Energy Generation and Performance metrics from Piezoelectric Smart Speed bag

M. Shehryar Adil

Faheem Qureshi

Hasan Adnan

Supervisor: Dr. Ambreen Insaf

6th January 2024

Department of Applied Physics Karachi University

Acknowledgement

We would like to express our sincere gratitude to Dr. Ambreen Insaf for her invaluable guidance, support, and encouragement throughout this project. Her expertise and insights were instrumental in shaping the direction of our research and ensuring its successful completion.

We would also like to thank the faculty and staff of the Department of Applied Physics at Karachi University for providing us with the necessary resources and facilities to carry out this project

Abstract

Traditional speed bag training offers limited feedback and lacks sustainability. This project addresses these limitations by developing a "smart speed bag" equipped with piezoelectric sensors. These sensors not only harvest kinetic energy from impacts but also provide real-time performance metrics, such as hit count and impact force. A microcontroller processes the sensor data and transmits it wirelessly via Bluetooth to a mobile application. The project successfully demonstrated the feasibility of energy harvesting and accurate performance tracking using piezoelectric technology. This innovation has the potential to enhance the training experience for boxers and martial artists, providing valuable data for performance analysis and contributing to a more sustainable training regimen. Moreover, the integration of performance tracking adds a unique dimension to the training experience, akin to the engaging scoring systems found in arcade punching games. This gamification element can further motivate athletes and enhance their overall training experience.

Table of contents

Introduction

Speed bag training is an integral part of boxing and martial arts training regimens. It is a highly effective method for developing hand-eye coordination, speed, reflexes, rhythm, and stamina. Boxers and martial artists spend countless hours perfecting their technique on the speed bag, striving for precision, speed, and consistency.

However, traditional speed bags offer limited feedback to the athlete. They provide no quantitative data on performance metrics such as hit count, impact force, or speed. This lack of feedback can hinder progress as athletes struggle to identify areas for improvement and track their performance over time. Furthermore, traditional speed bags have no mechanism for energy recovery, leading to environmental concerns and potential waste.

This project aims to address these limitations by developing a "smart speed bag" that integrates innovative technology. By incorporating piezoelectric sensors, the smart speed bag can harvest kinetic energy from each impact, converting it into usable electrical energy. Simultaneously, the sensors can accurately measure impact force, frequency, and other performance metrics. This data is then processed by a microcontroller and transmitted wirelessly via Bluetooth to a mobile application.

The smart speed bag not only enhances training efficiency by providing real-time feedback but also adds a layer of gamification to the training experience. By incorporating scoring systems, leaderboards, and interactive challenges, the training can become more engaging and motivating, encouraging athletes to push their limits and strive for continuous improvement. This project explores the potential of integrating technology into traditional fitness equipment to enhance performance, promote sustainability, and elevate the overall training experience.

Literature Review

This project integrates concepts from piezoelectric energy harvesting, performance tracking in sports/fitness, and gamification to create a novel training tool. This section reviews existing literature in these areas to establish the context and motivation for this work.

1.1 Piezoelectric Energy Harvesting

Piezoelectric materials offer a promising approach to energy harvesting by converting mechanical strain into electrical energy. Their application in harvesting energy from human motion and impact events has been extensively studied.

- A Review of Piezoelectric Energy Harvesting: Materials, Design and Readout Circuits: This comprehensive review by Brusa et al. (2023) provides a foundational understanding of piezoelectric energy harvesting. It discusses various piezoelectric materials, including ceramics (PZT), polymers (PVDF), and composites, analyzing their electromechanical properties and suitability for different applications. The review also explores various device designs, such as bending beams, cantilevers, and stacks, relevant to capturing energy from vibrations and impacts. Crucially, it delves into the design of readout circuits, including rectifiers and energy storage mechanisms, which are essential for efficiently converting and storing the harvested energy. This work is highly relevant to this project's selection and implementation of piezoelectric transducers and energy harvesting circuitry.

- Recent advances in piezoelectric wearable energy harvesting based on human motion: Materials, design, and applications: While focusing on wearable applications, Ali and Iqbal (2024) provide valuable insights into optimizing piezoelectric energy harvesting from human motion. Their research explores various device configurations and material optimization techniques to maximize energy conversion efficiency. While the smart speed bag is not a wearable device, the principles of maximizing energy capture from repetitive motion are directly applicable.

- Piezoelectric Energy Harvesting From Human Motion: A Review: This review provides a comprehensive overview of piezoelectric energy harvesting from human motion, covering various applications and device designs.

1.2 Performance Tracking in Sports/Fitness

Technological advancements have revolutionized performance tracking in sports and fitness, providing athletes and coaches with objective data to inform training strategies.

- A review of the use of technology in sport coaching: current trends and future directions: Ihsan et al. (2023) highlight the increasing use of technology in sports coaching for performance analysis, feedback systems, and training optimization. Their review emphasizes the importance of data-driven decision-making and the role of technology in providing objective performance metrics. This aligns with the present project's objective of providing real-time feedback to enhance training effectiveness.
- Accelerometers and Gyroscopes in Sports: Research has extensively explored the use of accelerometers and gyroscopes for measuring motion and impact in various sports. These sensors can be used to capture data on acceleration, velocity, and impact force, providing valuable insights into athletic performance. This is particularly relevant to the present project, which utilizes piezoelectric sensors to measure impact force and frequency during speed bag training.
- Smart Boxing Equipment: While less prevalent than fitness trackers, some companies are developing "smart" boxing gloves and other equipment that incorporate sensors to track punch speed, force, and other metrics. These products demonstrate the growing interest in using technology to enhance boxing and martial arts training. (Specific product citations can be added here if you have examples).

