



## Engineering Need

Female youth athletes have **high concussion rates** because most helmets are **designed for adult males**, leading to **poor fit, missed impacts, and higher injury risk**.

## Engineering Objective

**Obj. 1a:** Mechanically optimize a helmet liner to **reduce linear and rotational acceleration**, with liner materials and internal geometries that **maximize impact energy dissipation** while still being light.

**Obj. 2a:** Engineer a **low-cost, real-time impact-sensing alert system**.

## Introduction and Methodology

- Female youth athletes face **disproportionately high concussion rates** (Broshek et al.).
- Standard liners do not account for **hair volume**, causing a physical gap that leads to sensor "swim" and inaccurate data (Patton et al.).
- Commercial monitors cost **\$200–\$500**, which is **inaccessible** for most youth leagues (Siegmond et al.).

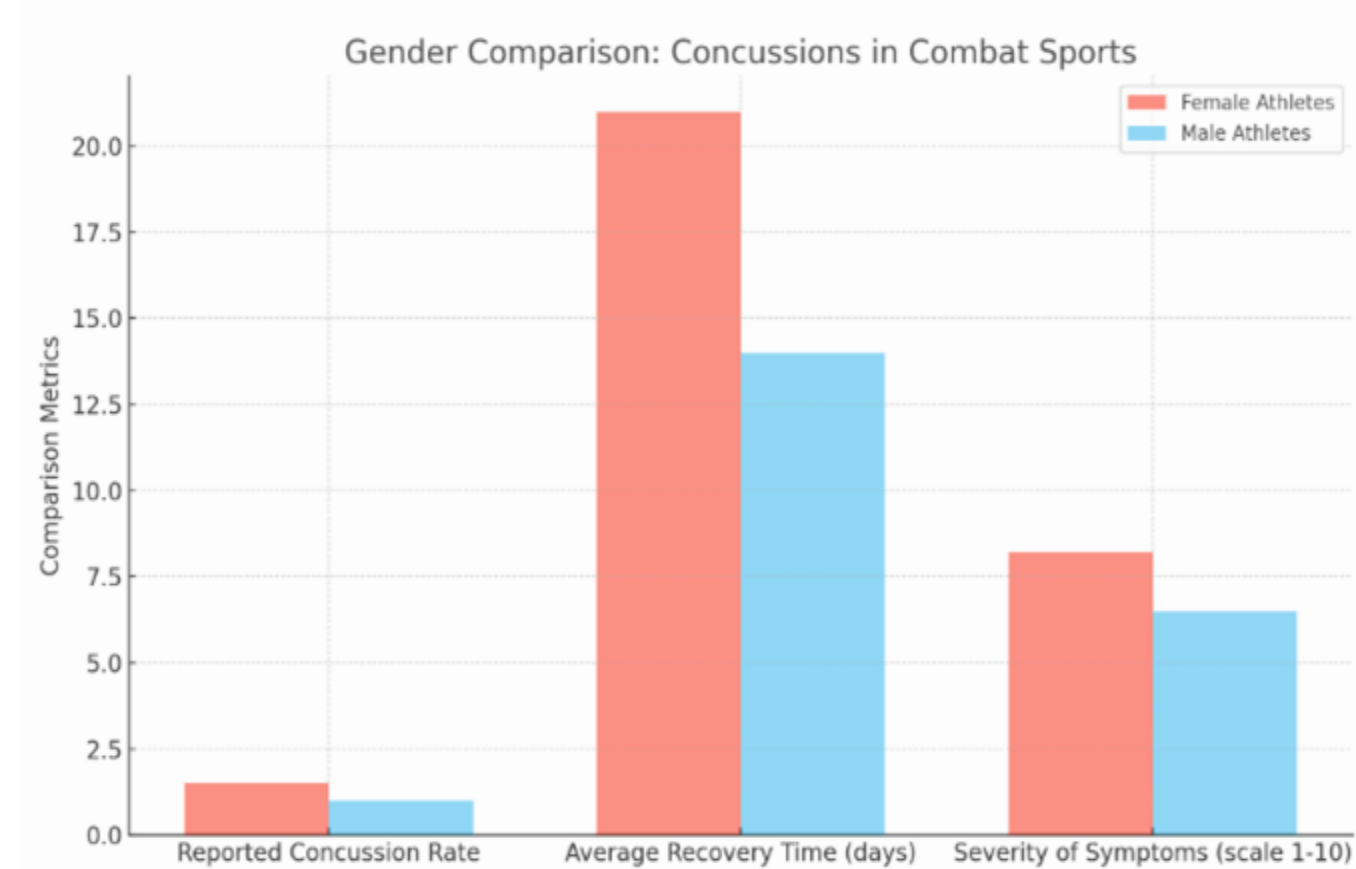
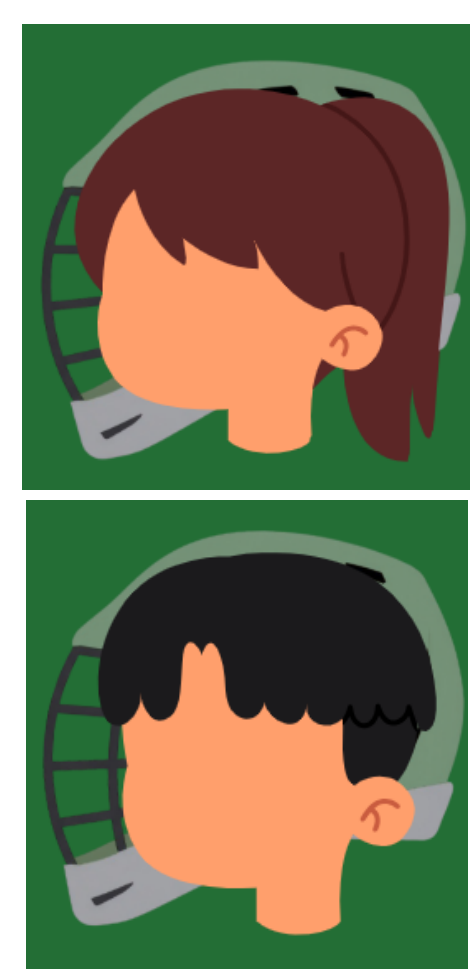


