PROJECT OVERVIEW

Title:Posture Correcting Belt Using Arduino and MPU6050 Accelerometer

1. Introduction

In recent years, the incidence of poor posture among individuals has increased significantly. This trend is largely attributed to prolonged sedentary behavior, extensive use of digital devices, and non-ergonomic workspaces. Over time, bad posture can lead to serious health issues such as musculoskeletal disorders, chronic back pain, and reduced lung capacity. Addressing this problem requires a proactive and continuous approach to posture correction.

To help users develop better posture habits, this project introduces a wearable solution in the form of a posture-correcting belt. The belt uses embedded electronics to detect poor posture and provide immediate feedback to the user, thereby promoting spinal health and encouraging long-term behavioral change.

2. Objective

The primary goal of this project is to create a functional and user-friendly posture correction device that:

- Continuously monitors the user's spinal alignment in real-time.
- Detects when the posture deviates from an optimal or neutral position.
- Waits for a sustained deviation (over 4 seconds) before triggering an alert to avoid unnecessary false positives.
- Provides haptic feedback through a vibration motor, gently nudging the user to correct their posture.

- This device is particularly beneficial for individuals who spend extended periods sitting, such as office workers, students, and drivers.

3 Real-World Problem and Possible Solutions

Problem Statement:

Poor posture is a widespread issue in modern society, leading to chronic back pain, spinal issues, and reduced quality of life. Addressing this requires continuous monitoring and corrective action.

Possible Solutions:

1. Physical Therapy and Chiropractic Care

Pros: Expert supervision, tailored exercises, long-term benefits.

Cons: Expensive, time-consuming, requires regular visits.

2. Posture Braces

Pros: Inexpensive, easy to wear, provides passive support.

Cons: Can cause muscle dependency, does not alert user actively, fixed tightness may be uncomfortable.

3. Ergonomic Furniture

Pros: Improves posture passively during sitting, enhances comfort.

Cons: Costly, not portable, limited to use in specific environments (like office or home).

4. Mobile Apps with Camera-Based Tracking

Pros: Convenient, uses existing devices.

Cons: Limited accuracy, requires proper lighting and position, not real-time for wearable use.

5. Smart Wearables (Our Project)

Pros: Real-time monitoring, immediate feedback, wearable and portable, customizable settings.

Cons: Requires initial calibration, electronic components need care, battery maintenance.

Justification for Our Project:

Among all the solutions, our posture-correcting belt offers the best balance of affordability, mobility, and effectiveness. Unlike braces, it trains the user rather than enforcing posture externally. Unlike physical therapy, it works in real-time and is always with the user. Unlike furniture, it moves with the person. Its haptic feedback gently encourages behavior correction without causing discomfort or dependency.

This proactive and lightweight design allows for daily use, adaptable to a wide range of users and environments, making it an ideal and innovative solution for modern posture correction.

4. System Overview

The system comprises three core components: a microcontroller (Arduino Uno), a motion sensor (MPU6050), and a feedback mechanism (coin vibration motor). The MPU6050 provides six degrees of motion sensing (three axes of acceleration and three axes of gyroscope data). The Arduino processes this sensor data to detect deviations from the calibrated baseline posture.

Once a poor posture is detected and maintained for a threshold period (e.g., 4 seconds), the Arduino activates the vibration motor. This motor, placed near the user's spine, vibrates in a pulsating pattern (1 second on, 1 second off), reminding the user to correct their posture.

The entire setup is integrated into an elastic belt, allowing the user to wear it around the chest for optimal sensor placement and comfort.

5. Components Used

1. Arduino Uno: The central microcontroller used for data processing, sensor reading, and output control.

2. MPU6050 Sensor: Combines a 3-axis gyroscope and a 3-axis accelerometer, essential for tracking orientation and movement.

3. Coin Vibration Motor: Provides tactile feedback when bad posture is detected.

4. NPN Transistor (e.g., 2N2222): Acts as a switch to control the motor since the Arduino cannot provide sufficient current directly.

5. Flyback Diode (1N4007): Protects the transistor and Arduino from voltage spikes caused by the inductive load of the motor.

6. Electrolytic Capacitor (100uF): Stabilizes voltage and reduces electrical noise.

7. Resistors (220 ohm): Used in the transistor base circuit to limit current and protect components.

8. 9V Battery with DC Barrel Jack: Supplies portable power to the system.

9. Elastic Chest Belt: Ensures the system is securely and comfortably worn by the user.

10. Breadboard and Jumper Wires: Used for prototyping and making electrical connections between components.

6. Working Principle

The system begins by calibrating the user's current posture during the initial startup. This baseline angle serves as the reference for detecting posture deviations. As the user moves, the MPU6050 continually sends orientation data to the Arduino.

If the orientation deviates beyond a set angular threshold (e.g., 10°), and this deviation is maintained for longer than four seconds, the Arduino interprets this as bad posture. In response, it activates the vibration motor using a transistor switch. The motor vibrates intermittently (1s on, 1s off) to alert the user.

Once the posture is corrected and returns within acceptable limits, the Arduino turns off the motor. This method ensures timely feedback without being overly sensitive to brief movements.