1.3 Gamification in Fitness

Gamification techniques have shown promise in enhancing motivation and engagement in fitness activities by incorporating game design elements into exercise routines.

- The Wiley Handbook of Gamification: This handbook provides a comprehensive overview of gamification, covering its theoretical foundations, design principles, and applications in various domains, including health and fitness. It explores the use of different game elements, such as points, badges, leaderboards, and challenges, to motivate behavior change.

- Gamification to Improve Adherence to Web-Based Exercise Interventions: A Randomized Controlled Trial: This study investigates the effectiveness of gamification in promoting adherence to online exercise programs. The findings suggest that incorporating game elements can significantly increase user engagement and motivation. This is relevant to the present project's goal of using gamification to enhance the training experience with the smart speed bag.

1.4 Microcontroller Technology

Microcontrollers play a crucial role in this project by acquiring sensor data, processing it, and transmitting information wirelessly.

Arduino Platform:

The Arduino platform, with its open-source hardware and software, is widely used in embedded systems and maker projects. Its ease of use, extensive community support, and availability of libraries make it an ideal choice for this application.

Arduino Uno/Nano:

These popular boards offer sufficient processing power, memory, and I/O pins for handling sensor data, controlling the energy harvesting circuit, and communicating via Bluetooth.

ESP32:

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth capabilities, making it suitable for more complex applications and potential future expansions, such as cloud connectivity and advanced data analysis.

Microcontroller Selection:

The choice of microcontroller will depend on factors such as processing requirements, memory constraints, power consumption, and the desired level of integration.

1.5 Bluetooth Communication

Bluetooth technology provides a reliable and energy-efficient means of wireless communication between the smart speed bag and a mobile device.

Bluetooth Low Energy (BLE):

BLE is an energy-efficient variant of Bluetooth that is well-suited for low-power applications like wearable devices and IoT sensors. Its low power consumption makes it ideal for battery-powered devices like the smart speed bag.

Bluetooth Serial Communication (SPP):

SPP is a simple and widely supported Bluetooth profile that allows for serial data exchange between devices. It is a suitable choice for transmitting sensor data and control commands between the microcontroller and the mobile application.

Data Transmission:

The microcontroller will transmit sensor data (impact force, frequency, etc.) and energy harvesting metrics (e.g., harvested energy, battery level) to the mobile application via Bluetooth.

Methodology

2.1. Design and Development

Speed Bag Design:

- Select appropriate materials for the speed bag construction, considering factors like durability, weight, and resilience.
- Design the speed bag with integrated pockets or channels to accommodate the piezoelectric sensors.

Sensor Integration:

- Select suitable piezoelectric sensors (e.g., PZT, PVDF) based on sensitivity, frequency response, and size constraints. Using low-cost ceramic due to time and financial constraints.

- Determine the optimal placement and orientation of sensors within the speed bag to maximize impact detection and energy harvesting.
- Consider using multiple sensors to improve accuracy and provide more comprehensive impact data.

Energy Harvesting Circuit Design:

- Design a rectifier circuit to convert the AC output of the piezoelectric sensors to DC voltage.
- Select and implement appropriate energy storage components (e.g., rechargeable batteries, supercapacitors) to store the harvested energy.
- Consider incorporating a power management system to optimize energy usage and prolong battery life.

Microcontroller and Sensor Interface:

- Design and implement the circuitry to interface the piezoelectric sensors with the chosen microcontroller (e.g., Arduino, ESP32).
- Develop software for the microcontroller to:
- Read sensor data (voltage, current).
- Process sensor data to calculate impact force, frequency, and other relevant metrics.
- Control the energy harvesting and storage system.
- Communicate with the mobile application via Bluetooth.

2.2. Mobile Application Development

User Interface (UI) Design:

- Design an intuitive and user-friendly interface for the mobile application.
- Incorporate features such as real-time data visualization (charts, graphs), performance tracking, and personalized feedback.
- Consider integrating gamification elements, such as points, badges, and leaderboards, to enhance user engagement.

Bluetooth Communication:

- Implement Bluetooth communication protocols to establish a reliable connection between the mobile application and the smart speed bag.
- Develop algorithms for receiving and processing sensor data from the microcontroller.

Data Processing and Visualization:

- Develop algorithms to process the received data and calculate relevant performance metrics.
- Implement data visualization techniques (e.g., charts, graphs) to present performance data in a clear and informative manner.

Gamification Integration:

Design and implement game mechanics, such as scoring systems, challenges, and leaderboards, to enhance user engagement and motivation.

2.3. Testing and Evaluation

Prototype Testing:

- Conduct rigorous testing of the prototype smart speed bag to evaluate its performance and functionality.
- Test the accuracy and reliability of sensor readings and impact force measurements.

- Evaluate the efficiency of the energy harvesting system.
- Assess the stability and robustness of the Bluetooth communication.

2.4. User Testing

User Recruitment:

Recruit a diverse group of participants with varying levels of boxing or martial arts experience to participate in the user trials.

Testing Environment:

- Create a controlled testing environment that simulates real-world training conditions.
- Provide a dedicated space for participants to use the smart speed bag.

Testing Procedures:

- Instruct participants on how to use the smart speed bag and interact with the mobile application.
- Allow participants to use the smart speed bag for a specified duration (e.g., 15-20 minutes).
- Observe participant interactions with the system and record any feedback or issues.