Figure 1: Gender Comparison: Concussions in Combat Sports ("Concussions: Are Female Athletes at Greater Risk?")

### V1

**Design:** 5mm foam liner with Arduino Pro Mini, MPU6050, and 4 piezo discs (A0–A3).

**Logic:** Trigger if >60G linear accel or >15 rad/s rotation on any axis.

**Verification:** Simulations, bench calibration, drop tests, motion simulation; high false positives.

### V2

**Design:** Improved foam mounting for better signal isolation with an ESP32 and MPU6050.

**Logic:** Trigger only when IMU spike.

**Verification:** Simulations and bench testing complete. Real-world testing of device needed.

## Results

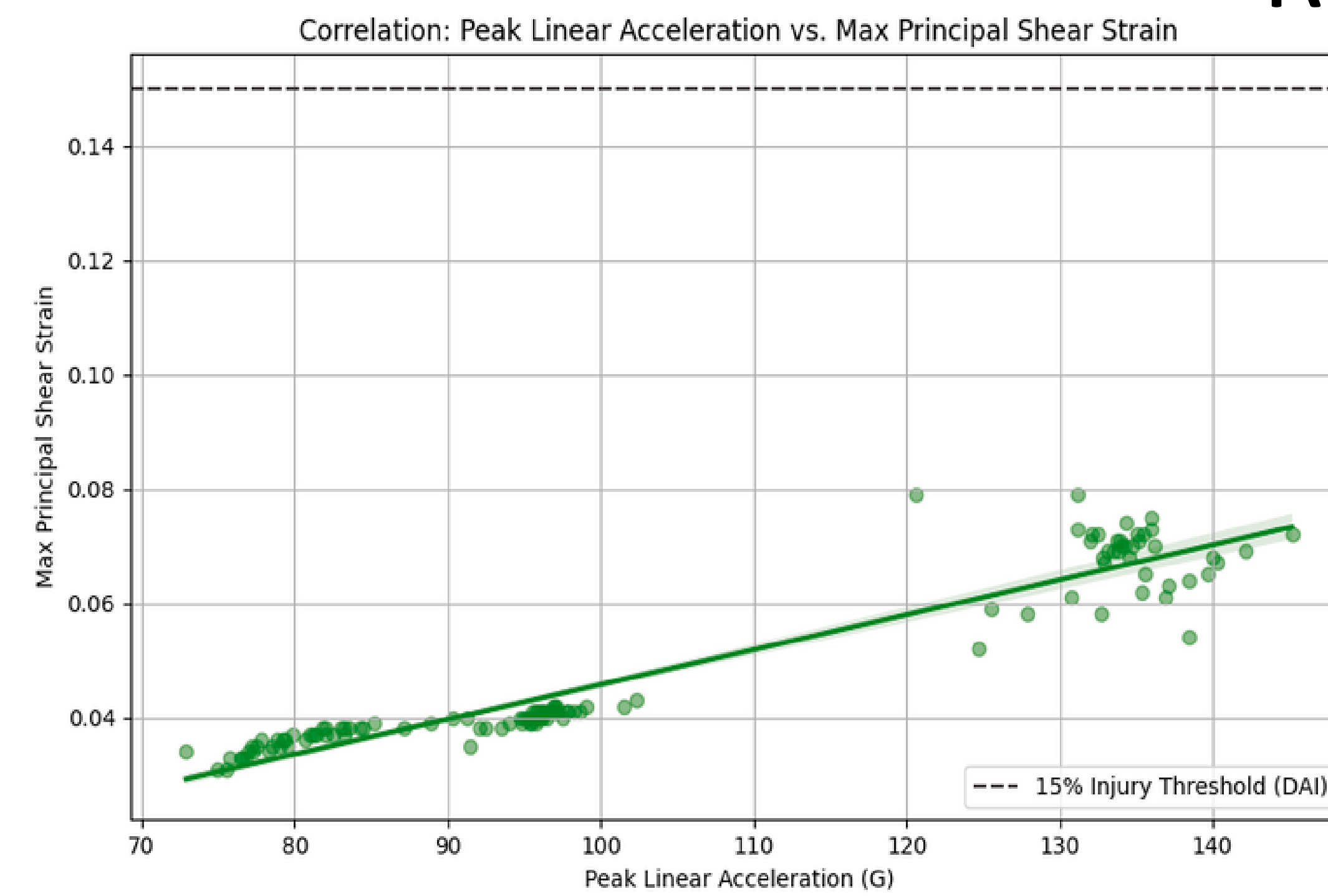


Figure 3: Correlation Analysis of Peak Acceleration and Brain Tissue Shear Strain.

Linear regression ( $R^2 = 0.89$ ) demonstrating the relationship between impact magnitude and Max Principal Shear Strain ( $\gamma_{max}$ ) using a fusion simulation. The dashed line denotes the 15% injury threshold for DAI; the prototype consistently maintained tissue strain at a mean of 7.6%, representing a nearly 50% safety margin.

