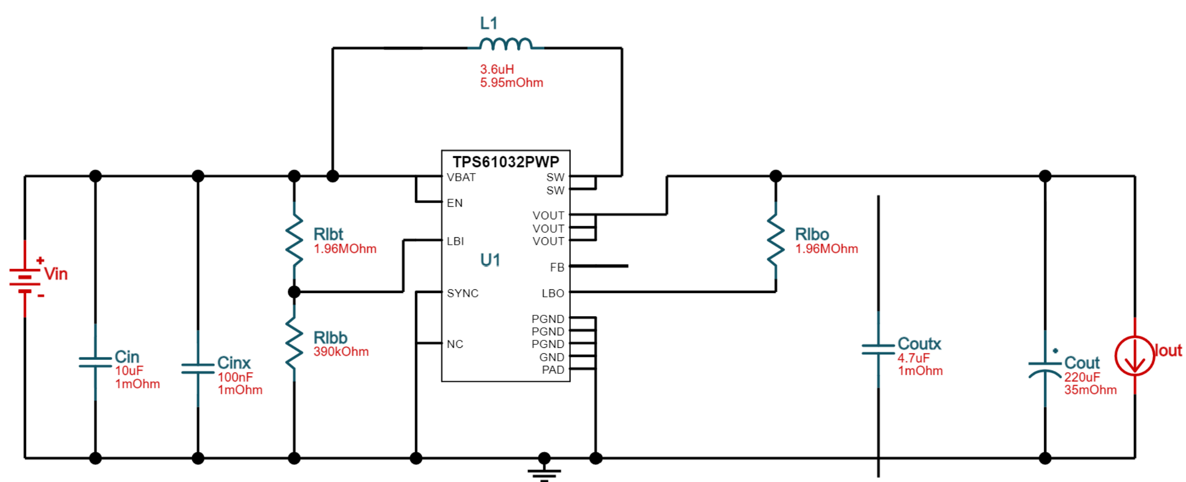
The power subsystem of the smart cricket bat consists of a lithium-ion polymer battery (LiPo), a DC-DC boost converter circuit, and a battery charger. A LiPo battery is rechargeable and can be charged with a battery charger circuit. A boost converter is used to raise the voltage from the LiPo battery in order to supply the power to the microcontroller which is part of the control subsystem. The boost converter was tested to confirm its stability and to validate that it performed correctly according to the design.

2.2. *Subsystem Details*

The source of power for the smart cricket bat is a 3.7V LiPo battery that has a capacity of 150 mAh. The reason for choosing LiPo battery as a power source is because it is rechargeable, small, and lightweight. These features are important due to the size and weight requirements of the smart cricket bat. The battery output ranges from 3.0V to 4.2V with a nominal voltage of 3.7V. When fully charged, the battery outputs 4.2V, and it will be completely cut out when the voltage goes below 3.0V.

A 5V DC-DC boost converter circuit was designed to raise the voltage levels of the battery that ranges from 3.0V to 4.2V to a 5V voltage level in order to power the microcontroller in the control subsystem. A boost converter circuit was designed based on the requirements using TI WEBENCH Power Designer. The chip used for the boost converter was TPS61032 from Texas Instruments. Figure 1 shows the schematic of a 5V DC-DC boost converter circuit.

Figure 1: Schematic of Boost Converter Circuit



A boost converter circuit was first built on a breadboard by connecting all the components according to the schematic and the output voltage was tested to confirm that it performed according to the design. The circuit was then moved from breadboard to a perfboard for a more robust and stable design after which it is tested to validate the load and line regulation of the boost converter circuit.

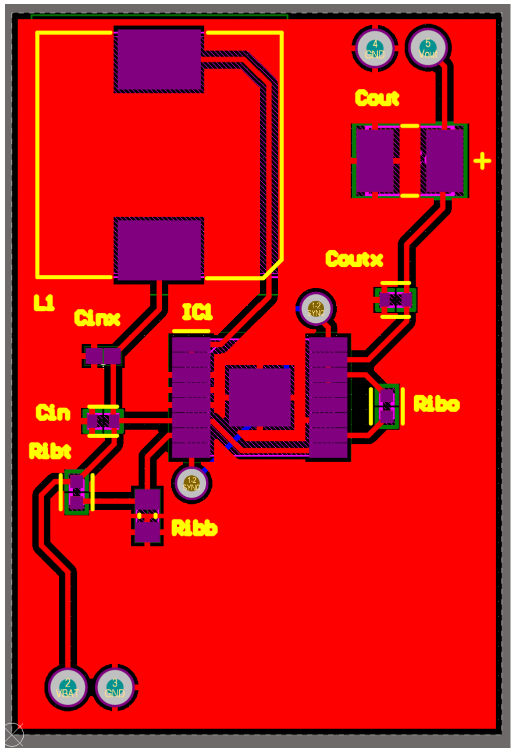
The SparkFun LiPo Charger is currently what is used as a LiPo battery charger where the battery can be charged with a micro USB cable as a charger input. It uses MCP73831 charge controller IC which employs a constant-current/constant-voltage charging method. The board is set to charge the battery at 500 mA by default, however, the LiPo battery currently used has a capacity of 150 mAh and has to be charged with a rate of 1C or less. This means that the charging current should be 150 mAh or less. So, a program resistor R_{PROG} = 2 kΩ on the board was desoldered and replaced with a resistor R_{PROG} = 6.67 kΩ to limit the charging current to 150 mA. The formula for the program resistor and charging current is found from the MCP73831 datasheet and are calculated using the following equation:

$$R_{PROG}(k\Omega) \ = \ 1000\,V\,/\,I_{REG}(mA)$$

So in order to limit the charging current of the battery to 150 mA, a program resistor should be 6.67 kΩ. The blue LED on the board turns on when the battery is charging and will turn off when the battery is fully charged.

A PCB for a boost converter circuit will replace a circuit on a perfboard in which all components on a PCB will be surface mount components. This will make the boost converter circuit to be even smaller so that it will be able to connect with the control subsystem in a housing unit that will be attached to the bottom of the cricket bat's handle. A PCB layout was designed using Altium and will later be sent to the PCB manufacturer for fabrication and assembly next semester. The PCB layout for the boost converter circuit is shown in Figure 2.

Figure 2: PCB Layout for Boost Converter Circuit



2.3. *Subsystem Validation*

The 5V DC-DC boost converter was tested for line regulation and load regulation. Line regulation is the ability of the power supply to maintain its specified output voltage over changes in input voltage. Load regulation is the ability of the power supply to maintain its specified output voltage over changes in the load. In this case, the output voltage of the boost converter should be able to maintain around 5V. Since a LiPo battery voltage ranges from 3V to 4.2V with a nominal voltage of 3.7V, the load and line regulation were tested for input voltages of 3V which is the minimum, 3.7V the nominal voltage, and 4.2V the maximum voltage of the battery. The load current was varied from 0 to 300 mA using the electronics load equipment and the output voltages were measured. Table 1 to Table 3 shown below are the data collected from the test, and each table has a corresponding plot.

Table 1: Load Regulation with Input Voltage Vin =3V

Vin (V)	Vout (V)	Iout (A)
3	4.956	0
3	4.963	0.037
3	4.973	0.057
3	4.99	0.077
3	4.961	0.1
3	4.948	0.12
3	4.934	0.14
3	4.924	0.16
3	4.914	0.18
3	4.907	0.2
3	4.895	0.22
3	4.887	0.24
3	4.877	0.26
3	4.868	0.28
3	4.855	0.3

Figure 3: Load Regulation of Boost Converter Circuit with Vin =3V

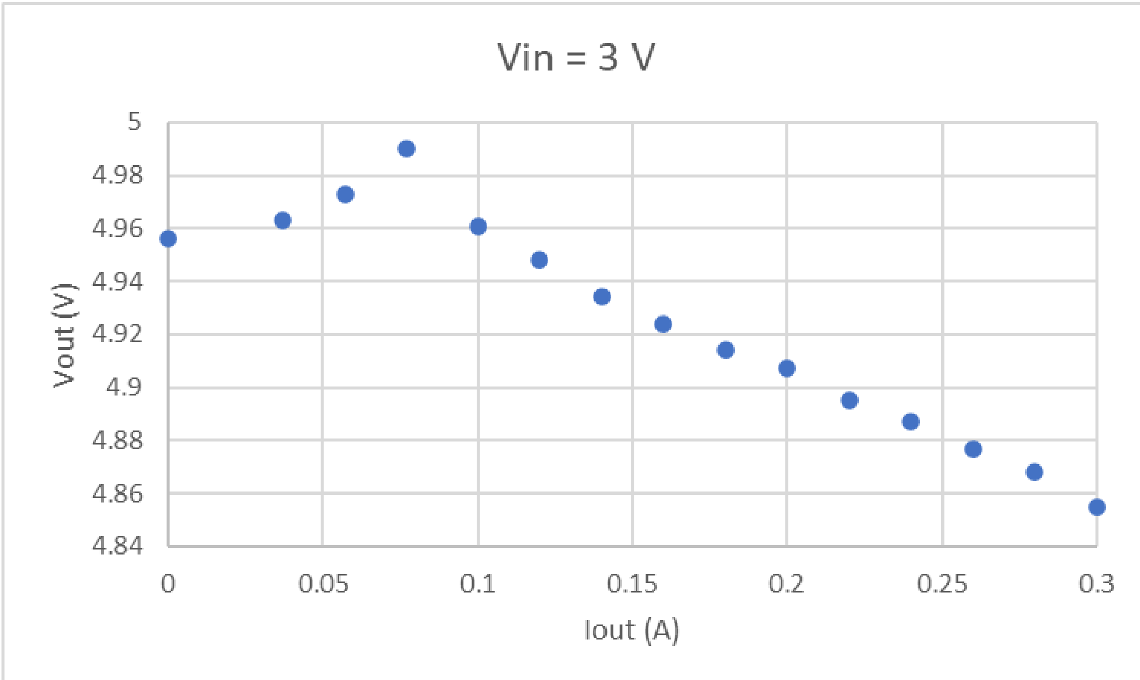


Table 2: Load Regulation with Input Voltage Vin =3.7V

Vin (V)	Vout (V)	Iout (A)
3.7	4.956	0
3.7	4.958	0.037
3.7	4.963	0.057
3.7	4.987	0.077
3.7	4.983	0.1
3.7	4.968	0.12
3.7	4.956	0.14
3.7	4.948	0.16
3.7	4.941	0.18
3.7	4.931	0.2
3.7	4.926	0.22
3.7	4.921	0.24
3.7	4.912	0.26
3.7	4.907	0.28
3.7	4.899	0.3