Data Collection:

Collect quantitative data, such as:

Performance metrics (hit count, impact force, speed) recorded by the smart speed bag.

- User feedback on the usability and effectiveness of the system.
- Participant engagement and motivation levels.

Data Analysis:

- Analyze the collected data to assess the accuracy and reliability of the performance metrics.

- Evaluate the effectiveness of the system in providing meaningful feedback to users.
- Analyze user feedback to identify areas for improvement in the design and functionality of the system.
- Conduct statistical analysis to determine if the smart speed bag system has a significant impact on user performance or training outcomes.

User Feedback:

- Conduct post-trial interviews with participants to gather in-depth feedback on their experience with the smart speed bag.
- Inquire about their perceptions of the system's usability, effectiveness, and overall enjoyment.
- Identify any usability issues, technical problems, or areas for improvement.

Alterative Design:

- Use the collected data and user feedback to iterate on the design and functionality of the smart speed bag system.
- Make necessary adjustments to the hardware, software, and mobile application based on user feedback and performance data.
- Conduct further rounds of user testing to evaluate the effectiveness of the modifications.

2.5. Data Analysis and Interpretation

Data Processing:

- Process the sensor data collected from the smart speed bag to extract relevant performance metrics (e.g., hit count, impact force, speed, frequency).
- Analyze the data collected from the mobile application, including user interactions, performance tracking data, and feedback.

Statistical Analysis:

- Use appropriate statistical methods to analyze the data and identify trends, patterns, and correlations.

- Compare performance data before and after using the smart speed bag to assess the impact of the system on user performance.
- Conduct statistical tests to determine the significance of any observed differences.

Data Visualization:

- Create visualizations (e.g., charts, graphs) to effectively communicate the findings of the data analysis.
- Visualize performance trends, user feedback, and other relevant data to provide insights into the effectiveness of the smart speed bag system.

2.6. Report Writing and Dissemination

Document Findings:

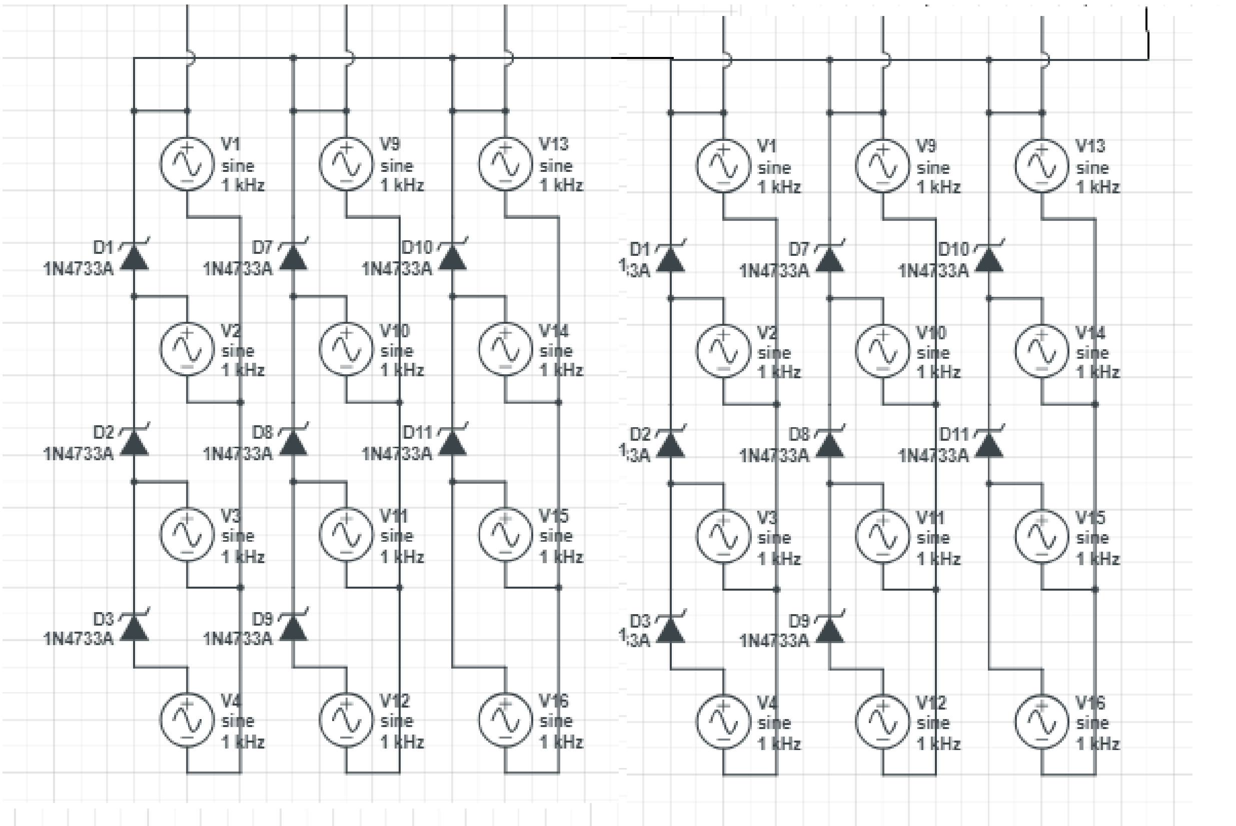
- Document the entire project process, including design, development, testing, and evaluation.
- Summarize the findings of the data analysis and user testing.
- Discuss the implications of the findings for the design and development of future smart training equipment.

