

# Wearable Basketball Jumpshot Mechanics Analyzer

Team 78

Team Members:

Aiden Zack, Arjun Vyas, Tanmay Nair

TA:

Mingrui Liu

ECE445

## Introduction

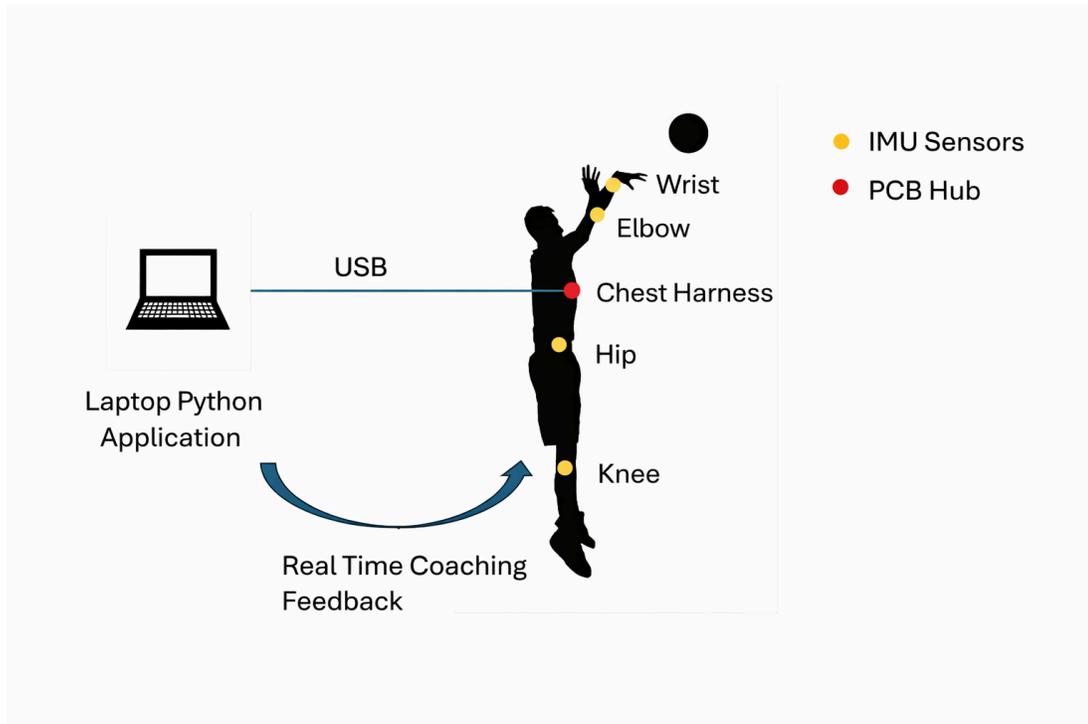
### 1. Problem

- a. A basketball jumpshot involves a chain of body mechanics that requires coordination from your feet to your wrist to achieve a simple goal that is much more complicated than what the average person sees: Making the ball go in the hoop. So many players across the world have exhibited different mechanics in their jumpshot, so when they reach out to coaching for help, they tend to hear subjective advice that is often inconsistent, difficult to put into numbers, and, more importantly, harder to fit into the player's perspective. Existing resolutions utilize shot trajectory and do not tap into the biomechanics that reside in the shooter. In essence, this leads to players lacking reliable, repeatable data to identify points of improvement in their mechanics, address consistency issues, and record progress.

### 2. Solution

- a. This project will implement a system dedicated to quantifying a user's basketball jumpshot by analyzing the consistency and timing of the "kinetic chain". It starts with node sensors that will be worn on the user's shooting wrist, shooting elbow, hip, and the knee of the user's shooting side. These sensors will hold an IMU, microcontroller, and wired communication. The knee sensor will focus on lower-body motion and take measures related to shot success, such as the timing of the jump and how much the knee flexes to determine the dip. The wrist sensor will look at the upper-body mechanics that finish out the shot, like the angular velocity and release timing of the wrist, along with how high it sits for the follow-through. The hip sensor will measure vertical displacement. The elbow sensor will measure the angular velocity. These 4 data nodes will be synchronized in our system, extracted for timing measures like jump-to-release, and then processed for evaluation and feedback. This will focus on the repeatability and timing of the user's body mechanics, providing user-oriented assistance that adjusts as the user progresses.