Figure 4: Load Regulation of Boost Converter Circuit with Vin =3.7V

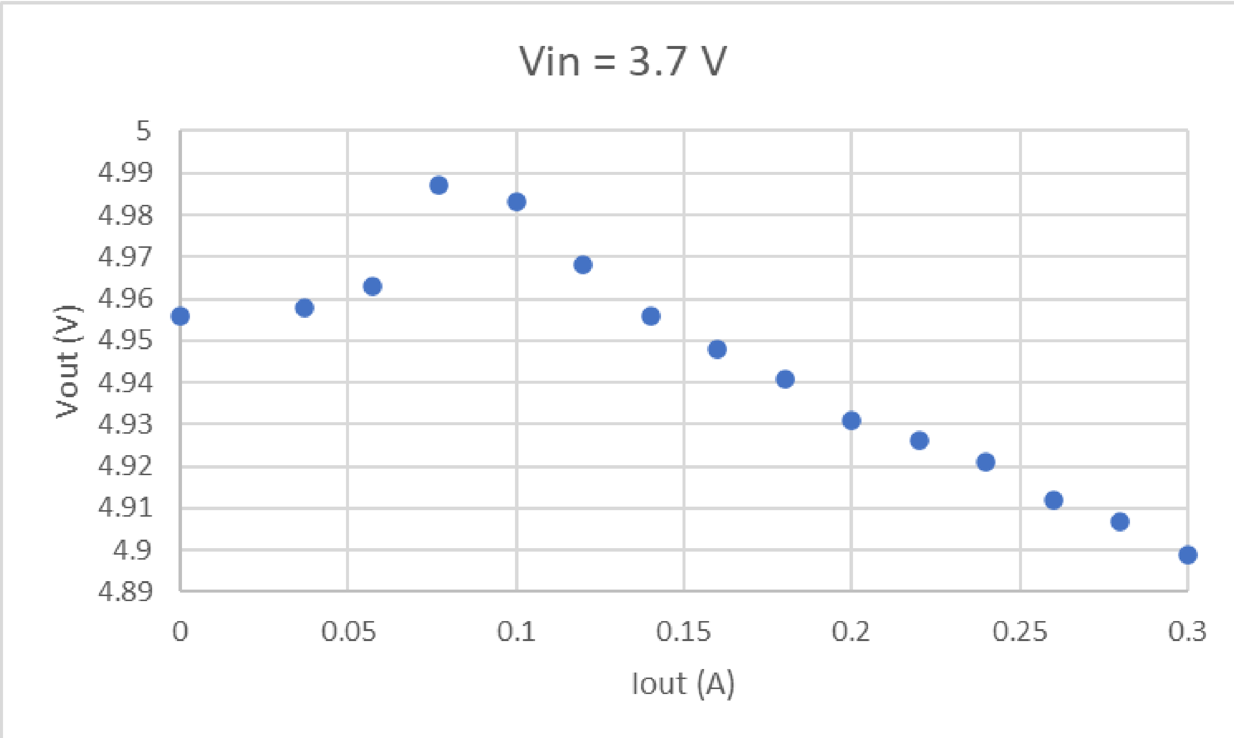
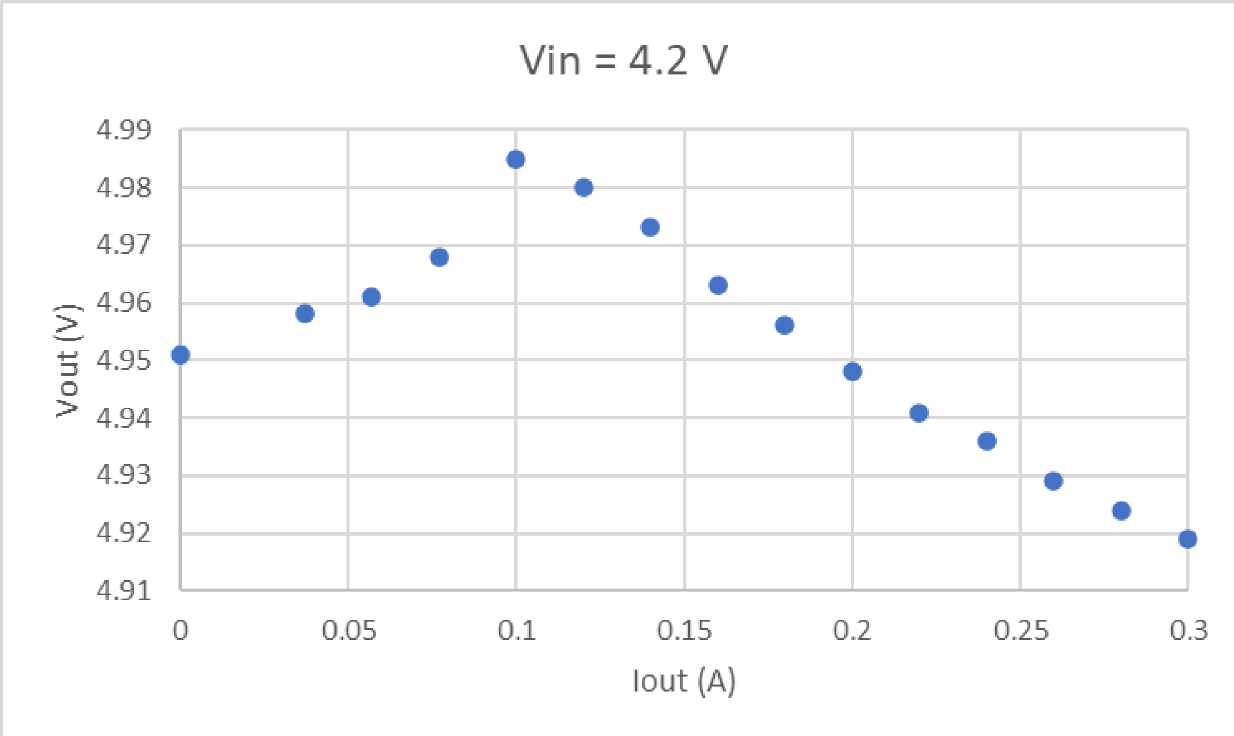


Table 3: Load Regulation with Input Voltage Vin =4.2V

Vin (V)	Vout (V)	Iout (A)
4.2	4.951	0
4.2	4.958	0.037
4.2	4.961	0.057
4.2	4.968	0.077
4.2	4.985	0.1
4.2	4.98	0.12
4.2	4.973	0.14
4.2	4.963	0.16
4.2	4.956	0.18
4.2	4.948	0.2
4.2	4.941	0.22
4.2	4.936	0.24
4.2	4.929	0.26
4.2	4.924	0.28
4.2	4.919	0.3

Figure 5: Load Regulation of Boost Converter Circuit with Vin =4.2V



As can be seen from the tables and figures above, the output voltages are very close to 5V even when the input voltages are varied from 3V to 4.2V and the load currents are varied from 0 to 300 mA. The lowest output voltage is 4.855V when Vin = 3 V and load current is at the maximum 300mA, which is still close to 5V. From the data collected, the load and line regulation were calculated. The load regulation is calculated using the following formula:

$$Load\ regulation\ = \frac{V_{(no-load)} - V_{(full-load)}}{V_{(full-load)}} \times\ 100$$

The line regulation is calculated using the following formula:

$$Line\ regulation\ = \frac{\Delta V_{out}}{\Delta V_{in}} \times\ 100$$

Using the above formula and the data above, the load and line regulation are

$$Load\ regulation\ = \frac{V_{(no-load)} - V_{(full-load)}}{V_{(full-load)}} \times\ 100\ = \frac{4.956-4.899}{4.899} \times\ 100\ =\ 1.1635\ \%$$

$$Line\ regulation\ = \frac{\Delta V_{out}}{\Delta V_{in}} \times\ 100\ = \frac{4.963\ -\ 4.961}{4.2\ -\ 3} \times\ 100\ =\ 0.167\%$$

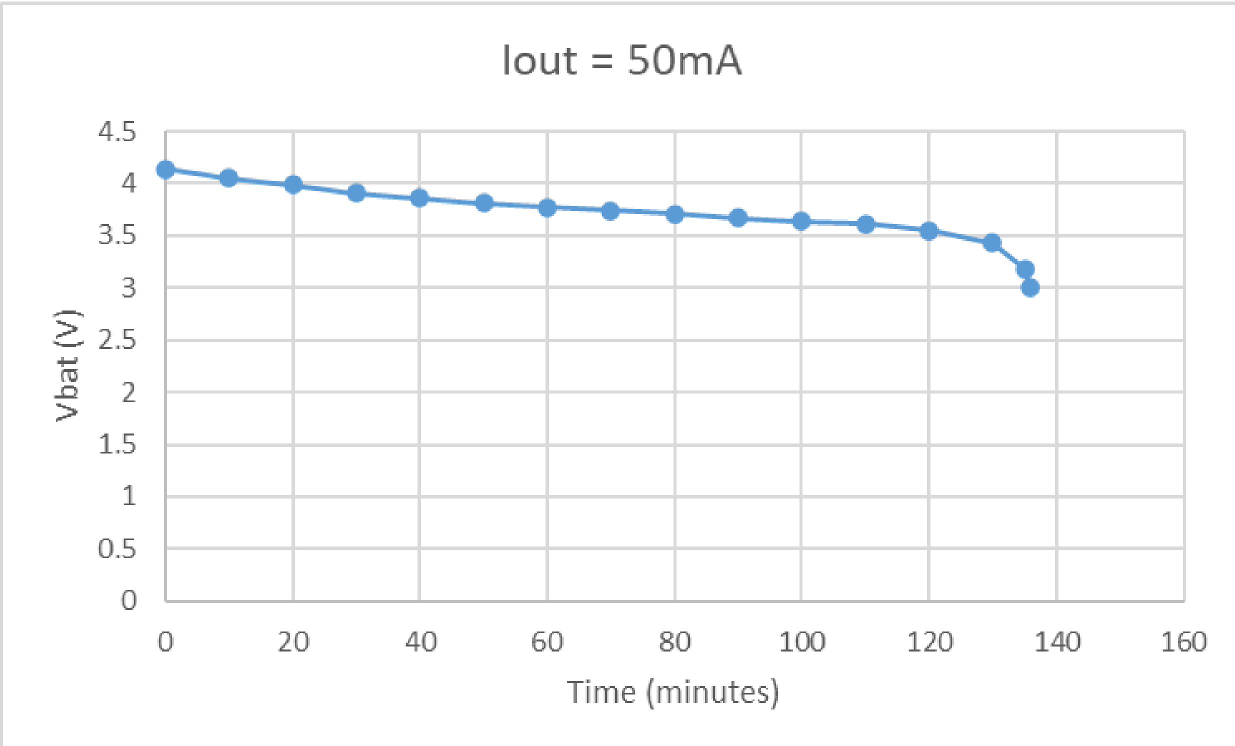
The values obtained from the load and line regulation calculations are small which means that the boost converter is well regulated. Thus, the ability for the boost converter to power the microcontroller was confirmed by the data collected and the calculations above, regardless of changes in the input voltage from 3V to 4.2V and load current from 0 to 300 mA.

In order to measure how long it can run, a LiPo battery was fully charged and connected to a boost converter circuit. The output of a boost converter is then connected to a load with current of about 50 mA. The load current of 50 mA is an estimate of the max current drawn for the entire system, including microcontroller, IMU, and bluetooth based on the information on the datasheets. The power was left running and the battery voltage was measured every 10 minutes until it went down to a cutoff voltage of 3V.

Table 4: Battery Voltage vs. Time with Load current =50mA

Time (min)	Vbat (V)
0	4.14
10	4.05
20	3.99
30	3.91
40	3.86
50	3.81
60	3.77
70	3.74
80	3.71
90	3.67
100	3.64
110	3.61
120	3.55
130	3.43
135	3.18
136	3

Figure 6: Battery Discharge Curve with Load current =50mA



Notice that the battery voltage went down from a maximum of about 4.2V to 3V in about 135 minutes or 2:15 hours when the load current is 50mA. So, the requirement that the system be able to run on a battery for at least 2 hours is validated.

2.4. *Subsystem Conclusion*

The DC-DC boost converter works as designed and was able to raise the voltage signal from a LiPo battery to 5V at the output of the converter. A battery was able to last for more than 2 hours as specified by the sponsor and can be charged with a LiPo charger. Overall, the power subsystem works as expected and reached the goal of this semester. Next step is to send a PCB layout to a manufacturer for fabrication and assembly of the converter and to integrate the power subsystem with the control subsystem in the next semester.

3. Control Subsystem Report

3.1. *Subsystem Introduction*

Since the purpose of the sensing unit is to relay swing data to the machine learning for accurate analysis, it is critical to confirm that the inertial measurement unit, and the control subsystem as a whole, operates correctly. To accomplish this, the microcontroller was programmed and was wired to establish a connection to the IMU. Once the two were interfacing, each of the two axes of the IMU, the 3-axis gyroscope and 3-axis accelerometer, were tested to validate that it performed as the manufacturer described.

3.2. *Microcontroller*

3.2.1. Overview

The hardware of the control system supports interfacing between the IMU sensor and the commercial user application. And as the bridge between the two, is essential in the data collection process, so it must run continuously to support the entire system.

3.2.2. Operation

A Bluno Beetle BLE microcontroller is currently what is used as the control system to deliver data from the IMU sensor to the app.The Bluno Beetle has a sufficient number of pins, supply voltages, and bluetooth modules to meet the requirements of the Smart Cricket Bat.

The Bluno Beetle BLE has 4 Digital I/O pins, 4 Analog Input pins, and a pair of Serial Clock Line and Serial Data Line pins. It is also able to provide a 5V output supply to power the IMU. This is more than enough to interface with the single IMU sensor as it only needs power, a pair of SDA and SCL pins, and a single digital pin for the interrupt. The Bluno Beetle can be powered on a range of 5 to 8 V, which will be supplied by the power subsystems boost converter.