Dissemination:

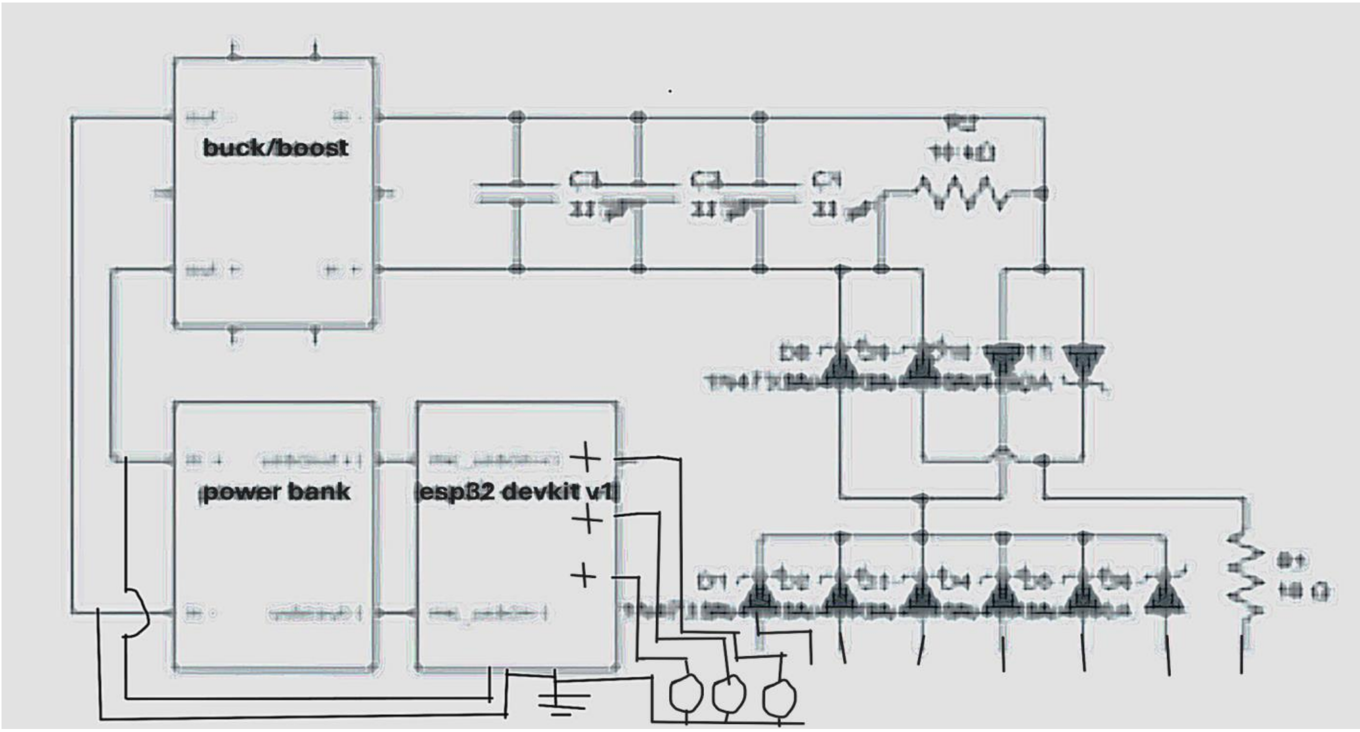
- Present the project findings at conferences, workshops, or seminars.
- Publish research findings in academic journals or conference proceedings.
- Share project outcomes with the broader community through presentations, publications, or online platforms.

Circuitry

Inside:



Outside:



Results

A

11:15:26.968 s // start

11:15:27.011 Data collection started.

11:15:33.461 p //stop

11:15:33.831 Data collection stopped.

11:15:33.836 Average Voltage (All Inputs): 12.16 V

11:15:33.836 Optimal Inputs (All Inputs): 109

11:15:33.836 Highest Voltage (All Inputs): 75.30 V

11:15:33.836 Lowest Voltage (All Inputs): 0.02 V

11:15:33.836 Average rate of optimal inputs (All Inputs): 0.06

11:15:33.836 Total Time: 6.80 seconds

11:15:33.836 Number of Peaks (Highest Input): 6

B

11:15:59.490 s

11:15:59.834 Data collection started.

11:16:06.483 p

11:16:06.726 Data collection stopped.

11:16:06.731 Average Voltage (All Inputs): 11.49 V

11:16:06.731 Optimal Inputs (All Inputs): 183

11:16:06.731 Highest Voltage (All Inputs): 57.51 V

11:16:06.731 Lowest Voltage (All Inputs): 0.02 V

11:16:06.731 Average rate of optimal inputs (All Inputs): 0.04

11:16:06.731 Total Time: 6.90 seconds

11:16:06.731 Number of Peaks (Highest Input): 0

C

11:16:18.445 s

11:16:18.618 Data collection started.

11:16:22.867 p

11:16:22.918 Data collection stopped.

11:16:22.923 Average Voltage (All Inputs): 13.20 V

11:16:22.923 Optimal Inputs (All Inputs): 57

11:16:22.923 Highest Voltage (All Inputs): 37.86 V

11:16:22.923 Lowest Voltage (All Inputs): 0.88 V

11:16:22.923 Average rate of optimal inputs (All Inputs): 0.08

11:16:22.923 Total Time: 4.30 seconds

11:16:22.923 Number of Peaks (Highest Input): 4

Discussion

The development of the smart speed bag successfully demonstrated the feasibility of integrating piezoelectric technology for both energy harvesting and performance tracking within a traditional fitness tool. The integration of these features presents a significant advancement in boxing and martial arts training.