### 3. Visual Aid



#### 4. High-Level Requirements

##### a. Temporal Synchronization and Resolution:

- i. This system will utilize a hardware FSYNC trigger to align time stamps of samples, along with a chip select to match incoming data to its corresponding IMU sensor. As a result, the system must meet (at a minimum) a sampling frequency of 200 Hz across all 4 IMU nodes. A temporal skew across sensors should be kept to less than 1 ms to match the FSYNC trigger. Meeting all of these specifications will enable the system to ensure that even high-velocity movements, like a wrist flick, will be sampled with enough data points and within a synchronized timeline to allow for latency calculations between movements.

##### b. Measurement Range and Stability

- i. Each IMU subsystem must support a gyroscopic dynamic range of up to  $\pm 2000$  degrees/second and an accelerometer range of at least  $\pm 16g$  due to explosive movements that may disrupt the signal and sensor performance. This is especially important at peak angular velocities of the wrist flick and high-G impacts like the shooter's landing. This will allow for efficient data collection and ensure its usability for the Python program throughout the entire jump shot.

##### c. Proper closure of the Feedback Loop

- i. Our Python backend program must identify each data point to its corresponding sensor and time stamp, and interpret "kinetic milestones" such as maximum knee dip and peak jump acceleration. This will be done using peak detection and zero-crossing algorithms (precision of  $\pm 5$  ms) to calculate temporal offsets. These offsets, along with the 6-axis IMU data, which we can use to quantify the magnitude of movements, will be stored according to the user's inputter

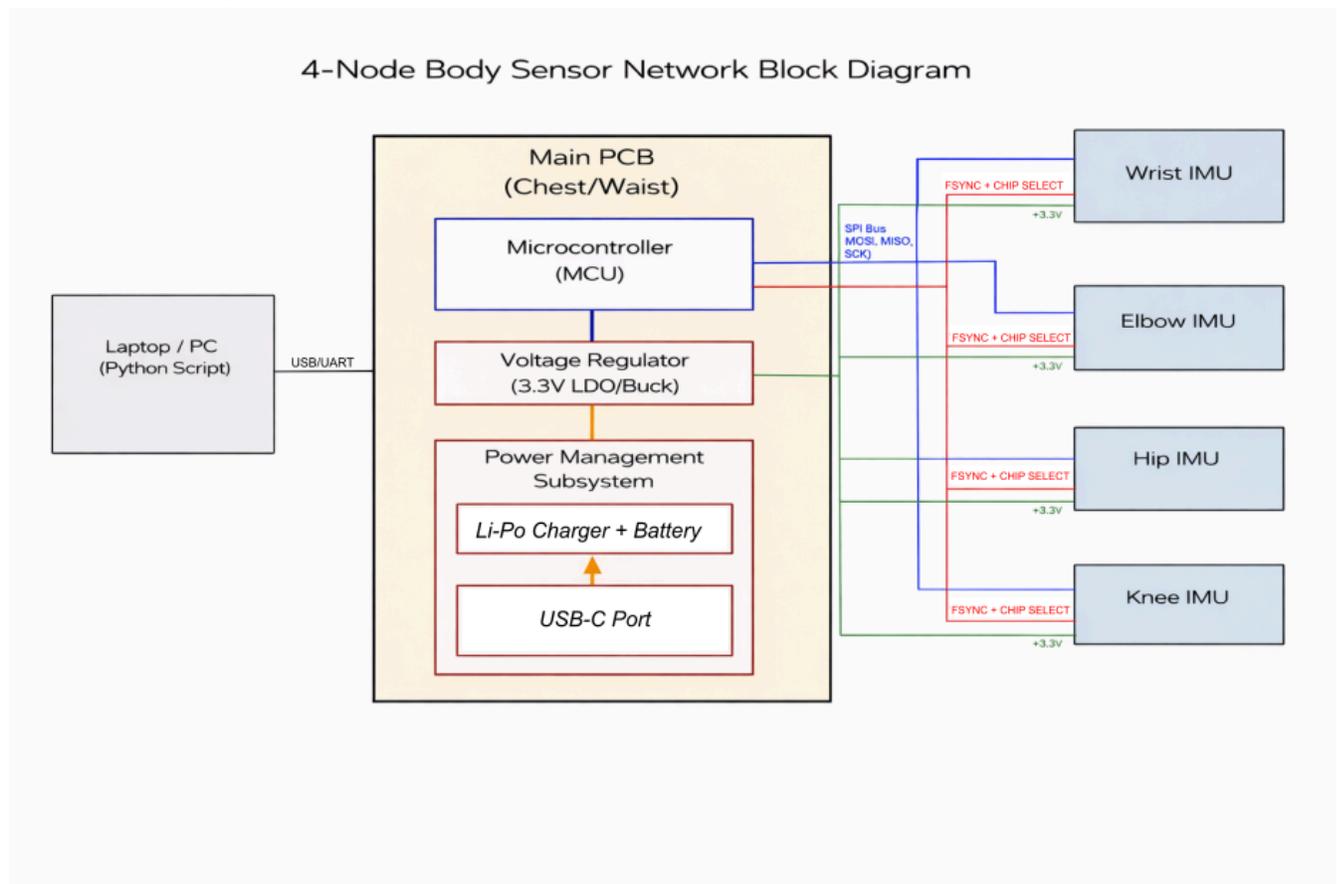
outcomes (make or miss). From here, the program will validate the shot quality based on these trials, producing interpretable feedback that encompasses timing, power, and form errors.