The key feature of the microcontroller is to function as the bridge between the sensor data and the application that holds the machine learning algorithm, it must be able to maintain this connection for the duration of a training session. Bluetooth will be used to effectively do this, currently using standard bluetooth before trying to establish the low-energy bluetooth 4.0 module within the microcontroller. Code for this operation was written and successfully tested as part of the other subsystems.

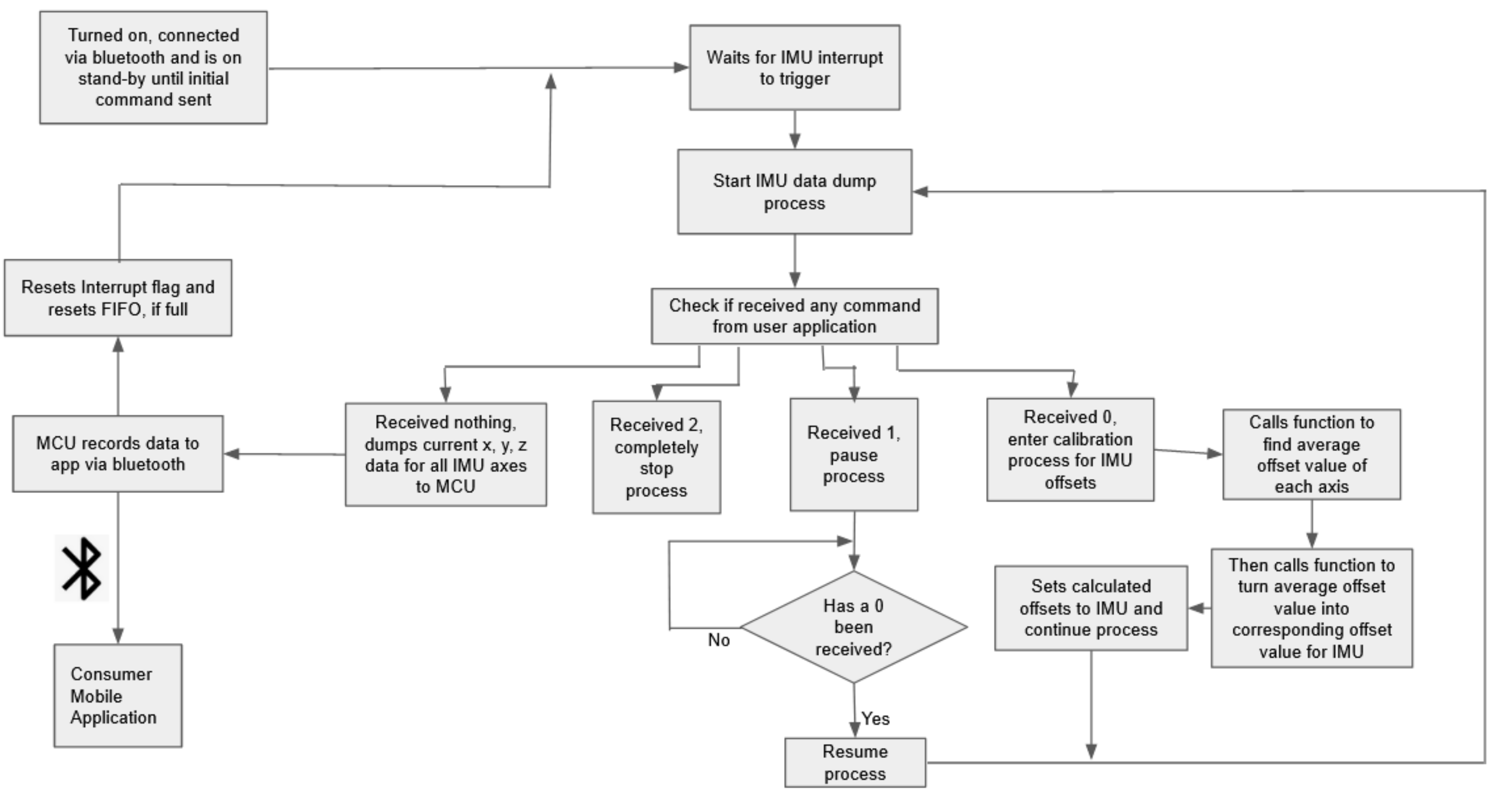
A PCB will later be added to replace the perfboard prototype and provide more secure connections between the power and control subsystems.

3.2.3. Validation

The control system was validated to meet all requirements. Since its primary function is to communicate with the application subsystem, it was validated by first linking and pairing with the manufacturers recommended application for bluetooth to confirm the low-energy bluetooth module onboard the MCU was operating as intended. Then to confirm the code written for the MCU works as intended with standard bluetooth, a dummy control system was made with a standard HC-05 module and connection was established and information was both sent and received. Integration will involve moving from standard bluetooth to low-energy bluetooth.

The flowchart in figure 7 shows the logical operation of how the microcontroller goes about gathering the IMU data, interpreting user input, and delivers the data to the consumer application.

Figure 7: Flow Diagram of Microcontroller Logic



Something not shown in the flowchart is the fact that for the IMU, the interrupt is constantly triggered after the first trigger, so within the code is a buffer step that checks for more data dumps from the IMU before the MCU checks the interrupt again allowing for quicker revolving of data. However, this on paper functions the exact same way as checking the interrupt each time, so it can be thought of the same way to avoid unnecessary confusion.

This logic was successfully executed on the Bluno Beetle with the validated IMU sensor results in turn validating the microcontroller. The only thing remaining unvalidated is the communication between the MCUs low-energy bluetooth module and the app, which will be done during the integration stage.

3.3. *Gyroscope*

3.2.1. Operation

The inertial measurement unit used is a gy-521 board that uses an MPU-6050. The MPU-6050 has both a gyroscope and accelerometer inside, this was chosen for two option, one was to save on space as the design must stay small and unobtrusive, and two, its best if the gyro and accelerometer axes are lined up and this is are to do with two separate devices. For the gyroscope, it has 4 full scale ranges, ±250, ±500, ±1000, ±2000 °/sec with the sensitivity being 131, 65.5, 32.8, 16.4 LSB/°/sec, respectively. For the purposes of the Smart Cricket Bat, the range will stay at the default range of ±250 °/sec, which has a noise rate of 0.005 mdps/rtHz.

The gyroscope uses an I2C interface, which allows the microcontroller to take the measurements in via the serial clock and serial data lines. This is the standard way I2C devices communicate, by transmitting data that are 9 bits long along these 2 lines.

3.2.2. Validation

The gyroscope’s angle accuracy was validated by attaching the gyroscope to the center of a protractor and rotating the protractor to the desired degree and comparing that to and confirming that the gyroscope's readings were accurate. This was done for each axis of the gyroscope and was done from angle ranges of 0-45°, 0-90°, 0-135°, 0-180°, 0-215°, 0-270°, 0-305°, 0-360°. Below are some of the graphs of the received gyro values as each axis is rotated from 0-180°, all 21 other angle graphs were also made but not shown here for space purposes.