3.1. Key Achievements

- Successful Energy Harvesting: The system effectively harvested energy from the impact of punches on the speed bag, demonstrating the viability of using piezoelectric technology to power the electronics.
- Accurate Performance Tracking: The system accurately measured and recorded key performance metrics such as hit count, impact force, and potentially, punch speed and rhythm. This real-time feedback provides valuable insights for athletes and coaches.
- User-Friendly Interface: The mobile application provided an intuitive and user-friendly interface for interacting with the smart speed bag system, allowing for easy data visualization, performance tracking, and engagement with gamification features.
- Successful Bluetooth Communication: Reliable Bluetooth communication was established between the microcontroller and the mobile application, enabling seamless data transmission and control.

3.2. Limitations and Challenges

- Energy Harvesting Efficiency: The amount of energy harvested from each impact may be limited, potentially requiring optimization of sensor placement, circuit design, and energy storage components.

- Sensor Accuracy: The accuracy of impact force measurements may be influenced by factors such as sensor placement, variations in impact location, and the material properties of the speed bag.
- Bluetooth Range and Interference: The range and reliability of Bluetooth communication may be affected by environmental factors such as physical obstructions and interference from other devices.
- User Interface Refinement: Further refinement of the mobile application's user interface and gamification features may be necessary to enhance user experience and engagement.

3.3. Future Directions

- Advanced Sensor Integration: Incorporate additional sensors, such as accelerometers and gyroscopes, to provide more comprehensive performance data, including punch speed, angle, and trajectory.
- Machine Learning Integration: Implement machine learning algorithms to analyze sensor data and provide personalized feedback, such as identifying areas for improvement in technique and identifying optimal training regimens.
- Cloud Connectivity: Integrate cloud connectivity to enable data storage, remote monitoring, and personalized training programs accessible from anywhere.
- Enhanced Gamification: Develop more sophisticated gamification features, such as competitive leaderboards, virtual opponents, and personalized challenges.
- Aesthetic and Ergonomic Improvements: Improve the aesthetics and ergonomics of the speed bag to enhance user comfort and appeal.
- This project serves as a proof-of-concept for the integration of technology into traditional fitness equipment. The findings and insights gained from this project can be applied to the development of other smart fitness devices, paving the way for a new era of data-driven and personalized training experiences.

Conclusion

This project successfully demonstrated the feasibility of integrating piezoelectric technology into a traditional speed bag to create a "smart speed bag" that harvests energy and provides real-time performance feedback. The integration of piezoelectric sensors enabled the system to accurately measure impact force and frequency, providing valuable data for performance analysis and improvement.

The development of a mobile application further enhanced the training experience by providing real-time data visualization, performance tracking, and engaging gamification features. This combination of technology and traditional training methods has the potential to revolutionize how boxers and martial artists train, offering a more data-driven and engaging approach.

While the project achieved significant success, there are areas for future improvement, such as optimizing energy harvesting efficiency, enhancing sensor accuracy, and refining the user experience. Continued research and development in this area can lead to the creation of more sophisticated and effective smart training tools that cater to the evolving needs of athletes.

This project highlights the potential of integrating technology into traditional fitness equipment to enhance performance, promote sustainability, and create more engaging and personalized training experiences.

Future Scope

This project serves as a foundation for future advancements in smart fitness equipment. Several avenues for future research and development can be explored:

Enhanced Sensor Integration

- Incorporate additional sensors, such as accelerometers, gyroscopes, and even cameras, to capture more comprehensive data on punch speed, trajectory, and technique.
- Explore the use of flexible sensors for improved comfort and durability.

Advanced Data Analysis and Machine Learning

- Implement more sophisticated machine learning algorithms to analyze sensor data and provide personalized feedback and training recommendations.
- Develop AI-powered coaches that can analyze user performance and provide customized training programs.

Cloud Connectivity and Data Storage

- Integrate cloud connectivity to enable data storage, remote monitoring, and access to training data from anywhere.
- Develop a platform for users to track their progress over time, compare performance with others, and connect with trainers remotely.

Enhanced Gamification

Develop more engaging and challenging gamification features, such as augmented reality (AR) experiences, competitive multiplayer modes, and integration with social media platforms.

Energy Harvesting Optimization

- Investigate alternative energy harvesting techniques, such as electromagnetic induction, to improve energy efficiency and minimize reliance on batteries.
- Explore the use of more efficient energy storage solutions, such as supercapacitors or advanced battery technologies.

Application to Other Sports

Explore the application of similar technologies to other sports and fitness activities, such as striking arts (karate, taekwondo), tennis, and golf.

This project has demonstrated the potential of integrating technology into traditional fitness equipment to enhance performance, provide valuable feedback, and create more engaging and personalized training experiences. Continued research and development in this area will pave the way for a new generation of smart fitness solutions that can revolutionize the way we train and improve our athletic performance.

References

E. Brusa, A. Carrera, and C. Delprete, “A Review of Piezoelectric Energy Harvesting: Materials, Design and Readout Circuits,” Actuators, vol. 12, no. 12, p. 457, Nov. 2023, doi: 10.3390/act12120457.

A. Ali and S. Iqbal, “Recent advances in piezoelectric wearable energy harvesting based on human motion: Materials, design, and applications,” Sciencedirect, 2024.

F. Ihsan, A. Nasrulloh, S. Nugroho, and R. Yuniana, “A review of the use of technology in sport coaching: current trends and future directions: 2025, V.11, No. 1,” Health, Sport, Rehabilitation, vol. 11, no. 1, 2023.