Total cost

SI NO	COMPONENTS	QUANTITIES	PRICE
1	Elastic Belt	1	255
2	Arduino Uno	1	250
3	UNO Cable	1	35
4	Accelerometer MPU6050	2	240
5	Breadboard	1	80
6	Jumper Wires	20	40
7	1N4007 Diodes	1	2
8	2N2222 Transistors	1	9
9	220 Ohm Resistors	1	10
10	9V Battery	1	20
11	Coin Vibration Motor	1	15
12	Ceramic Capacitor	1	1
13	9V Battery Jack	1	20
			Total: 977

7. Circuit Design

MPU6050 to Arduino: The sensor connects via I2C—SDA to A4, SCL to A5. VCC is powered with 3.3V or 5V, and GND is connected to Arduino ground.

Motor Control Circuit: The Arduino's digital pin sends a signal to a resistor, which connects to the base of an NPN transistor. The motor's negative terminal is connected to the transistor's collector, and the emitter is grounded. The motor's positive terminal connects to the battery's positive rail.

Flyback Diode: Placed across the motor terminals to handle inductive spikes when the motor is turned off.

Capacitor: Placed near the motor to buffer voltage and prevent current fluctuations.

This design allows for safe operation of the motor without drawing excessive current from the Arduino, and it prevents component damage due to electrical noise or spikes.

IMPLEMENTATION CODE

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

MPU6050 mpu;

const int motorPin = 8;
float baselineAngleX = 0.0;
float baselineAngleY = 0.0;
const float angleThreshold = 10.0; // Degrees of tilt allowed
const unsigned long badPostureDelay = 2000; // 4 seconds
const unsigned long vibrationCycle = 1000;  // 1 second ON/OFF

unsigned long postureStartTime = 0;
unsigned long lastVibrationToggle = 0;
bool postureIsBad = false;
bool vibrating = false;
bool motorState = false;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();

  pinMode(motorPin, OUTPUT);
  digitalWrite(motorPin, LOW);

  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed");
    while (1);
  }

  Serial.println("Calibrating... Keep good posture.");
  delay(3000); // Hold still

  int16_t ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  baselineAngleX = atan2(ay, az) * 180 / PI;
  baselineAngleY = atan2(ax, az) * 180 / PI;
  Serial.println("Calibration complete.");
}

void loop() {
  int16_t ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  float angleX = atan2(ay, az) * 180 / PI;
  float angleY = atan2(ax, az) * 180 / PI;

  float deviationX = abs(angleX - baselineAngleX);
  float deviationY = abs(angleY - baselineAngleY);

  // Print continuous data
  Serial.print("Angle X: "); Serial.print(angleX);
  Serial.print(" | Angle Y: "); Serial.print(angleY);
  Serial.print(" | Dev X: "); Serial.print(deviationX);
  Serial.print(" | Dev Y: "); Serial.println(deviationY);

  bool badPosture = (deviationX > angleThreshold || deviationY > angleThreshold);
  unsigned long currentTime = millis();

  if (badPosture) {
    if (!postureIsBad) {
      postureIsBad = true;
      postureStartTime = currentTime;
```

```
    } else if ((currentTime - postureStartTime >= badPostureDelay)) {
      vibrating = true;

      // Toggle vibration every 1 second
      if (currentTime - lastVibrationToggle >= vibrationCycle) {
        motorState = !motorState;
        digitalWrite(motorPin, motorState ? HIGH : LOW);
        lastVibrationToggle = currentTime;

        Serial.println(motorState ? ">> VIBRATION ON" : ">> VIBRATION OFF");
      }
    }
  } else {
    postureIsBad = false;
    vibrating = false;
    digitalWrite(motorPin, LOW);
    motorState = false;
    Serial.println(">> Posture OK. Vibration stopped.");
  }

  delay(100);
}
```

8. Mounting and Wearability

The compact hardware is mounted on a mini breadboard and affixed to an elastic belt. The belt is worn snugly around the chest with the sensor centered along the spine. This configuration provides optimal data accuracy and user comfort.

To ensure usability:

- All components are securely fastened to prevent disconnection.
- Wires are arranged neatly to avoid tangling.
- The belt allows easy wearing and removal without compromising the sensor’s position.
- The system is light, non-intrusive, and suitable for daily use without impeding physical activity.

9. Applications

This belt has wide applicability in various domains:

Workplace Ergonomics: For office workers who sit for extended hours.

Education: For students who adopt poor posture while studying.

Healthcare: For individuals undergoing rehabilitation or physical therapy.

Fitness: For athletes looking to maintain correct spinal alignment during workouts.

Elderly Care: For seniors who need posture support to prevent falls or injury.

10. Conclusion

This posture correcting belt provides an innovative and affordable solution for real-time posture monitoring. It combines sensors, microcontrollers, and feedback mechanisms in a wearable format that encourages better habits and reduces the risk of postural injuries.

Through intelligent detection and minimal disruption, users are guided to maintain good posture consistently. The prototype can be further improved and commercialized for wide-scale use.

11. Future Improvements

The prototype can be enhanced with several features:

Bluetooth Connectivity: To sync with a smartphone app for alerts and data visualization.

Rechargeable Battery: Integrating Li-ion batteries with USB charging for convenience.

Data Logging: Track user posture history for insights and improvements.

Flexible PCB: To reduce size and make the device less obtrusive.

Enclosure Design: A protective casing to safeguard components and improve aesthetics.

Machine Learning Integration: For predictive alerts and adaptive sensitivity settings based on user behavior.

These improvements would make the system smarter, more user-friendly, and suitable for a commercial market.

End of Report