### PLA Reduction by Impact Site

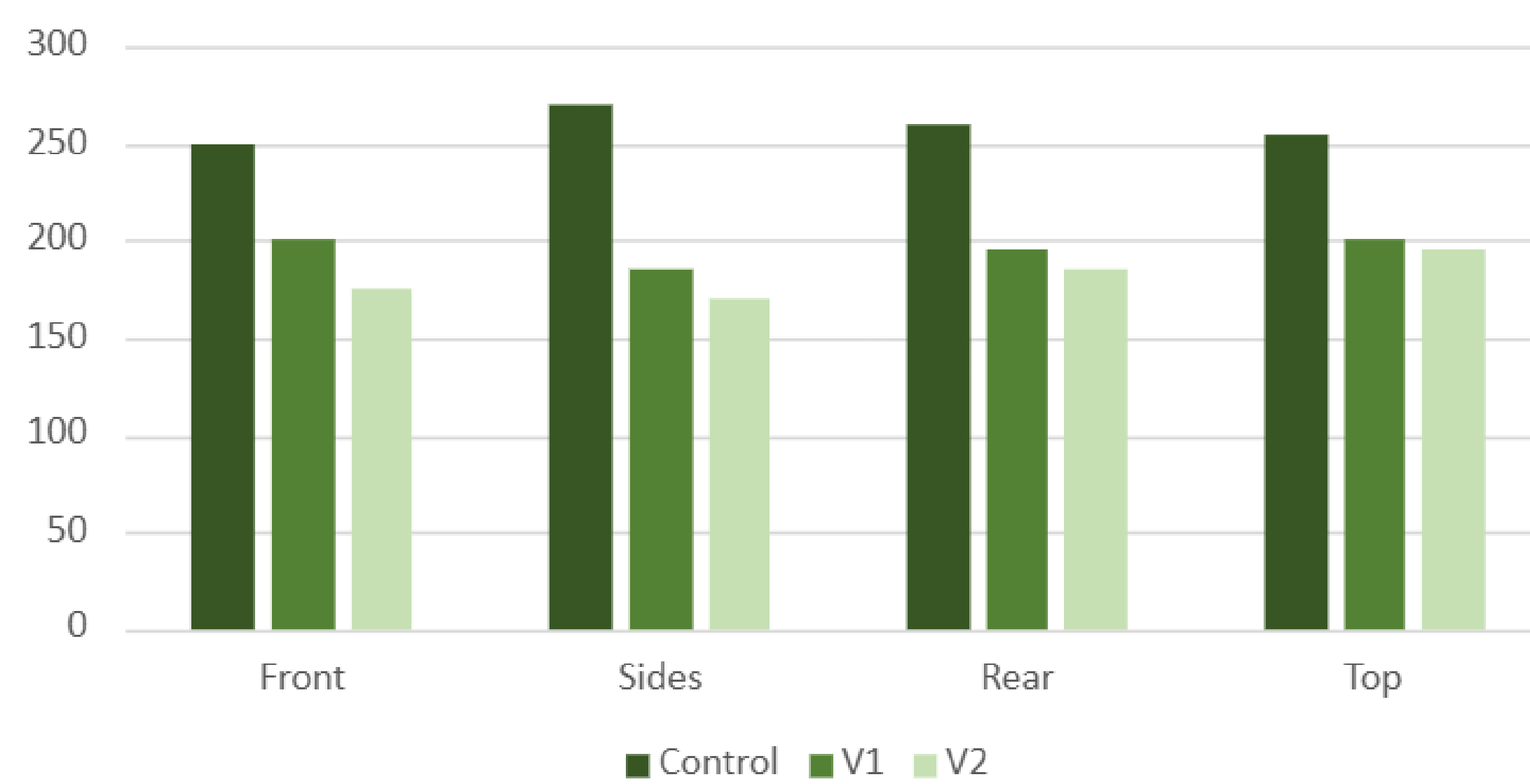


Figure 4: Peak Linear Acceleration (PLA) Reduction by Impact Site

Comparison of impact forces between a standard helmet (Control) and the two versions of the liner (V1 and V2) using simulated data. Data labels represent mean G-force across  $n=10$  trials per site through Fusion Simulation.

### Simulated Real World Testing for V1

| Activity Type       | Trials | Max Piezo (A0-A3) | Max Gyro (deg/s) | Alert Triggered? |
|---------------------|--------|-------------------|------------------|------------------|
| Sprinting/Stopping  | 10     | 120 ± 15          | 178 ± 28         | NO               |
| Wall Ball (Passing) | 10     | 232 ± 38          | 141 ± 18         | NO               |
| Incidental Bump     | 10     | 603 ± 24          | 302 ± 37         | YES              |
| High-G Impact       | 20     | 710 ± 45          | 440 ± 52         | YES (100%)       |

Table 1: V1 system's ability to distinguish between athletic movement and injurious impact.