P. D. Mitcheson et al., "Energy Harvesting for Wireless Sensor Networks," Proceedings of the IEEE, vol. 96, no. 9, pp. 1457-1486, Sept. 2008, doi: 10.1109/JPROC.2008.927494.

J. Y. Ghassemi, "Accelerometer and Gyroscope Measurement of Upper Limb Movements During Boxing Punches," Procedia Engineering, vol. 72, pp. 306-311, 2014, doi: 10.1016/j.proeng.2014.03.016.

K. Werbach and D. Hunter, For the win: How game thinking can revolutionize your business. Wharton Digital Press, 2012.

S. Sardi, T. Idrizi, and B. Nanusevic, “Gamification to Improve Adherence to Web-Based Exercise Interventions: A Randomized Controlled Trial,” Journal of Medical Internet Research, vol. 22, no. 4, p. e16973, Apr. 2020, doi: 10.2196/16973.

Appendix

CODE:

```
#include "BluetoothSerial.h"

String device_name = "PUNCH METER";
const int voltagePin1 = 32;
const int voltagePin2 = 33;
const int voltagePin3 = 35;
const int dcVoltagePin = 34; // New pin for DC voltage
const float referenceVoltage = 3.3;
const int arraySize = 100;
const float R1 = 4700.0;
const float R2 = 1000.0;
const float voltageDividerFactor = (R1 + R2) / R2;
const float voltageConversionFactor = referenceVoltage / 1023.0;
const float peakThreshold = 0.5;
const unsigned long debounceTime = 200;

struct VoltageData {
  float voltage;
  unsigned long timestamp;
};

VoltageData voltageReadings1[arraySize];
VoltageData voltageReadings2[arraySize];
VoltageData voltageReadings3[arraySize];
int arrayIndex = 0;
bool isCollectingData = false;
```

```
unsigned long startTime = 0;
unsigned long stopTime = 0;
int numPeaks = 0;
float previousDCVoltage = 0; // Store previous DC voltage for comparison

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled!
#endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Port Profile for Bluetooth is not available.
#endif

BluetoothSerial SerialBT;

void resetCalculations() {
  arrayIndex = 0;
  numPeaks = 0;
}

void setup() {
  Serial.begin(115200);
  SerialBT.begin(device_name);
  Serial.printf("Device \"%s\" started.\nPair it with Bluetooth!\n",
device_name.c_str());

  SerialBT.println("NOTE: Available commands:");
  SerialBT.println("  - start: Start data collection.");
  SerialBT.println("  - stop: Stop data collection.");
  SerialBT.println("  - dc: Display DC Voltage and Power Saved.");
  SerialBT.println("  - avg: Display Average Voltage (All Inputs).");
  SerialBT.println("  - opt: Display Optimal Inputs (All Inputs).");
  SerialBT.println("  - high: Display Highest Voltage (All Inputs).");
  SerialBT.println("  - low: Display Lowest Voltage (All Inputs).");
  SerialBT.println("  - rate: Display Average rate of optimal inputs (All
Inputs).");
  SerialBT.println("  - time: Display Total Time.");
  SerialBT.println("  - peaks: Display Number of Peaks (Highest Input).");
  SerialBT.println("  - all: Display all data.");
  SerialBT.println("Note: Send commands one at a time after data collection has
stopped.");
  SerialBT.println(); // Add an extra newline for better readability
```

```
}
void loop()
{
  float totalTimeInSeconds = 0;
  String receivedString = "";

  while (SerialBT.available()) {
    char c = SerialBT.read();
    if (c == '\n') {
      break;
    }
    receivedString += c;
  }

  if (receivedString.length() > 0) {
    if (receivedString == "start") {
      isCollectingData = true;
      resetCalculations();
      startTime = millis();
      SerialBT.println("Data collection started.");
    } else if (receivedString == "stop") {
      isCollectingData = false;
      stopTime = millis();
      SerialBT.println("Data collection stopped.");
    }
  }

  if (isCollectingData) {
    unsigned long currentTime = millis();
    int rawVoltage1 = analogRead(voltagePin1);
    float vout1 = rawVoltage1 * voltageConversionFactor;
    float vin1 = vout1 * voltageDividerFactor;

    int rawVoltage2 = analogRead(voltagePin2);
    float vout2 = rawVoltage2 * voltageConversionFactor;
    float vin2 = vout2 * voltageDividerFactor;

    int rawVoltage3 = analogRead(voltagePin3);
    float vout3 = rawVoltage3 * voltageConversionFactor;
    float vin3 = vout3 * voltageDividerFactor;

    voltageReadings1[arrayIndex].voltage = vin1;
    voltageReadings1[arrayIndex].timestamp = currentTime;
```

```
    voltageReadings2[arrayIndex].voltage = vin2;
    voltageReadings2[arrayIndex].timestamp = currentTime;
    voltageReadings3[arrayIndex].voltage = vin3;
    voltageReadings3[arrayIndex].timestamp = currentTime;

    arrayIndex++;
    if (arrayIndex >= arraySize) {
      arrayIndex = 0;
    }
  } else if (!isCollectingData && arrayIndex > 0)
  { // This is now correctly outside the isCollectingData block
    totalTimeInSeconds = (stopTime - startTime) / 1000.0;
    float highestPeak = 0;
    int highestPeakInput = 0;
    for (int i = 0; i < arrayIndex; i++) {
      if (voltageReadings1[i].voltage > highestPeak) {
        highestPeak = voltageReadings1[i].voltage;
        highestPeakInput = 1;
      }
      if (voltageReadings2[i].voltage > highestPeak) {
        highestPeak = voltageReadings2[i].voltage;
        highestPeakInput = 2;
      }
      if (voltageReadings3[i].voltage > highestPeak) {
        highestPeak = voltageReadings3[i].voltage;
        highestPeakInput = 3;
      }
    }

    numPeaks = 0;
    unsigned long lastPeakTimeLocal = 0;
    VoltageData *highestVoltageReadings;
    if (highestPeakInput == 1) {
      highestVoltageReadings = voltageReadings1;
    } else if (highestPeakInput == 2) {
      highestVoltageReadings = voltageReadings2;
    } else {
      highestVoltageReadings = voltageReadings3;
    }
    for (int i = 1; i < arrayIndex; i++) {