Figure 8: X-Axis Gyroscope Validation

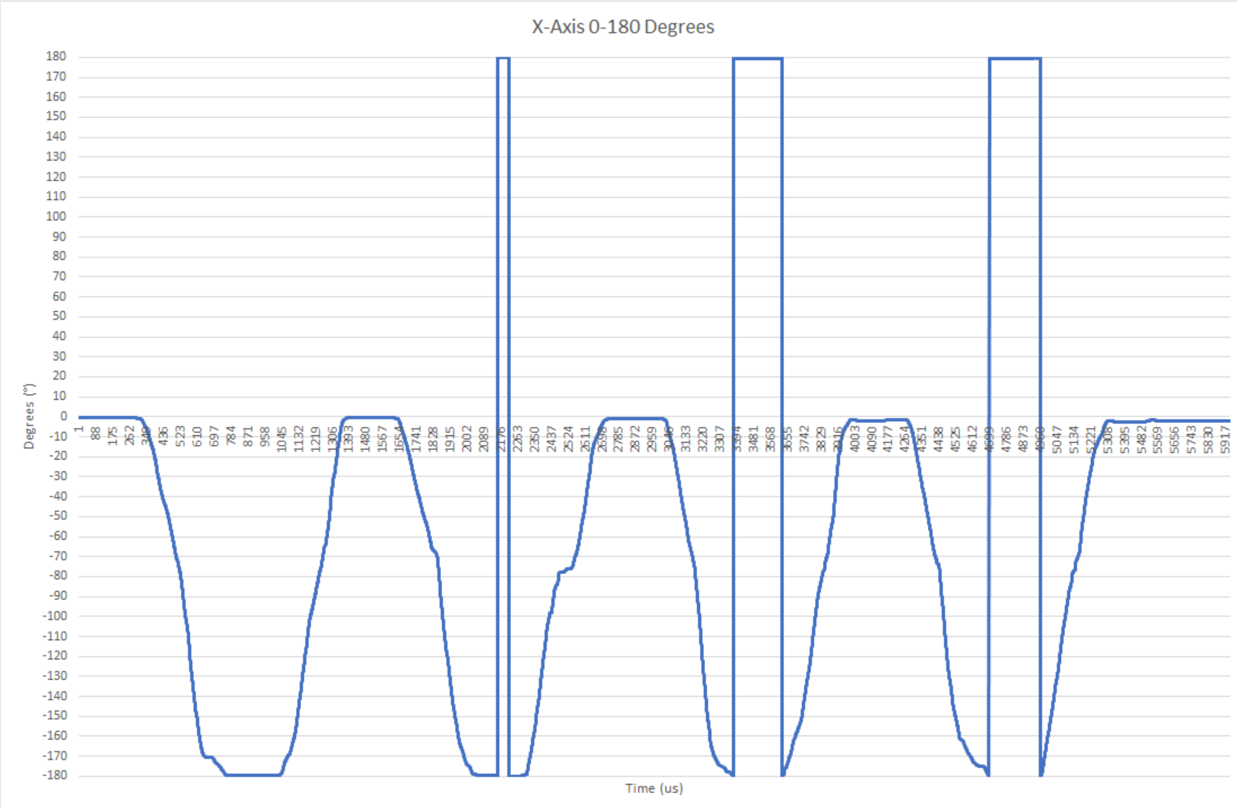


Figure 9: Y-Axis Gyroscope Validation

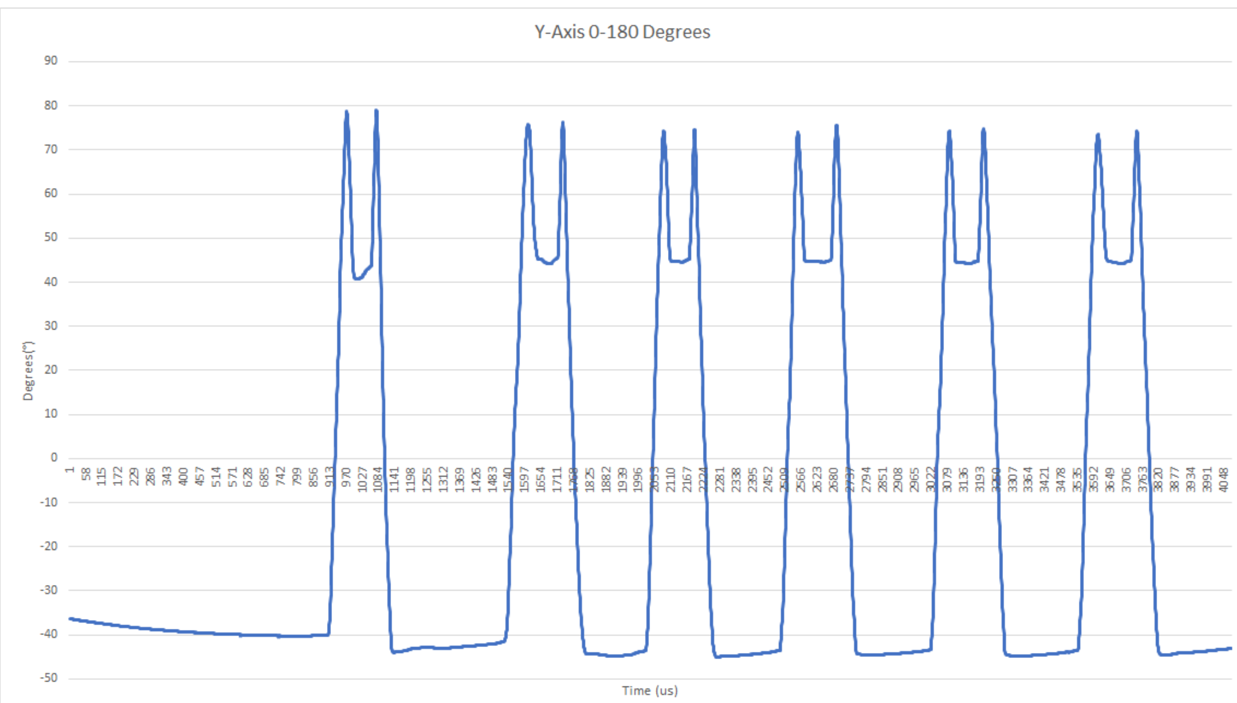
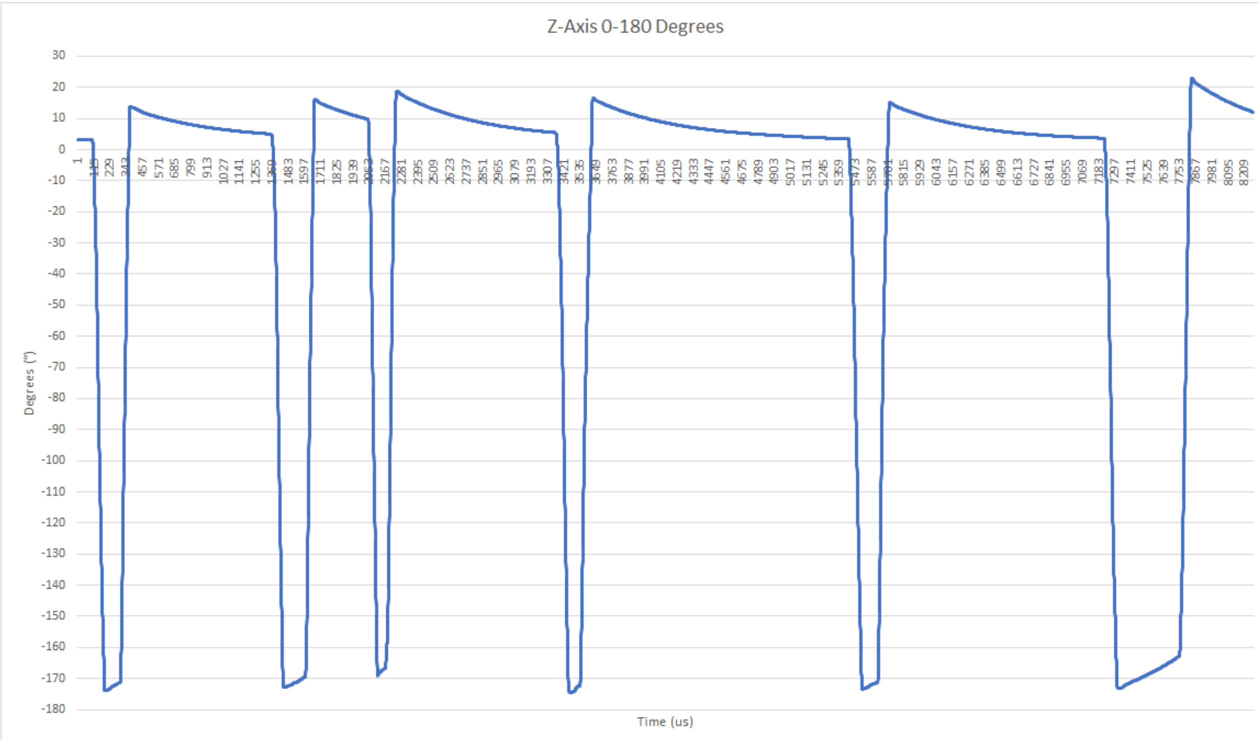


Figure 10: Z-Axis Gyroscope Validation



Notice, since the graphs for the y and z axes drift a bit more, which causes them to lose track of where they started, being a consistent 5° and 10° off on return, respectively. However, this won't be an issue because, one, we will primarily be using the x and z axes to find the angle of the bat on impact, which has the least drift. And two, since most swings won't start at 0° anyway, the machine learning algorithm will already be subtracting the starting degree from the finished degree to get the true angle of the bat.

Through extensive validation, the gyroscope within the IMU is confirmed to be working as intended and gives the accurate degrees of change for the system. However, replacing the gyroscope (and by extension the IMU) with one with more consistency, less tendency to drift, and overall less noise is being considered.

3.4. *Accelerometer*

3.4.1. Operation

The inertial measurement unit used is a gy-521 board that uses an MPU-6050. For the accelerometer, it has 4 full scale ranges, ±2, ±4, ±8, ±16 g with the sensitivity being 16384, 8192, 4096, 2048 LSB/g, respectively. The raw data received from the accelerometer is in terms of least significant bit per g (LSB/g) and must be divided by the sensitivity to get the data in terms of g's. For the purposes of the Smart Cricket Bat, the range will be the range of ±8 g, which has a noise rate of 0.005 mdps/rtHz. This range and sensitivity may be subject to change as we integrate the subsystems, it depends on what gets the best results out of the machine learning algorithm.

The accelerometer uses an I2C interface, which allows the microcontroller to take the measurements in via the serial clock and serial data lines. This is the standard way I2C devices communicate, by transmitting data that are 9 bits long along these 2 lines.

3.4.2. Validation

To validate the accelerometer, the device was dropped from 4 different heights ranging from 1 to 15 feet. The heights were restricted to a max of 15 feet because the bluetooth module cannot currently be used and the power system is not in a state to be dropped from so high, so the length of the microUSB being used to power and receive info from the device is its limiting factor. There were also problems with the microUSB's connection being unstable after the first couple of drops, so the data would sometimes stop right after the collision, specifically in the case of 15 feet. With this in mind, the following two kinds of graphs are produced from the acceleration of the x, y, and z axes when falling and hitting concrete from heights at 5, 10, and 15 feet. The first being the average magnitude of the acceleration of the entire unit during the fall and the other is the raw accelerations for each axis.

Figure 11: 5 Foot Fall Accelerometer Validation

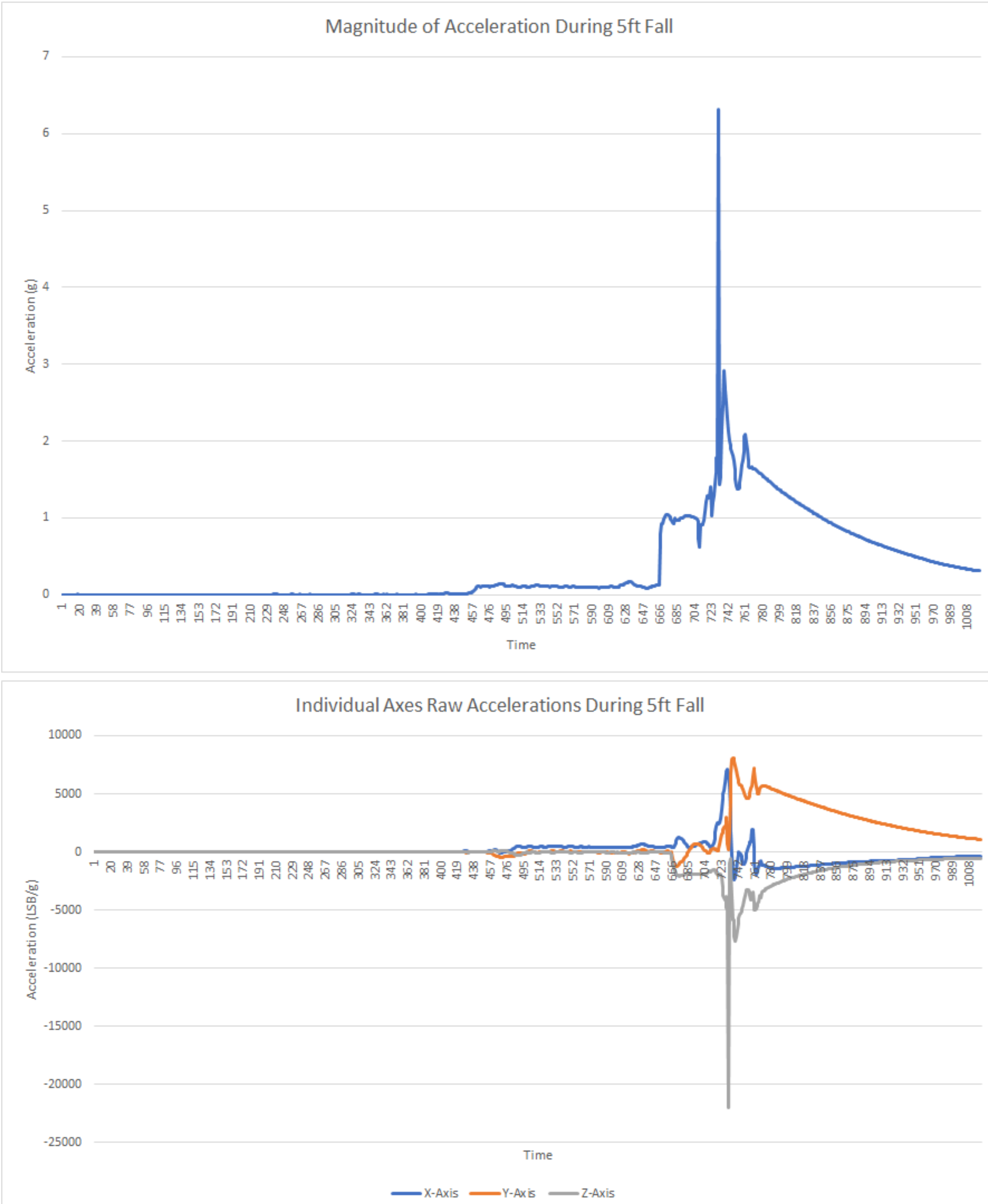
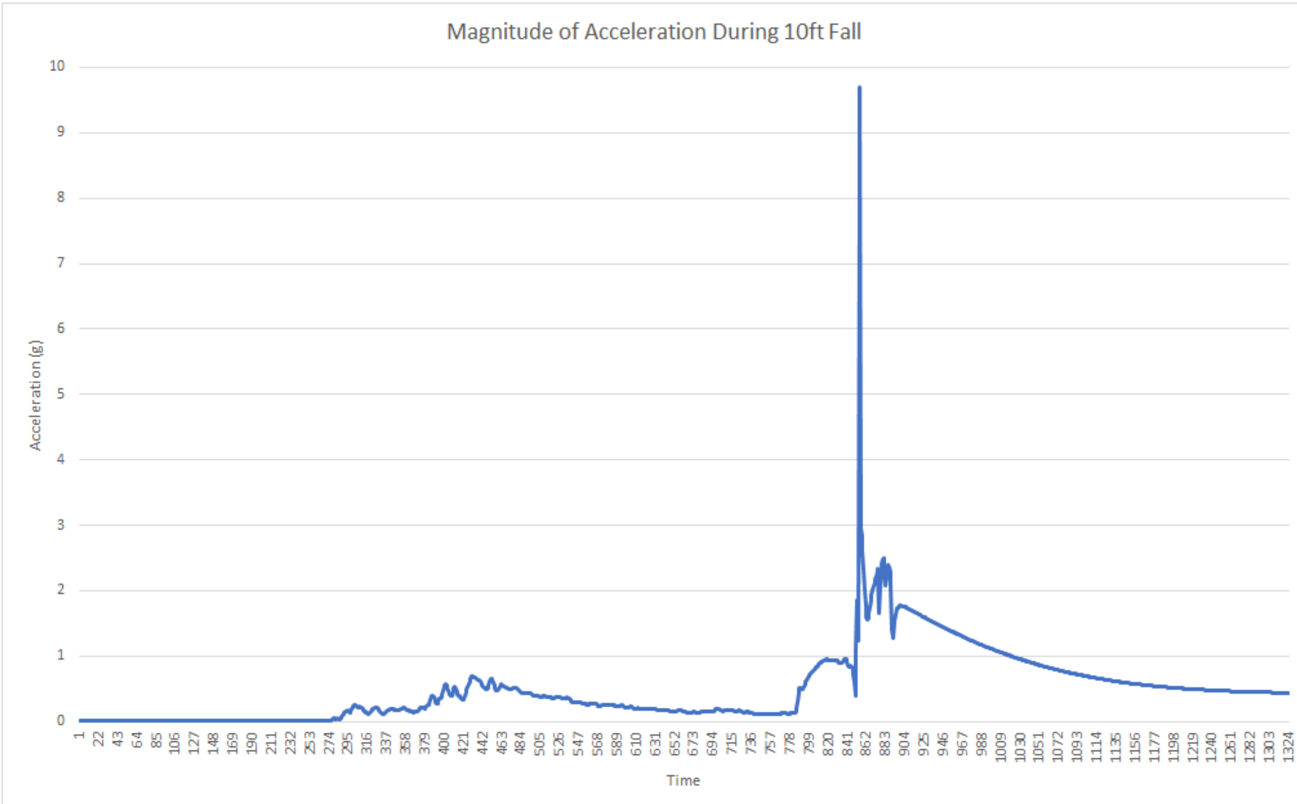