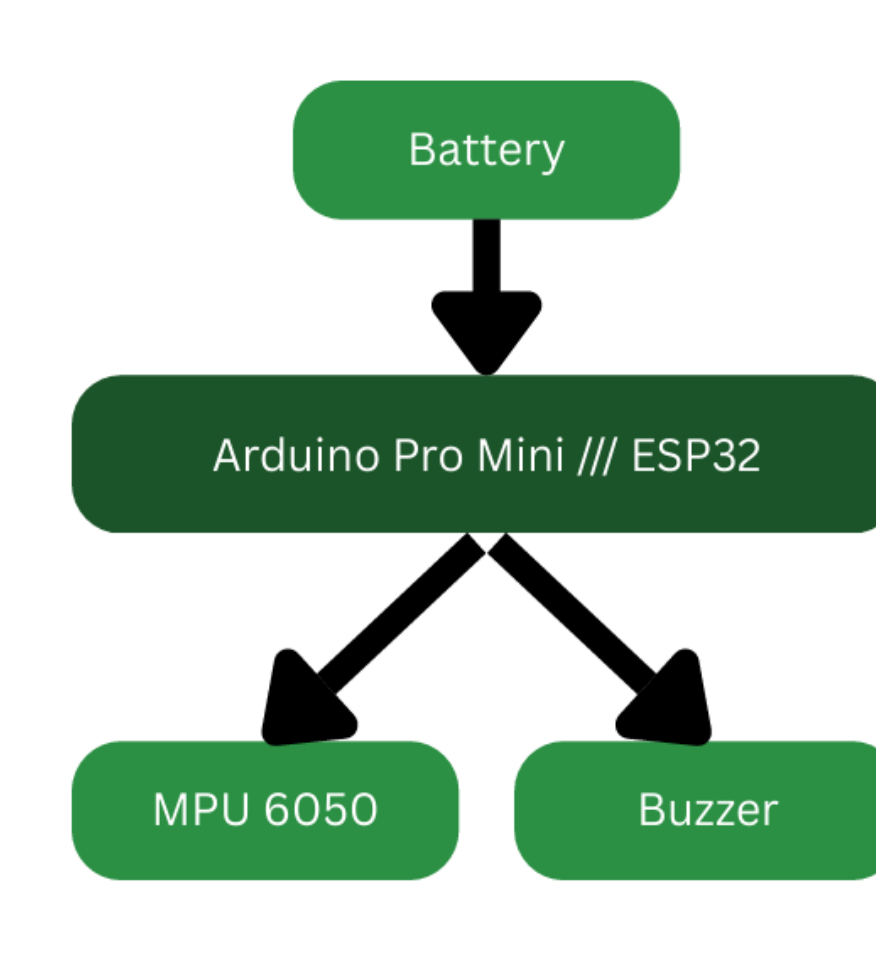


Figure 5: V1 and V2 Circuitry Comparison

- Arduino Pro Mini**
  - Good for simple threshold systems
  - Limited processing for fusion logic
  - Needs extra modules for wireless
- ESP32**
  - Fast sensor fusion + filtering
  - Built-in BLE/WiFi for live data
  - Handles complex multi-sensor logic

## V1 Code

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;
#define BUZZER 18
#define PIEZO1 34
#define PIEZO2 35
#define PIEZO3 32
#define PIEZO4 33

float G_THRESHOLD = 2.0;
int PIEZO_THRESHOLD = 500;
int impactCount = 0;

void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22);
  pinMode(BUZZER, OUTPUT);
  Serial.println("V1 System Starting...");
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  mpu.initialize();
  if (mpu.testConnection()) {
    Serial.println("IMU READY");
  } else {
    Serial.println("IMU FAILED");
    while (1);
  }
}

void loop() {
  int16_t ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  float ax_g = ax / 16384.0;
  float ay_g = ay / 16384.0;
  float az_g = az / 16384.0;
  float total_g = sqrt(ax_g * ax_g + ay_g * ay_g + az_g * az_g);

  int p1 = analogRead(PIEZO1);
  int p2 = analogRead(PIEZO2);
  int p3 = analogRead(PIEZO3);
  int p4 = analogRead(PIEZO4);
  if (total_g > G_THRESHOLD || p1 > PIEZO_THRESHOLD ||
      p2 > PIEZO_THRESHOLD || p3 > PIEZO_THRESHOLD ||
      p4 > PIEZO_THRESHOLD) {
    impactCount++;
    Serial.println("IMPACT DETECTED (V1)");
    digitalWrite(BUZZER, HIGH);
    delay(100);
    digitalWrite(BUZZER, LOW);
    delay(50);
  }
}
```

Figure 6: V1 Code for Arduino

## V2 Code

