## IMU-Based Vertical Motion Detection Algorithm

### Purpose:

Detect vertical motion events (e.g., takeoff or bounce) using 6-axis IMU data (ICM-20948). Focus on Z-axis acceleration forming a smooth sinusoidal-like cycle, while filtering out noise, tilt, or sudden disturbances.

### Algorithm Steps

#### 1. Idle (Reset) State Detection

* **Condition:**
  + All axes (AccX, AccY, AccZ, GyroX, GyroY, GyroZ) are approximately zero ± small noise margin.
* **Action:**
  + Confirm system is in rest/reset state.

#### 2. Motor Activation Detection

* **Condition:**
  + All axes show low-level ripple noise.
  + If any axis exceeds a large threshold → **Reset** to Idle.
* **Action:**
  + Confirm motors are spinning, but system is not lifting yet.

#### 3. Sine Wave Pattern (First Half Cycle) – Z Gradual Rise → Fall

##### 3.1 Gradual Z Rise Detection

* **Condition:**
  + AccZ increases **gradually** over a fixed time window (e.g., 10–15 samples).
  + No spikes on AccX/Y or any gyros.
* **Method to Detect:**
  + Use a **sliding window** and check if AccZ values are mostly increasing:
  + count = 0  
    for i in range(1, len(window)):  
     if window[i] > window[i-1] - margin:  
     count += 1  
    if count >= threshold:  
     gradual\_rise = True
  + Typical: margin = 0.05g, threshold = 7/10.

##### 3.2 Gradual Z Fall Detection

* **Condition:**
  + AccZ now **gradually decreases** from peak value (after previous rise).
  + Again, no spikes on other axes.
* **Method:**
  + Same as above, but check for decreasing trend.
* **If both rise + fall detected in sequence → Sine Wave Half-Cycle detected.**

#### 4. Sine Wave Pattern (Second Half Cycle) – Z Gradual Fall → Rise

##### 4.1 Gradual Z Fall Detection

* **Condition:**
  + Start from steady Z, detect a **smooth drop** in AccZ over time.
  + Other axes remain stable.
* **Action:**
  + Track sliding window for mostly decreasing values.

##### 4.2 Gradual Z Rise Detection

* **Condition:**
  + After minimum, AccZ **gradually increases** again toward zero.
  + Maintain ripple-only status on other axes.
* **Action:**
  + If both fall + rise patterns completed → Second half-cycle complete.

#### 5. Steady State Detection

* **Condition:**
  + AccZ returns to low-ripple region (± small noise).
  + All axes within ripple limits.
* **Action:**
  + Confirm motion is over and return to stable state.

### Gradual Trend Detection – Sliding Window Logic

Use a buffer of recent Z samples:

def is\_gradual\_trend(window, direction, margin=0.05, threshold=7):  
 count = 0  
 for i in range(1, len(window)):  
 if direction == 'rising':  
 if window[i] > window[i-1] - margin:  
 count += 1  
 elif direction == 'falling':  
 if window[i] < window[i-1] + margin:  
 count += 1  
 return count >= threshold

### Summary of Conditions:

| Step | AccZ Trend | Other Axes | Result |
| --- | --- | --- | --- |
| Step 1 | Near zero | Near zero | Idle |
| Step 2 | Ripples | Ripples | Motor active |
| Step 3 | Gradual rise → fall | Ripple only | Sine wave 1st half-cycle |
| Step 4 | Gradual fall → rise | Ripple only | Sine wave 2nd half-cycle |
| Step 5 | Ripple zone | Ripple zone | Stable / return to reset |

### State Diagram:

[Idle State]  
 ↓  
[Motor ON – ripples only]  
 ↓  
[Gradual Z Rise]  
 ↓  
[Gradual Z Fall]  
 ↓  
[Gradual Z Fall]  
 ↓  
[Gradual Z Rise]  
 ↓  
[Steady State]  
 ↑ (Any spike → Reset to Idle)