Figure 12: 10 Foot Fall Accelerometer Validation



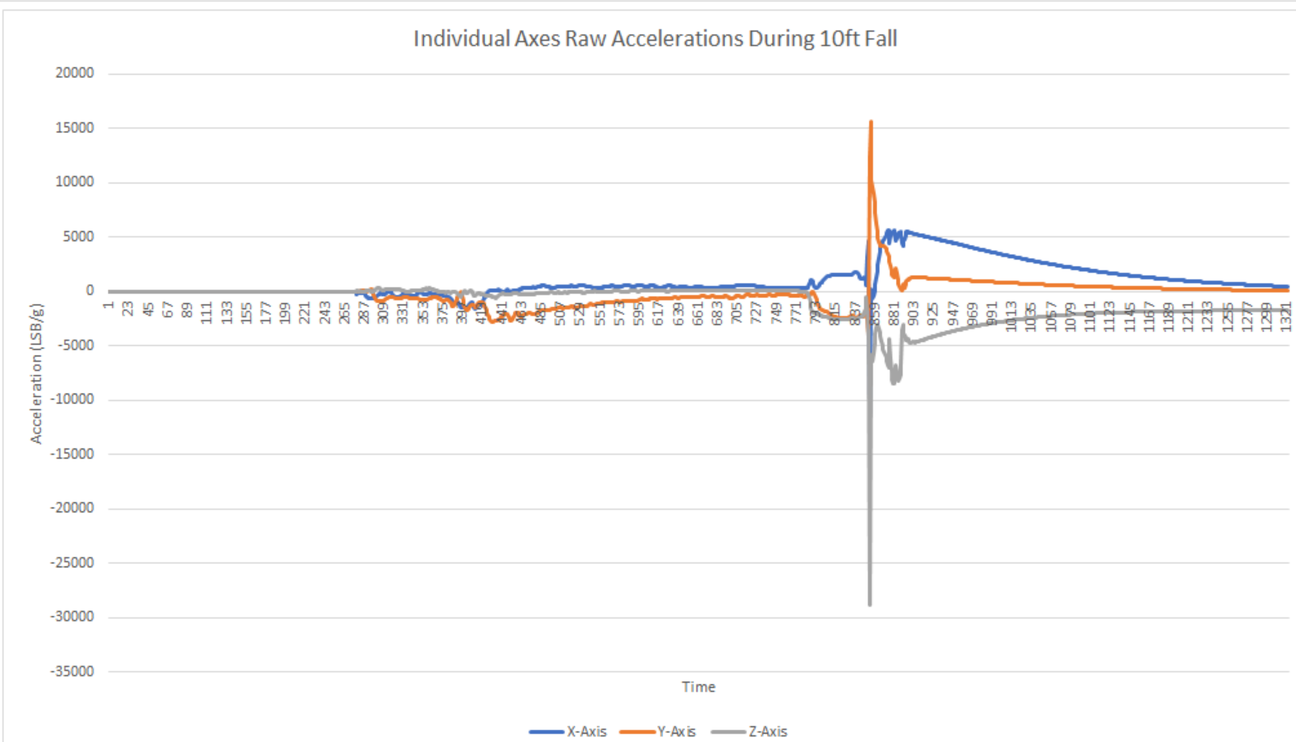
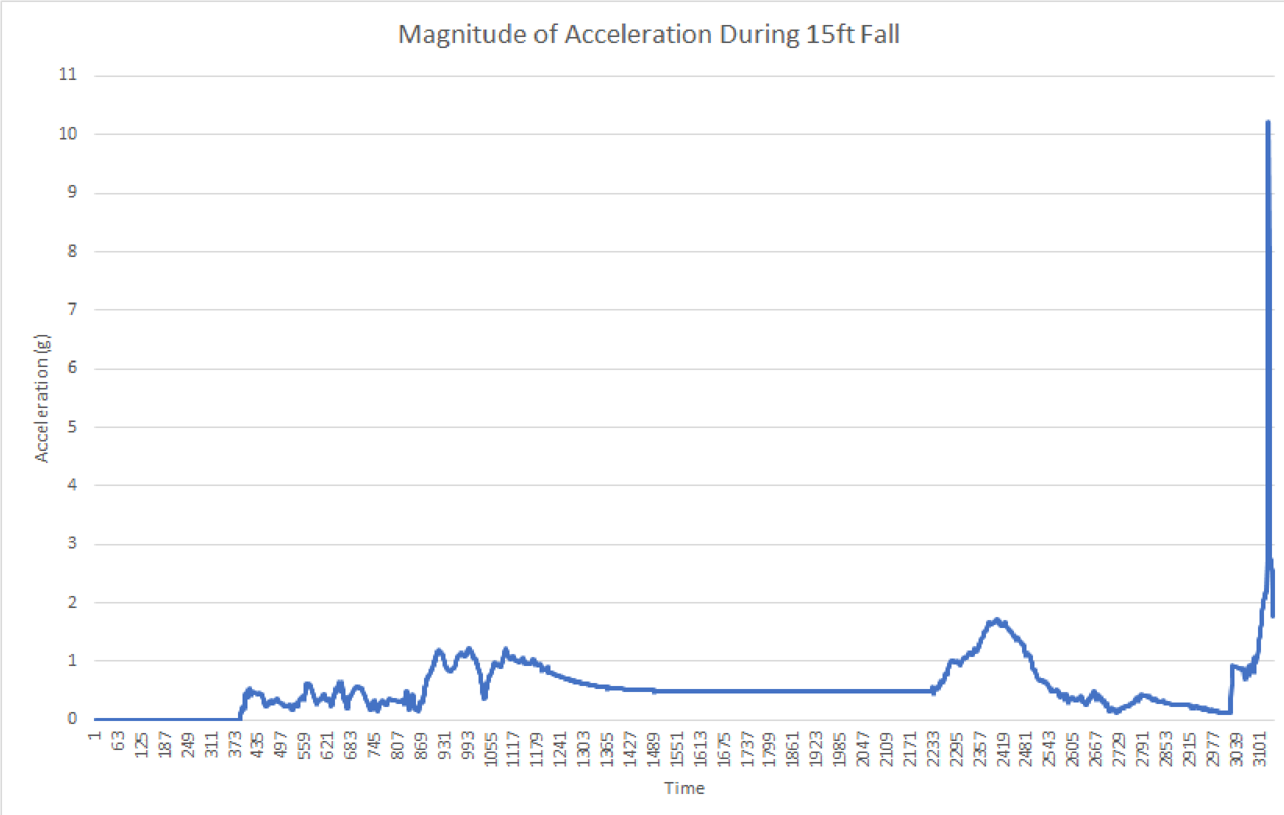
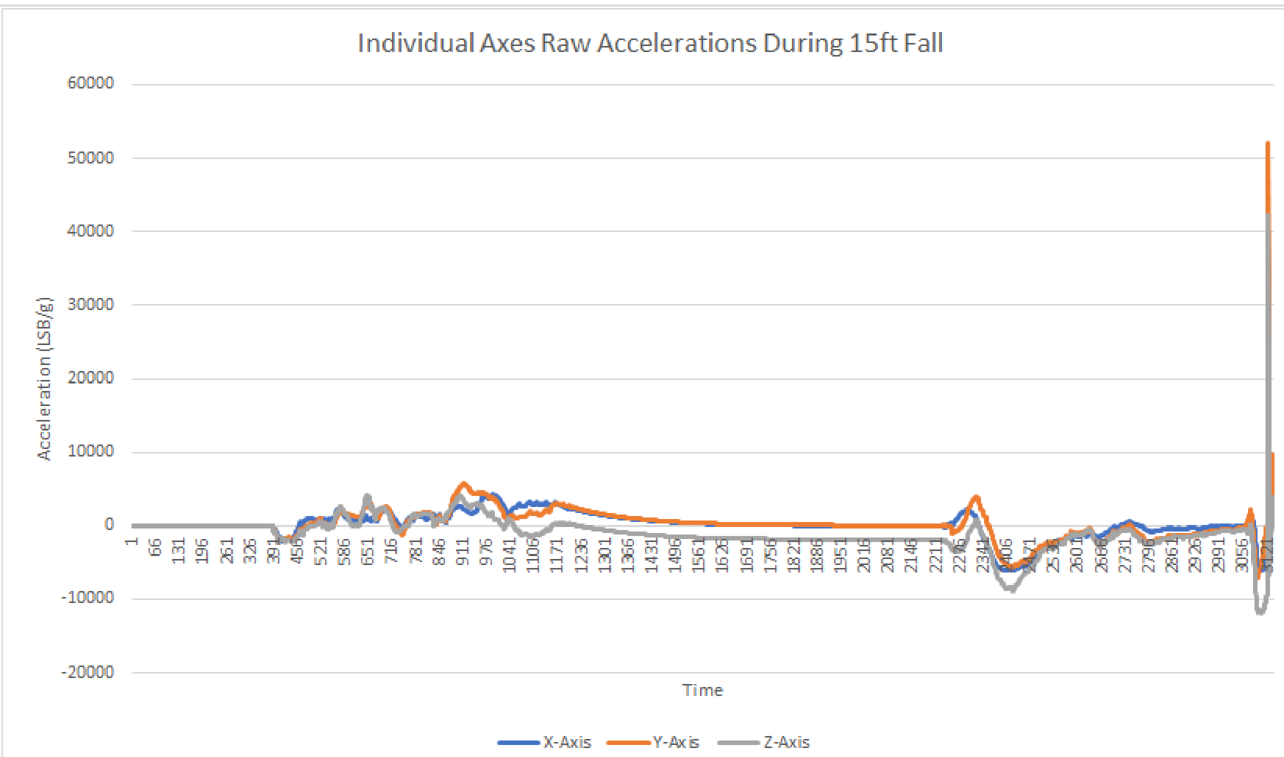


Figure 13: 15 Foot Fall Accelerometer Validation





Compare these values to what the expected acceleration during the collision should be, based on calculations done using an estimated collision time of 10ms and the assumption that the device loses about 75% of its acceleration after the collision. The values obtained are all about 10 times larger than the values received from the IMU, there could be a few reasons for this. One, the IMU automatically removes gravity from given values which may affect total gs, two, the calculations were done incorrectly leading them to be larger, or three, the IMU is giving wrong values. However, out of these options the most likely to affect the validation process is the removal of gravity, as it is unclear how this factors into the calculations, and could be the cause for the calculated accelerations being almost a factor of 10 larger.