```
#include <Wire.h>
#include <MPU6050.h>
#include <math.h>

MPU6050 mpu;

float IMPACT_THRESHOLD = 0.6;
unsigned long lastImpact = 0;
int impactCount = 0;
#define BUZZER 18
float gBuffer[5] = {0.0, 0.0, 0.0};
int idx = 0;

float smoothG(float newVal) {
  gBuffer[idx] = newVal;
  idx = (idx + 1) % 5;
}

float sum = 0;
for (int i = 0; i < 5; i++) sum += gBuffer[i];
return sum / 5.0;
}

void setup() {
  Serial.begin(115200);
  Wire.begin(21, 22);
  pinMode(BUZZER, OUTPUT);
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  mpu.initialize();
  if (mpu.testConnection()) {
    Serial.println("IMU READY");
  } else {
    Serial.println("IMU FAILED");
    while (1);
  }
}

void loop() {
  int16_t ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  float ax_g = ax / 16384.0;
  float ay_g = ay / 16384.0;
  float az_g = az / 16384.0;
  float total_g = sqrt(ax_g * ax_g + ay_g * ay_g + az_g * az_g);
  float filtered_g = smoothG(total_g);
  float net_g = fabs(filtered_g - 1.0); //gravity baseline
  static float last_g = 1.0;
  float spike = fabs(filtered_g - last_g);
  last_g = filtered_g;

  if (net_g > IMPACT_THRESHOLD
      && spike > 0.4
      && millis() - lastImpact > 800) {
    lastImpact = millis();
    impactCount++;
    Serial.println("IMPACT DETECTED");
    tone(BUZZER, 2200, 120);
    delay(150);
    noTone(BUZZER);
    delay(30);
  }
}
```

Figure 7: V2 Code for ESP32

### Engineering Performance vs. Safety Standards

| Performance Metric                        | Experimental Mean ( $\mu$ ) | Safety/Target Threshold                   | Status |
|---|-----------------------------|---|--------|
| Peak Linear Accel (PLA)                   | 133.0 G                     | < 250 G (Injury Limit)                    | PASS   |
| Peak Rotational Accel (PRA)               | 3142 rad/s <sup>2</sup>     | < 6000 rad/s <sup>2</sup> (Tierney, 2024) | PASS   |
| Max Brain Shear Strain ( $\gamma_{max}$ ) | 7.60%                       | 15.0% (DAI Threshold)                     | PASS   |
| Max Intracranial Pressure                 | 131 kPa                     | < 150 kPa                                 | PASS   |
| System Build Cost                         | \$28.42                     | < \$50.00                                 | PASS   |

Table 1: Comparison of Experimental Prototype Performance against established Concussion Biomechanics thresholds for V1.

## Methodology Flowchart

| Phase 1: Research & Preparation   | Phase 2: Design and Prototyping | Phase 3: Validation and Analysis    |
|-----------------------------------|---------------------------------|-------------------------------------|
| 1. Define Safety Goals            | 1. Component Sourcing           | 1. Controlled Drop Testing          |
| 2. Design Iteration and Selection | 2. Liner Manufacturing          | 2. Rotational and Real-Life Testing |
| 3. Simulation Testing             | 3. Sensor Integration           | 3. Data Analysis                    |
|                                   |                                 | 4. Refinement                       |

## Future Work

- PCB Miniaturization:** Moving from a breadboard prototype to a custom, flexible Printed Circuit Board (PCB) to reduce the system's footprint.
- Material Innovation:** Testing Non-Newtonian foams (like D30) that remain soft for comfort but harden instantly upon impact to maximize energy dissipation.
- Wireless Ecosystem:** Integrating Bluetooth Low Energy (BLE) to transmit real-time "Impact Heatmaps" directly to a parent's or coach's smartphone app.

## Conclusion

- V1** demonstrated **basic feasibility** of detecting helmet impacts using MPU6050 and piezo sensors, but **suffered from high false positives** due to motion noise and piezoelectric sensors.
- V2 improved** accuracy through **sensor fusion, signal filtering, and refined code** for the IMU.
- System performance became significantly **more stable** compared to the initial prototype.
- V2 was validated through simulations and controlled bench testing.
- Full real-world field data collection (e.g., live gameplay conditions and activities) was not yet completed.
- Results indicate **strong potential** for a lightweight, low-cost helmet impact detection system **with further refinement**.