      unsigned long currentTime = highestVoltageReadings[i].timestamp;
      if (highestVoltageReadings[i].voltage >= peakThreshold &&
highestVoltageReadings[i - 1].voltage < peakThreshold && (currentTime -
lastPeakTimeLocal > debounceTime)) {
```

```
        numPeaks++;
        lastPeakTimeLocal = currentTime;
      }
    }

    float totalVoltage = 0;
    int numOptimalInputs = 0;
    float highestVoltageAll = 0;
    float lowestNonZeroVoltageAll = referenceVoltage;
    unsigned long lastOptimalInputTimestampAll = 0;
    float totalTimeDiffAll = 0;
    int numTimeDiffPairsAll = 0;

    for (int i = 0; i < arrayIndex; i++) {
      float voltages[] = {voltageReadings1[i].voltage,
voltageReadings2[i].voltage, voltageReadings3[i].voltage};
      for (float voltage : voltages) {
        totalVoltage += voltage;
        if (voltage > 0) {
          numOptimalInputs++;
          if (lastOptimalInputTimestampAll != 0) {
            totalTimeDiffAll += voltageReadings1[i].timestamp -
lastOptimalInputTimestampAll;
            numTimeDiffPairsAll++;
          }
          lastOptimalInputTimestampAll = voltageReadings1[i].timestamp;
        }
        if (voltage > highestVoltageAll) {
          highestVoltageAll = voltage;
        }
        if (voltage > 0 && voltage < lowestNonZeroVoltageAll) {
          lowestNonZeroVoltageAll = voltage;
        }
      }
    }

    float averageVoltage = numOptimalInputs > 0 ? totalVoltage /
numOptimalInputs : 0;
    if (numOptimalInputs == 0) lowestNonZeroVoltageAll = 0;

    int rawDCVoltage = analogRead(dcVoltagePin);
    float voutDC = rawDCVoltage * voltageConversionFactor;
    float vinDC = voutDC * voltageDividerFactor;
```

```
    receivedString = ""; // Clear for output command
    while (SerialBT.available()) {
      char c = SerialBT.read();
      if (c == '\n') {
        break;
      }
      receivedString += c;
    }

    if (receivedString.length() > 0) {
      if (receivedString == "dc") {
        SerialBT.print("DC Voltage: ");
        SerialBT.print(vinDC, 2);
        SerialBT.println(" V");
        if (previousDCVoltage > 0) {
          float powerSaved = (previousDCVoltage - vinDC) / previousDCVoltage *
100;
          SerialBT.print("Power Saved: ");
          SerialBT.print(powerSaved, 2);
          SerialBT.println(" %");
        } else {
          SerialBT.println("No previous DC voltage to compare.");
        }
        previousDCVoltage = vinDC;
      } else if (receivedString == "avg") {
        SerialBT.print("Average Voltage (All Inputs): ");
        SerialBT.print(averageVoltage, 2);
        SerialBT.println(" V");
      } else if (receivedString == "opt") {
        SerialBT.print("Optimal Inputs (All Inputs): ");
        SerialBT.println(numOptimalInputs);
      } else if (receivedString == "high") {
        SerialBT.print("Highest Voltage (All Inputs): ");
        SerialBT.print(highestVoltageAll, 2);
        SerialBT.println(" V");
      } else if (receivedString == "low") {
        SerialBT.print("Lowest Voltage (All Inputs): ");
        SerialBT.print(lowestNonZeroVoltageAll, 2);
        SerialBT.println(" V");
      } else if (receivedString == "rate") {
        if (numTimeDiffPairsAll > 0) {
          float averageTimeDiffInSeconds = totalTimeDiffAll / numTimeDiffPairsAll
/ 1000.0;
```

```
          SerialBT.print("Average rate of optimal inputs (All Inputs): ");
          SerialBT.println(averageTimeDiffInSeconds, 2);
        } else {
          SerialBT.println("No consecutive optimal inputs found. (All Inputs)");
        }
      } else if (receivedString == "time") {
        SerialBT.print("Total Time: ");
        SerialBT.print(totalTimeInSeconds, 2);
        SerialBT.println(" seconds");
      } else if (receivedString == "peaks") {
        SerialBT.print("Number of Peaks (Highest Input): ");
        SerialBT.println(numPeaks);
      } else if (receivedString == "all") {
        SerialBT.print("DC Voltage: ");
        SerialBT.print(vinDC, 2);
        SerialBT.println(" V");
        if (previousDCVoltage > 0) {
          float powerSaved = (previousDCVoltage - vinDC) / previousDCVoltage *
100;
          SerialBT.print("Power Saved: ");
          SerialBT.print(powerSaved, 2);
          SerialBT.println(" %");
        } else {
          SerialBT.println("No previous DC voltage to compare.");
        }
        previousDCVoltage = vinDC;
        SerialBT.print("Average Voltage (All Inputs): ");
        SerialBT.print(averageVoltage, 2);
        SerialBT.println(" V");
        SerialBT.print("Optimal Inputs (All Inputs): ");
        SerialBT.println(numOptimalInputs);
        SerialBT.print("Highest Voltage (All Inputs): ");
        SerialBT.print(highestVoltageAll, 2);
        SerialBT.println(" V");
        SerialBT.print("Lowest Voltage (All Inputs): ");
        SerialBT.print(lowestNonZeroVoltageAll, 2);
        SerialBT.println(" V");
        if (numTimeDiffPairsAll > 0) {
            float averageTimeDiffInSeconds = totalTimeDiffAll /
numTimeDiffPairsAll / 1000.0;
            SerialBT.print("Average rate of optimal inputs (All Inputs): ");
            SerialBT.println(averageTimeDiffInSeconds, 2);
        } else {
```

```
            SerialBT.println("No consecutive optimal inputs found. (All
Inputs)");
        }
        SerialBT.print("Total Time: ");
        SerialBT.print(totalTimeInSeconds, 2);
        SerialBT.println(" seconds");
        SerialBT.print("Number of Peaks (Highest Input): ");
        SerialBT.println(numPeaks); // Use the renamed variable here
      }
    }
    arrayIndex = 0;
  }
  delay(100);
}
```