Method Used to Calculate gs

$$Use\ Kinetic\ Energy\ to\ find\ velocity\ right\ before\ collison:$$

$$KE = \frac{1}{2}mv^2 \ -> \ v_1 = \sqrt{2gh} = \sqrt{2 * 9.81 * h}$$

$$Use\ velocity\ to\ find\ acceleration\ during\ collision,\ assume\ t = 10\ ms\ \&\ v_2 = \frac{v_1}{4}:$$

$$a = \frac{v_2 - v_1}{t} = \frac{\frac{\sqrt{2*9.81*h}}{4} + \sqrt{2*9.81*h}}{0.01}$$

$$Finally,\ convert\ acceleration\ from\ m/s^2\ to\ gs:$$

$$= a/9.81 \ -> \ gives\ acceleration\ during\ collision\ in\ terms\ of\ g$$

Table 5: Accelerometer Measured Vs Calculated Acceleration

Dropped Height	Calculated Acceleration	Measured Acceleration
5 ft	69.673 g	6.3214 g
10 ft	98.534 g	9.6987 g
15 ft	120.6804 g	10.223 g

After comparing the two data sets it can be inferred that, despite the values from the accelerometer being off by about a multiple of 10, the accelerometer gives consistent data that is representative/follows the trend of the overall changes in acceleration. This is more than passable for the purposes of the Smart Cricket Bat, as it only needs the relative gs enacted on the bat by the ball in a given section, relative to the other sections. However, replacing the accelerometer (much like the gyroscope) with one with more consistency, accuracy, and sensitivity is being considered.

3.5. *Housing Unit*

The housing unit consists of 2 elements, the mounting mechanism and the housing unit for the power system and control system. The current housing unit prototype is 3-D printed with PLA material and is 65.5 mm in length and 51mm in radius. The prototype, primarily being used to collect data for the machine learning, holds the perfboard MCU and IMU design, has a small slit to power the MCU, and is attached to the handle via 4 pronged teeth that easily slide onto the bat and clip to it for security. Through the process of collecting data, the prototype has proven to be robust enough to withstand the average hits and shocks that are to be expected from a cricket bat while in use. The prototype is also fairly unobtrusive to the user when swinging, which is a high priority for the final design.

For the final housing unit design, the same material and mounting mechanism will be used, however the overall size and design of the control and power system housing will be changed. Changing from perfboard to PCBs will allow for a smaller compartment and a sleeker design overall. It will be approximately half the prototype's height, will contain a single slot that allows the device to be charged when not in use, and a power button to turn the device on and off.

3.6. *Subsystem Conclusion*

The Bluno Beetle BLE microcontroller works as designed and is able to utilize a classic bluetooth module to link and pair with a mobile device. Each of the sensors on the MPU-6050 operate as designed. Despite some small discrepancies in the behavior of the IMU, the sensors function well within the scope of what the Smart Cricket Bat will need them to. However, the decision to switch to a more robust IMU is still an option, and it depends upon the needs of the machine learning. The microcontroller and sensors are a critical part of the overall system, and their ability to integrate with the consumer application ensures that they will be able to continuously collect data for processing.

4. App Subsystem Report

4.1. *Subsystem Introduction*

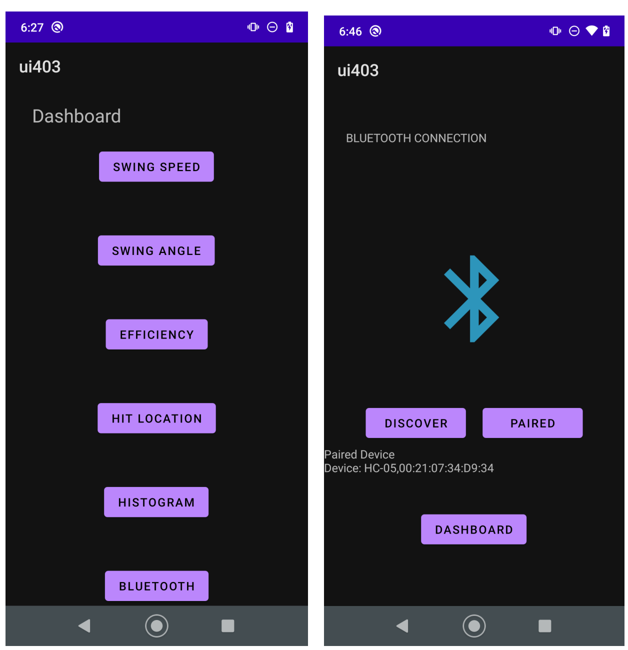
The android app has two main functions: communication between the control subsystem and the ML algorithm and handling user interaction. The app will receive the swing data gathered by the MCU via a bluetooth connection which will then be sent to the ML algorithm, that will run on the app, to be processed. Once processed, the user can navigate through different pages of the app by clicking various buttons on the dashboard to view their results. Each page will consist of a certain data output i.e. hit location, efficiency, or swing speed. The app is user friendly and does not require any advanced technical skill to allow any Cricket player to use our product.

4.2. *Subsystem Details*

4.2.1. Bluetooth Communication

As previously mentioned, the android app will handle the communication between the ML algorithm that will run on the app and the control subsystem via a bluetooth connection. To establish a bluetooth connection between the MCU, we will be using an HC-05 Bluetooth module. After each swing the user will press the designated bluetooth button on the dashboard that will then send a signal to trigger the MCU to send the swing data. An image of the dashboard and bluetooth page is shown below. The user is now free to navigate through the app to view their results. The Bluetooth page will also allow the user to make sure they are connected to the correct device by displaying its name and MAC address.

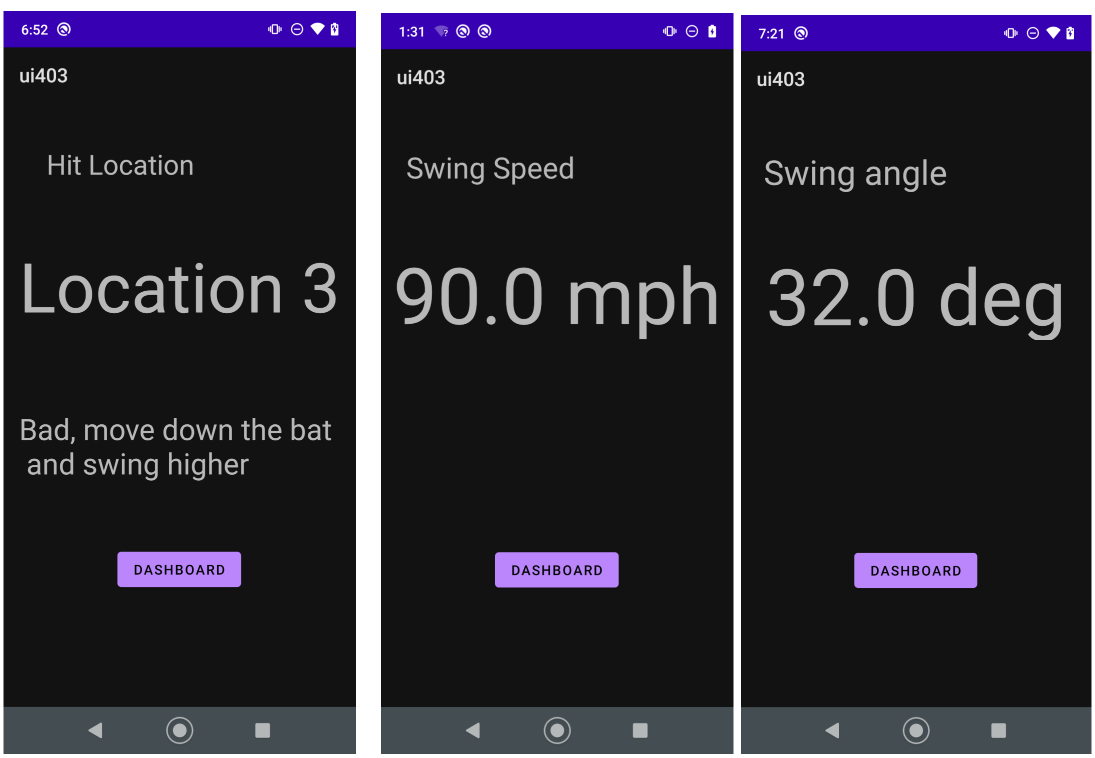
Figure 14: Bluetooth UI



4.2.2. User Interaction / Dashboard

The user interaction is controlled by the dashboard seen above. Each button is named corresponding to the result type it will output. The dashboard was designed to gather all the results calculated by the ML algorithm and send them to the individual pages once the user presses a button. The result types are as follows: Swing speed, Swing angle, Efficiency, Hit Location, and a heat map showing where the user is predominantly hitting the bat. The most important page is the hit location. In this page the user will be given the region where the collision with the ball occurred. The regions are numbers 1 through 15, where 1 is near the tip of the bat and 15 is closer to the handle of the bat. These regions will be marked on the bat. Depending on which region the ball hit, the user will be given advice as to how to get closer to the “sweet spot" of the bat. An example of the swing results being displayed in their corresponding page is shown below. In this example the user struck the bat near the tip of the bat on the upper edge, therefore the advice in the hit location page states “move down the bat and swing higher”. The location was above the sweet spot and on the upper edge, so the player needs to swing higher and get the collision lower on the bat to get closer to the center of the bat. The user can then navigate back to the dashboard by clicking on the dashboard button displayed on each page.

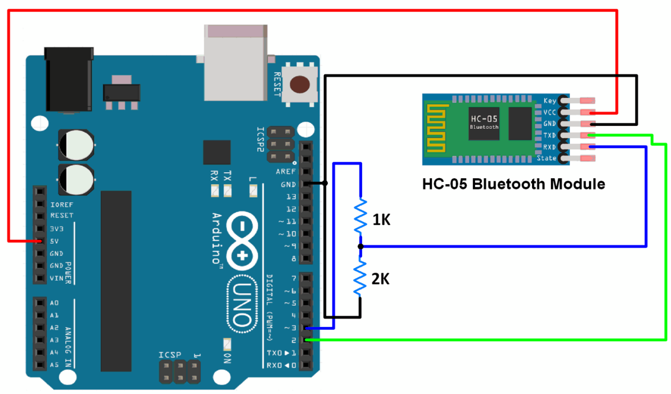
Figure 15: User Feedback



4.3. *Subsystem Validation*

Since the MCU, the Bluno Beetle, uses the Arduino IDE and schematic, I separated the HC-05 bluetooth module and connected it to an Arduino Uno to be able to validate without needing the control subsystem. The circuit that will be used to connect the HC-05 module with the Arduino and Bluno Beetle is shown below. Since the TTL levels of the arduino is 5V while the HC-05 module TTL level is 3.3V, a voltage divider with a resistance ration of 2:1 is used to attenuate the signal level.

Figure 16: HC-05 Bluetooth Module Circuit



To validate the bluetooth connectivity and data sending between the MCU and the android app, the MCU was programmed to send the entire alphabet once the app sent an input of 48 signifying the ASCII code for the character ‘0’. An image of the console output showing data being received by the android app is shown in Fig. 17. Although data was able to be sent via bluetooth, random data reading errors occur when first connecting the bluetooth module. The console also shows which device the app is connected to and whether we are connected to the bluetooth socket created. An image of the simple arduino code is also shown in Fig. 17.

```
I/System.out: HC-05
I/System.out: android.bluetooth.BluetoothSocket@c7ac960
W/BluetoothAdapter: getBluetoothService() called with no BluetoothManagerCallback
I/System.out: true
I/System.out: oStream
I/System.out: oStream
I/System.out: android.bluetooth.BluetoothOutputStream@e1c6ede
I/System.out: iStream
I/System.out: iStream
I/System.out: A
I/System.out: B
I/System.out: C
I/System.out: D
I/System.out: E
I/System.out: F
I/System.out: G
I/System.out: H
I/System.out: I
I/System.out: J
I/System.out: K
I/System.out: L
I/System.out: M
I/System.out: N
I/System.out: O
I/System.out: P
I/System.out: Q
I/System.out: R
I/System.out: S
```

```
byte receivedData;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

  if(Serial.available() > 0) {
    //reads the signal recieved from app
    receivedData = Serial.read();

    // if the signal recieves is 48 then
    // the entire alphabet will be sent to the app
    if(receivedData == 48) {
      for(int i = 65; i <= 90; i++) {
        Serial.print((char)i);
        delay(50);
      }
    }
  }

  delay(200);
}
```

Figure 17: Bluetooth validation data and arduino code

4.4. *Subsystem Conclusion*

The android app is easy to use without the need of an advanced technical skill. The main skill the user will need to know is how to pair their phone to a bluetooth device in their device's settings menu. Once paired, everything will be available to the user with a simple click of a button. The android app is also able to communicate and receive data from the MCU via a bluetooth connection, although there are some random reading errors that occur when first booting up the device. A link to the github repository containing the app source code is given below.

https://github.com/pablobarron7/Capstone_App.git

5. Machine Learning Subsystem Report

5.1. *Machine Learning Introduction*

The Machine Learning subsystem for this project is returning the region that the ball and bat impact to the user with the input of swing data. The method we are using is to divide the bat into different regions and do test swings on each region. Then ML will take the data from IMU and process them to give feedback of which region the impact happened. Then the users can look at the regions of their hit so that they can improve with their training and try to hit their “Sweet Spot” that they are looking for.