d. **Note:** Demo Procedure

- i. Demo will include a video demonstration of a “bad” shot as well as a “good” shot to match the data we collect. This provides significance and ease of inference to the telemetry data, which will help justify the feedback our Python program will suggest.

## Design

### 1. Block Diagram



### 2. Subsystem Overview

#### a. The Main PCB Hub

This houses our power management subsystem, microcontroller, and USB-to-UART connection to our Python program via PC. Our Power Management Subsystem starts at the USB-C Port, which takes in a 5V input to feed the Li-Po Charger and the subsequent Li-Po Battery. The battery voltage is then stabilized into the 3.3V Voltage LDO Regulator, which is then distributed to the microcontroller and 4 IMU Sensor Nodes.

Along with this, an Arduino Nano ESP32 will serve as our MCU. Its 32-bit Dual-Core processor, along with its 240 MHz clock speed, allows it to manage the SPI bus and FSYNC timing along with the

USB/UART communication simultaneously. With its FSYNC signal, it can broadcast to our 4 sensor nodes and ensure simultaneous latching of data. The SPI Bus is managed here as well, with Chip Select (CS) lines toggled to sequentially gather data from the Wrist, Elbow, Hip, and Knee sensors. This is all then packaged into synchronized telemetry data (with associated time stamps and source labels) and then transferred to the PC via its internal USB-to-UART bridge.

Our Python-based backend via PC takes our raw telemetry data and transforms it into a readable format. By smoothing signals, storing magnitudes of movement, and calculating temporal offsets, the “Success Profile” can be built off of the user’s trial jumpshots, which defines a valid database that the program can then compare to, correlating biomechanical form and timing with real-time critical feedback.

#### b. Sensor Subsystem

This encompasses our 4 6-axis IMUs that capture the full “Kinetic Chain” involved in a person’s basketball shot. These 4 sensors are wired and connected to the same SPI Bus and common FSYNC line, but hold their own Chip Select connection from the MCU. Each sensor plays a different role in the chain and, as a result, will have its raw biomechanical data processed in Python based on the key movement they involve (i.e., the knee sensor will primarily focus on the Y-Axis angular velocity for dip and Z-Axis acceleration and displacement for jump). Each sensor’s primary role is expanded on in the Subsystem Requirements section.

#### c. Subsystem Requirements:

- i. IMU sensor on the Knee
  1. Detects the initial energy burst and launch with the dip and jump
  2. Must handle high-G impact from the landing that the accelerometer must include within its range of measurement
  3. Sits along the femur (side of the quad)
  4. Significant Metrics: Z-Axis Acceleration, Displacement (Jump), and Y-Axis Angular Velocity (Degree of Dip)
- ii. IMU sensor on the Hip
  1. Serves as a connection between lower-body power generation and stability up and through to your wrist
  2. Sits at the center of the lower back
  3. This sensor will focus more on lowering the noise floor and being more sensitive to subtle shifts, versus having a high variability in its measurement range
  4. Significant Metrics: Z-Axis Displacement (in support with Knee IMU), Angular Pitch and Roll (prioritize high accuracy; subtle changes must be measured)
- iii. IMU sensor on the Elbow
  1. Rapid extension → Gyroscope must support  $\pm 2000$  degrees/second
  2. Place on the lateral side of the upper arm, right above the elbow
  3. Significant Metric: Angle of Extension relative to the timing of the jump
- iv. IMU sensor on the Wrist