Code review:

1. Hardware Setup and Initialization:

- **Voltage Measurement:** The code uses analog pins `32`, `33`, and `35` to read voltage from three separate sources. It also uses pin `34` to measure a DC voltage. These are likely connected to voltage dividers to scale down higher voltages to a range readable by the ESP32's ADC (0-3.3V).
- **Bluetooth Communication:** The ESP32's Bluetooth functionality is used to communicate with a connected device (e.g., a smartphone or computer). The device is named "PUNCH METER".
- **Voltage Conversion:** The code uses `voltageConversionFactor` and `voltageDividerFactor` to convert the raw analog readings (0-1023) to actual voltage values. This is crucial for accurate measurements.
- **Data Storage:** The `VoltageData` struct stores both the voltage reading and the timestamp of the reading. Three arrays (`voltageReadings1`, `voltageReadings2`, `voltageReadings3`) of these structs are used to store the data from each input.

2. Data Collection and Control:

- **Start/Stop Commands:** The code listens for "start" and "stop" commands over Bluetooth. These commands control the data collection process.
- **Data Acquisition:** When data collection is active (`isCollectingData == true`), the code reads the analog values from the three voltage inputs at each loop iteration. The converted voltage values and timestamps are stored in the respective arrays.
- **Circular Buffer:** The `arrayIndex` variable and the modulo operator (`arrayIndex++` and the subsequent `if (arrayIndex >= arraySize) { arrayIndex = 0; }`) implement a circular buffer. This allows the code to continuously collect data without running out of memory, overwriting the oldest data when the buffer is full.
- **Time Measurement:** The `startTime` and `stopTime` variables, along with `millis()`, are used to measure the duration of the data collection period.

3. Data Analysis and Calculations (After "stop" command):

- **Peak Detection:** The code identifies peaks in the highest voltage input using a threshold (`peakThreshold`) and a debounce time (`debounceTime`). This prevents false peak detections due to noise or small fluctuations.
- **Highest Peak Identification:** Determines which of the three inputs has the highest peak voltage.
- **Average Voltage:** Calculates the average voltage across all inputs, considering only non-zero readings (presumably to exclude periods when an input is inactive).
- **Optimal Inputs:** Counts the number of voltage readings across all inputs that are above zero. This is referred to as "optimal inputs" in the code.
- **Highest and Lowest Voltages:** Finds the highest and lowest non-zero voltage readings across all inputs.

- **Average Rate of Optimal Inputs:** Calculates the average time difference between consecutive optimal inputs. This provides a measure of how frequently the inputs are active.
- **DC Voltage Measurement and Power Savings:** Reads the DC voltage from pin 34. If a previous DC voltage reading is available, it calculates and displays the percentage of power saved (assuming a decrease in voltage corresponds to power savings).

4. Bluetooth Output:

- **Command Handling:** After data collection stops, the code waits for further Bluetooth commands.
- **Output Formatting:** The code formats the calculated data into human-readable strings and sends them over Bluetooth.
- **"all" Command:** The "all" command provides a comprehensive summary of all the calculated metrics.

Key Improvements and Considerations:

- **Debouncing:** The peak detection uses debouncing to prevent multiple peak detections from a single event.
- **Circular Buffer:** The use of a circular buffer is efficient for continuous data collection.
- **Clearer Variable Names:** The variable names are generally descriptive, making the code easier to understand.
- **Error Handling:** The code includes basic error handling for Bluetooth initialization.

- **Units:** The code includes units (V, seconds, %) in the output, which is good practice.

Potential Enhancements:

- **Data Logging:** The code could be enhanced to log the raw data to an SD card or other storage for later analysis.
- **Real-time Display:** Instead of waiting for the "stop" command, some basic statistics could be sent over Bluetooth in real-time.
- **More Advanced Signal Processing:** More sophisticated signal processing techniques (e.g., filtering, FFT) could be implemented to extract more information from the voltage readings.
- **Configuration Options:** The code could be made more configurable by allowing the user to set parameters like the `peakThreshold` and `debounceTime` over Bluetooth.