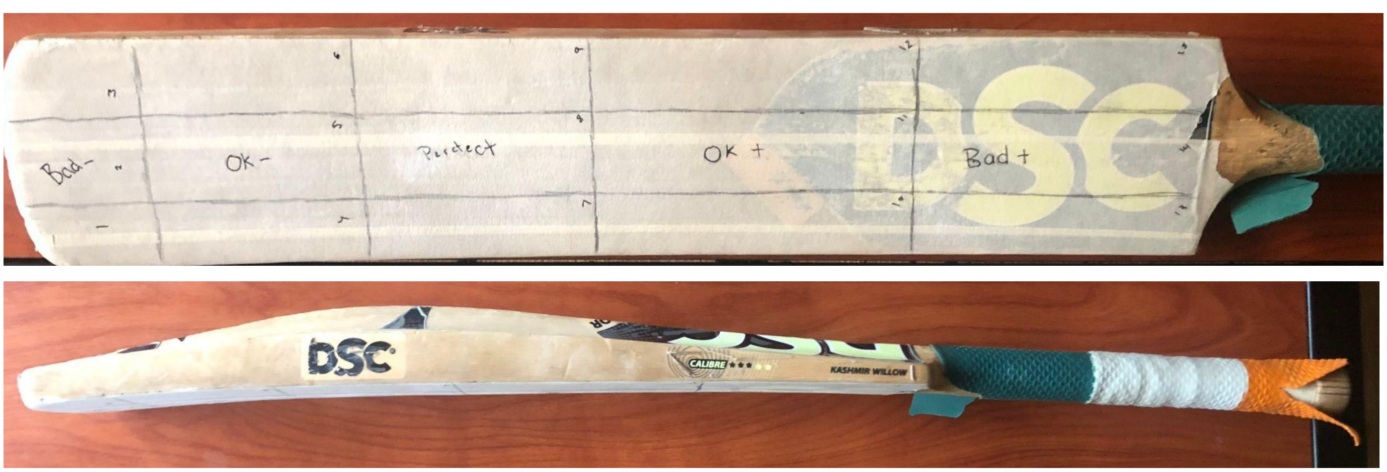
5.2. *Machine Learning Details*

The Machine Learning subsystem is done on python jupyter with tensor environments.

5.2.1. Bat Region Dividing

Based on the research, different regions of the bat provide different jerks when impact, which results in different eigenfrequency for us to determine. Based on that, we divided the bat into 15 regions like the image showing below. We marked the thiccest portion of the bad into “perfect” areas which is the “Sweet Spot” that batters usually refers to. For the rest of the regions, we named them after Bad-, OK-, OK+, and Bad+. Bad- is the tip of the bat which is where batter usually avoids while Bad+ is also a region that should be avoided by batter since it is close to the handle. The OK regions are closer to the sweet spot. They are better spots to hit, but they are not optimal. We also mark + and - to divide the region where + means impact region is closer to the handle and - means impact region is closer to the tip. Then we also marked all the regions as region 1-15 for processing in ML.

Figure 18: Bat Division



5.2.2. Data Gathering, Inputting, and Plotting

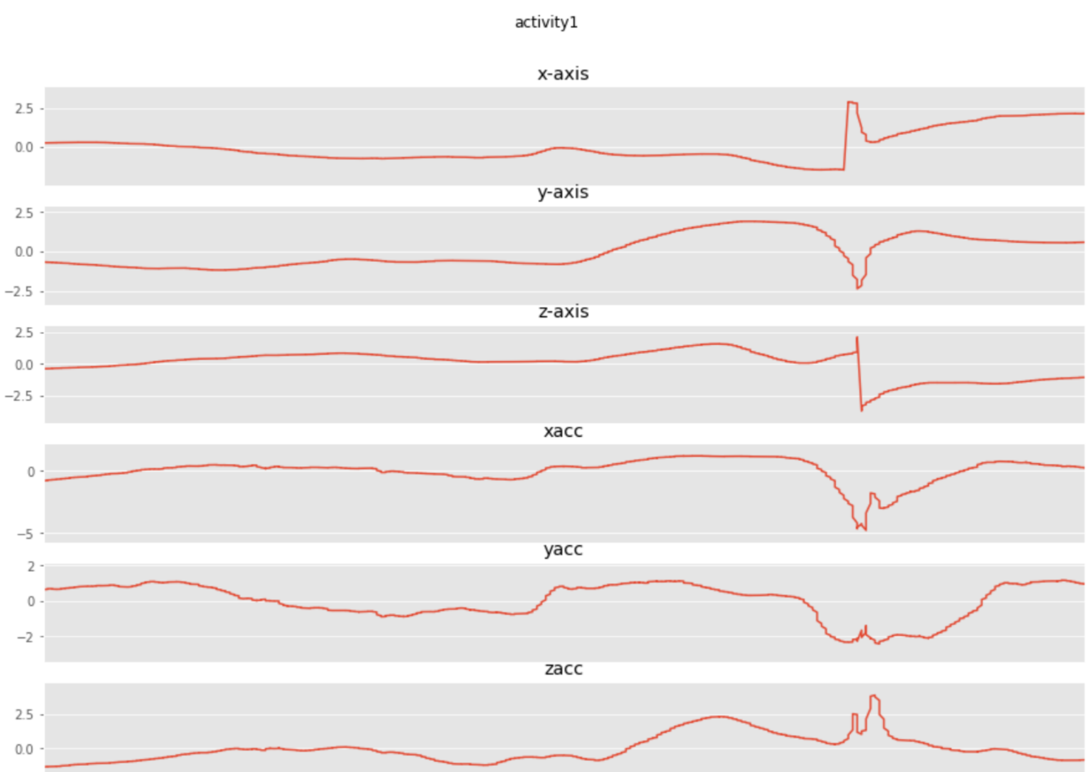
We did from 3 to 6 swings on each region and found out a total of 74 sets of data are usable. The ML takes the data gathered as input, then it does normalization on the data and plotting data over the entire duration out. The example figure shown below and all the examples later are from a set of data that corresponds to an impact on region “OK+12”.

Figure 19: Data Plotting

```
        timestamp    x-axis    y-axis    z-axis  space      xacc      yacc  \
0    19:08:08.569  0.233712 -0.661873 -0.416203     12 -0.824794  0.621280
1    19:08:08.569  0.235665 -0.666455 -0.413418     12 -0.808943  0.631412
2    19:08:08.569  0.238350 -0.670768 -0.410204     12 -0.793366  0.646189
3    19:08:08.601  0.241401 -0.675350 -0.406777     12 -0.776695  0.663077
4    19:08:08.601  0.244330 -0.679663 -0.402706     12 -0.762211  0.679542
..            ...       ...       ...       ...    ...       ...       ...
655  19:08:18.385  2.121169  0.566489 -1.121230     12  0.299233  1.018141
656  19:08:18.385  2.118118  0.569454 -1.115874     12  0.280649  1.000831
657  19:08:18.385  2.116409  0.572419 -1.109875     12  0.273544  0.987743
658  19:08:18.425  2.116043  0.574845 -1.103663     12  0.268898  0.974655
659  19:08:18.425  2.116409  0.577540 -1.097450     12  0.252227  0.959034

         zacc
0   -1.356982
1   -1.353737
2   -1.348638
3   -1.342612
4   -1.333805
..        ...
655 -0.866564
656 -0.862392
657 -0.851267
658 -0.840606
659 -0.839679

[660 rows x 8 columns]
```

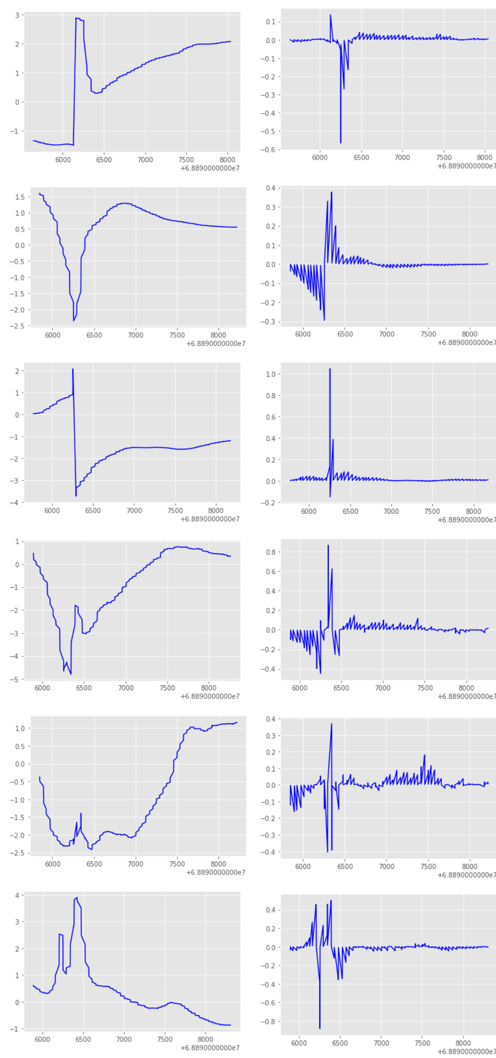


In the image, “x-axis”, “y-axis”, and “z-axis” are corresponding to the gyroscope and “xacc”, “yacc”, and “zacc” are corresponding to the accelerometer.

5.2.3. Finding the Impact

The ML took a derivative over the data and found the max point on the derivative to find out where the impact actually happens. From the max point on the derivative, we took from 32 values before the max to 128 values after the max to define the time window as impact times. Then it finds the same place on the original data to find the impact window. In the figure below, the left graphs are the original, and the right is its derivative.

Figure 20: Impact Window

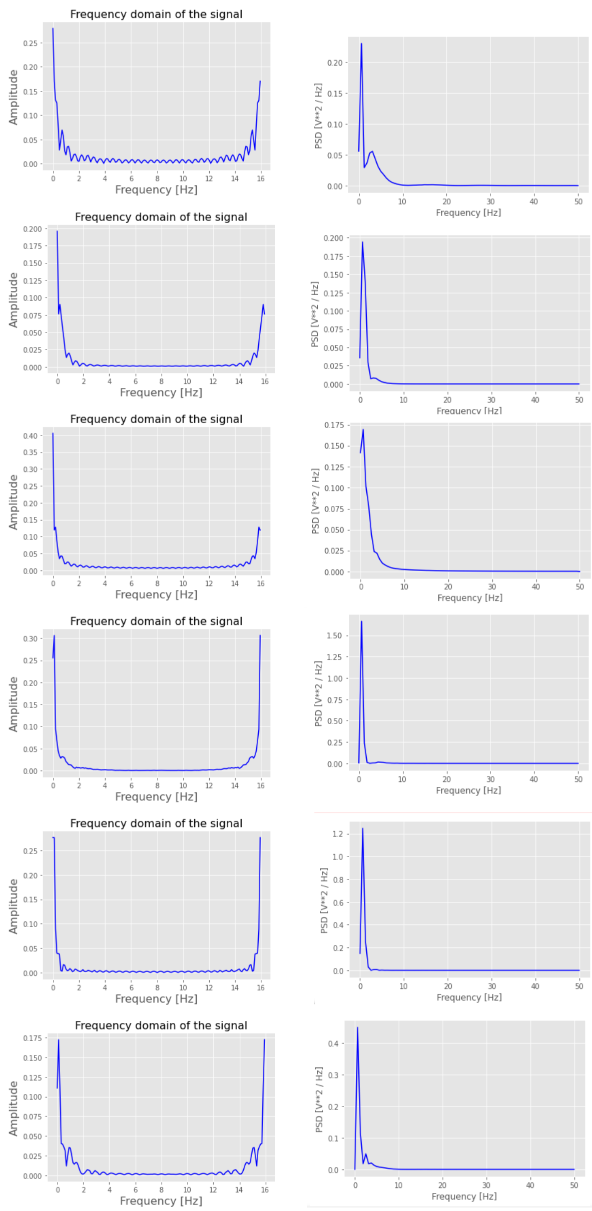


Timestamp is transformed here from hour, minute, sec, microsec format to total microseconds.