1. Involves the “flick” (fastest part of the shot) → smallest critical window of < 30 milliseconds
  2. Should be polled first in SPI sequence, or assigned the highest priority in Python code
  3. Requires a gyroscope capable of capturing  $\pm 2000$  degrees/second
  4. Keep lightweight
  5. Significant Metrics: X, Y-axes displacement (to support extension of elbow), Z-axis displacement (to support jump measurement), gyroscope for wrist angular velocity and rotation
- v. PCB Hub/Computer
1. Centerpiece of all external component communications
  2. IMU sensors will communicate with the MCU via SPI
  3. MCU will send data to the computer through USB/UART
  4. The system will maintain a UART baud rate of 115,200 to stream a 48-byte synchronized data packet to Python without buffer overflow/packet loss. (Low latency, near real-time feedback)
  5. Data will be interpreted in Python through closed-loop feedback communications with the user
- b. Tolerance Analysis
- i. One aspect that does pose a risk in our project is how we intend to synchronize data from our 4 sensor nodes and provide it to the microcontroller. Our solution involves using an FSYNC trigger, which is proven to be effective. The microcontroller will run a continuous cycle of sending an FSYNC pulse to the sensors, latching the sensors to grab the 6 data points at that instantaneous moment, polling the data from the sensors sequentially using chip selects, bundling this data, and sends it over to the PC. The SPI will then wait, depending on the sample rate, before cycling again.
  - ii. For instance, given a 200Hz clock, this cycle occurs once every 5 milliseconds, with ~0.1 milliseconds spent on reading from the sensors. Given that a basketball shot takes ~600-800 milliseconds and that even the fastest motion, like a wrist flick, takes 30 milliseconds (meaning we can capture 5-6 samples within that timespan), we can effectively pull many samples of synchronized data during the peak movements.
  - iii. If the FSYNC trigger is not working correctly, there are ways around this to ensure the system is still functional. Sampling rates can be increased in order for the data to be more clustered, essentially creating a physical synchronization.

### 3. Ethics, Safety, and Societal Impact

#### a. Ethical Considerations

- i. The ethical risk of injury resulting from changes in shooting mechanics relates directly to IEEE Code of Ethics Principles 1, 3, and 6. To avoid ethical breaches, the system is designed to prioritize user safety by limiting feedback to observational metrics focused on timing consistency and repeatability, rather than instructing users to make abrupt or corrective changes to their mechanics. In

accordance with Principle 3, the system avoids absolute claims about a “proper” shooting form, as there is no “perfect” form, and clearly communicates its limitations. To further prevent harm under Principle 6, safeguards are implemented to discourage abrupt form changes, including warnings against overuse and guidance to discontinue use if discomfort or pain occurs. These design choices ensure that feedback supports user awareness without encouraging unsafe changes that exceed the physical limitations of the user’s body.

b. Safety Considerations

- i. This system involves multiple electronic sensors worn directly on the body during dynamic athletic movement, which introduces several potential safety concerns. Electrical risks include battery overheating, short circuits, and improper charging of batteries. Mechanical risks include sensor detachment, sharp enclosure edges, and interference with natural joint motion, which could cause discomfort during use. To mitigate these concerns, all electronics operate at low voltage, and sensors are enclosed and mounted to the body. The system is intended for short-duration, dry, indoor use to further reduce electrical and mechanical hazards. Another potential safety concern relates to the feedback provided by the system, which may lead users to modify their shooting mechanics. Changes in form that do not align with an individual’s physical limitations may introduce abnormal joint movement or muscle strain, potentially resulting in injury. This system does not involve high voltage, hazardous materials, or unsafe current levels applied to the human body. While batteries are used, they are low-capacity and handled in accordance with established battery safety guidelines.

c. Impact of Engineering Solution

- i. From a societal and global perspective, this solution promotes more objective and accessible feedback for athletic training by allowing athletes to better understand their own shooting mechanics and improve self-guided training. By focusing on timing and consistency rather than subjective judgment, the system supports safer, more informed improvement compared to traditional subjective coaching. Globally, basketball is widely considered one of the most popular sports, and this solution can serve as a valuable training tool when access to coaches or training resources is limited or unavailable.