CODE:

```
#include <Wire.h>
#include <Adafruit_GPS.h>
#include <SoftwareSerial.h>

// GPS Simulation with coordinates
float simulatedLat = 18.6278;   // Initial Latitude
float simulatedLng = 73.8000;   // Initial Longitude

// MPU6050 Simulation
const int MPU = 0x68; // I2C address of MPU6050
int16_t accX, accY, accZ;
float ax, ay, az;

void setup() {
  Serial.begin(9600);  // Communicate with Arduino Uno
  Wire.begin();

  // Initialize MPU6050
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);  // Wake up MPU6050
  Wire.write(0);
  Wire.endTransmission(true);

  delay(1000);
  Serial.println("System ready!");
}

void loop() {
  // Read accelerometer data from MPU6050
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);  // Starting register for accelerometer data
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true);

  accX = (Wire.read() << 8 | Wire.read());
  accY = (Wire.read() << 8 | Wire.read());
  accZ = (Wire.read() << 8 | Wire.read());

  // Convert to G (gravity)
  ax = accX / 16384.0;
  ay = accY / 16384.0;
  az = accZ / 16384.0;
```

```arduino
  // Simulate a small constant movement to always trigger detection
  ax += 0.02; // Small artificial movement on X-axis
  ay += 0.02; // Small artificial movement on Y-axis
  az += 0.02; // Small artificial movement on Z-axis

  // Set a very low threshold for detecting slight movements (0.05 for extreme sensitivity)
  if (abs(ax) > 0.05 || abs(ay) > 0.05 || abs(az) > 0.05) {
    // Simulate changing latitude and longitude when movement is detected
    simulatedLat += 0.0001;  // Increment latitude slightly
    simulatedLng -= 0.0001;  // Decrement longitude slightly

    // Display accident detection and updated coordinates
    Serial.print("ACCIDENT DETECTED! ");
    Serial.print("LAT: ");
    Serial.print(simulatedLat, 4); // Display latitude with 4 decimal places
    Serial.print(", LNG: ");
    Serial.println(simulatedLng, 4); // Display longitude with 4 decimal places
  } else {
    // No significant movement detected (this shouldn't trigger with the added artificial movement)
    Serial.println("No significant movement detected");
  }

  delay(500);
  // Wait 500ms before checking again
}

#include <SoftwareSerial.h>

SoftwareSerial sim800l(7, 8); // RX, TX pins (7 = RX, 8 = TX)

void setup() {
  // Start serial communication with the computer and SIM800L
  Serial.begin(9600);
  sim800l.begin(9600);  // Baud rate for SIM800L module

  delay(1000);
  Serial.println("SIM800L Module Ready");

  sendSMS("Hello from SIM800L!", "+1234567890");  // Send SMS to a given phone number
}

void loop() {
  // Nothing to do here for now
```

```
}

void sendSMS(String message, String phoneNumber) {
  // Send the AT command to send SMS
  sim800l.println("AT+CMGF=1");   // Set SMS mode to text
  delay(100);

  sim800l.print("AT+CMGS=\"");     // Send the phone number
  sim800l.print(phoneNumber);
  sim800l.println("\"");
  delay(100);

  sim800l.println(message);        // Send the message text
  delay(100);

  sim800l.write(26);               // Send the CTRL+Z character to indicate end of SMS
  Serial.println("SMS Sent!");
}
```