5.2.4. FFT and PSD

Over the window that just got sectioned out, FFT and PSD are done over these data in order to observe differences between datas to get started on identifying the datas. In figure 20, left group of graphs is the FFT and the right is the PSD.

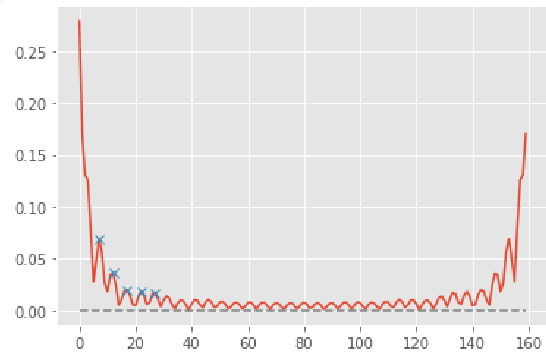
Figure 21: FFT & PSD



5.2.5. Peak Detection

The ML will try to find the first five peaks over each of the FFT and PSD that has been done and get x and y values for each of the peaks.

Figure 22: Peak Detection on FFT over X-axis Gyroscope of Region 12




```
[7, 12, 17, 22, 27]
[0.06947016418905683, 0.035941013579253965, 0.019924947096678643, 0.018412131514401943, 0.017477918489493868]
```

5.2.6. Making Feature and Label

The ML combines all the first five peaks' X and Y values of both FFT and PSD for all axes on both gyroscope and accelerometer together and labels them as the region of impact just like the image shown below.

Figure 23: Feature and Label

```
[7, 0.06947016418905683, 2, 0.09006751625397431, 2, 0.12767227491963032, 1, 0.3063243749729118, 8, 0.016275426277110244, 1, 0.1
721188313653635, 1, 0.2300152094951413, 1, 0.19432275818113218, 1, 0.16947977062715985, 1, 1.6632326246826332, 1, 1.2466560193
656304, 1, 0.4505763662772596, 12, 0.035941013579253965, 9, 0.01991166210807162, 6, 0.04236977487385565, 7, 0.0315174919720975,
13, 0.00821471824875793, 9, 0.03536262910857797, 5, 0.05560942229807287, 5, 0.008136180423151564, 77, 0.0005790551839976609, 7,
0.015281962565837153, 6, 0.007060219018616825, 4, 0.04941120526006808, 17, 0.019924947096678643, 14, 0.008656527765386113, 11,
0.02459476317046636, 18, 0.007412620345166541, 17, 0.0072096321522727765, 15, 0.016322491363520847, 24, 0.0015710708515240092,
53, 6.203373347488838e-07, 77, 0.0005790551839976609, 14, 0.0022544881259124886, 8, 0.0018676255255638857, 6, 0.020109585129433
418, 22, 0.018412131514401943, 20, 0.004923397990430082, 16, 0.018601935086806867, 20, 0.006778413974401534, 23, 0.006164822000
350596, 23, 0.007237363465328734, 27, 0.00158817276108223, 61, 3.8067118922260847e-07, 77, 0.0005790551839976609, 18, 0.0008774
767827991467, 10, 0.0008803244891070975, 21, 5.04884531937351e-05, 27, 0.017477918489493868, 25, 0.003399672552571814, 21, 0.01
5569839964794434, 23, 0.006607488311309628, 27, 0.004701290491062507, 29, 0.006181374417923675, 45, 0.00054201040338535, 63, 3.
5567824660933434e-07, 77, 0.0005790551839976609, 20, 0.0006962010743186173, 13, 0.00025298666216184344, 23, 0.00012066602998357
686]
[12]
```

The same process is done for all 74 sets of data that was acquired and combines them as features that contain all X and Y values and Labels that contain the impact regions.

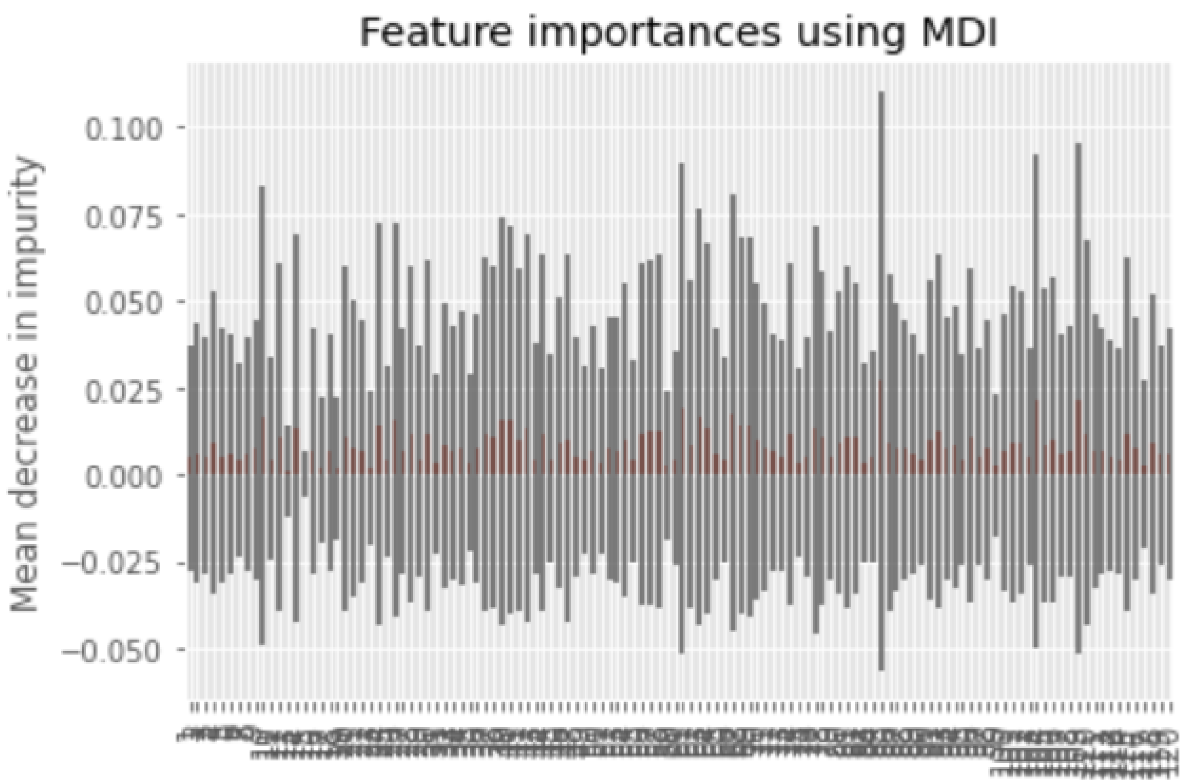
5.2.7. Training and Testing Sets

The data are divided into 60% for training and 40% for testing through trial and error.

5.2.8. Model and Feature Importance

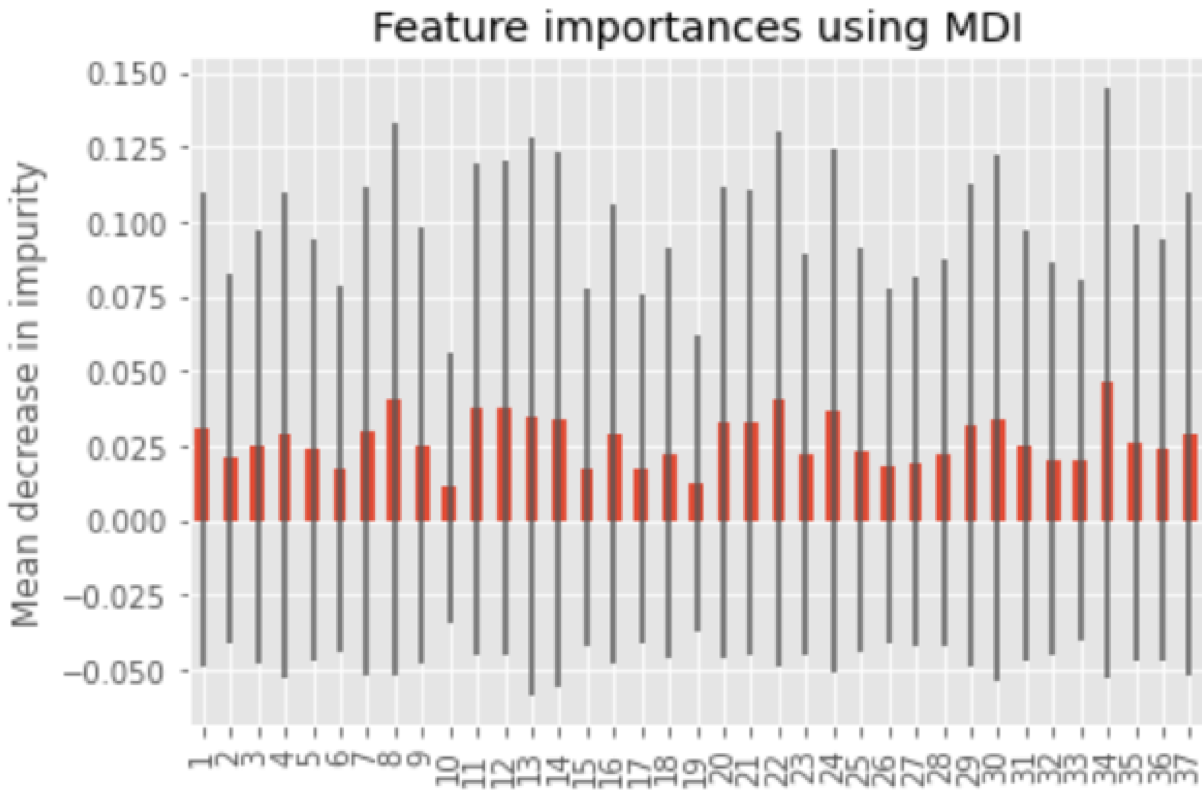
The data will be trained over Random Forest Classifier. Then the ML will use feature importance to find out all the important features to increase accuracy.

Figure 24: Feature Importance 1



The first time, a very low limit was set to take out the least important features. The training was done again and this time it is more clear to see the more important features.

Figure 25: Feature Importance 2



This process is done again so we have the most important features going over the Random Forest Classifier.

5.3. *Machine Learning Validation*

We have done more than 6 models and over 20 combinations like different portions of Train Test Split, or random shuffle with Random Forest, Logistic Regression, svm with kernel, rbf, linear or ploy and many more. From the results and the pattern of the accuracy increasing and decreasing, we found out we need more data to get further accuracy. As a result, this result with Random Forest Classifier with 60% training and 40% testing with random state of 42 is the best situation we can get with our amount of data.

Figure 26: Training and Testing Accuracy

```
Accuracy on training set is : 0.9607843137254902
Accuracy on test set is : 0.13043478260869565
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         1
           2       0.50      1.00      0.67         1
           3       0.00      0.00      0.00         3
           4       0.17      1.00      0.29         1
           5       0.00      0.00      0.00         2
           6       0.00      0.00      0.00         2
           7       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         2
          10       0.00      0.00      0.00         1
          11       0.00      0.00      0.00         2
          12       0.33      0.50      0.40         2
          13       0.00      0.00      0.00         2
          14       0.00      0.00      0.00         2
          15       0.00      0.00      0.00         2

    accuracy                           0.13        23
   macro avg       0.07      0.18      0.10        23
weighted avg       0.06      0.13      0.08        23

[12  4  7  1  2 12  2  7  7  8 15  7  4  5  4  4  4 15 12  7  8  4  8]
[12  4 15 12  2 11 14  3 14  5 10  3 13  8 13  1  5  8  3  6 11 15  6]
```

5.4. ***Subsystem Conclusion***

The Machine Learning subsystem is working as expected and reached the goal of this semester. The ML is able to receive the data and do the process needed like normalizing, windowing, fft, psd, peak detection, feature extraction, training, and testing. In order to have better accuracy, we need to do much more data gathering in the next semester. As a result, the ML subsystem has the ability to transform the necessary data to give feedback for users to see their impact region on the bat.