

# Lab 9 Milestone 2: Sensor Data Processing

## Part 1: Understanding Sensor Data Errors

### Objective

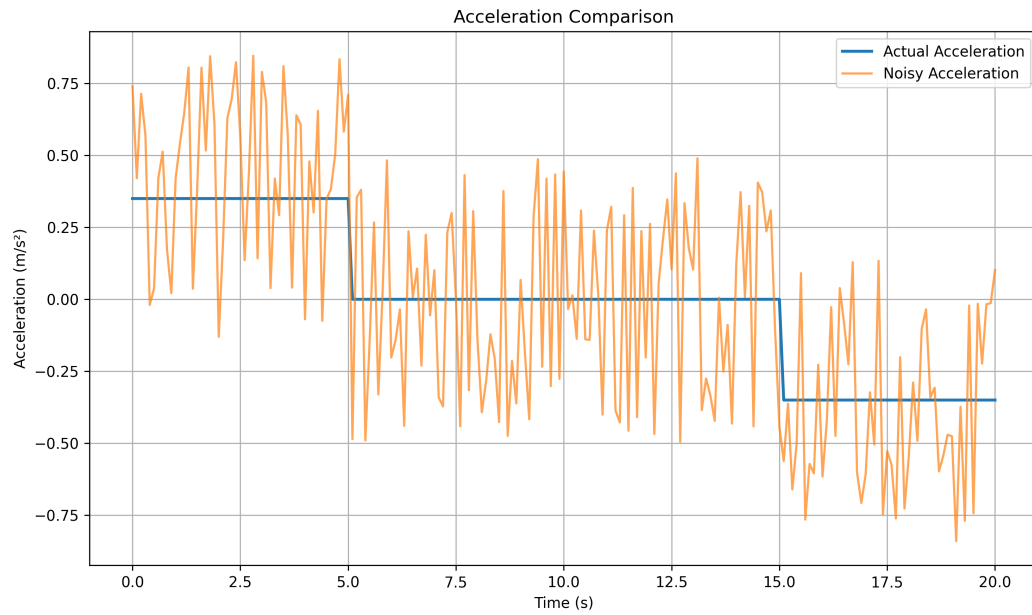
This analysis examines how sensor noise in accelerometer data affects distance estimation. Using synthetic acceleration data, we integrated acceleration values to compute velocity and distance, comparing results from clean data versus noisy sensor readings.

### Methodology

The ACCELERATION.csv dataset contains timestamped measurements with both actual and noisy acceleration values. The analysis used numerical integration with a timestep of 0.1 seconds. Velocity was computed as  $v(t) = v(t-1) + a(t-1) * dt$ , and distance as  $d(t) = d(t-1) + v(t-1) * dt$ , where  $dt$  is the time step between measurements.

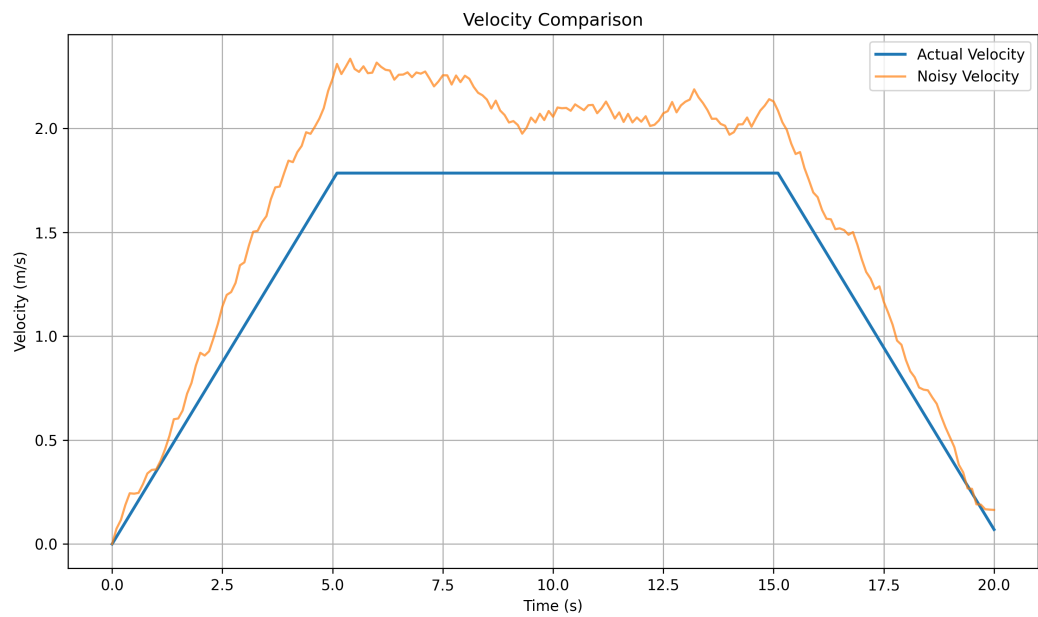
### Results

#### Figure 1: Acceleration Comparison



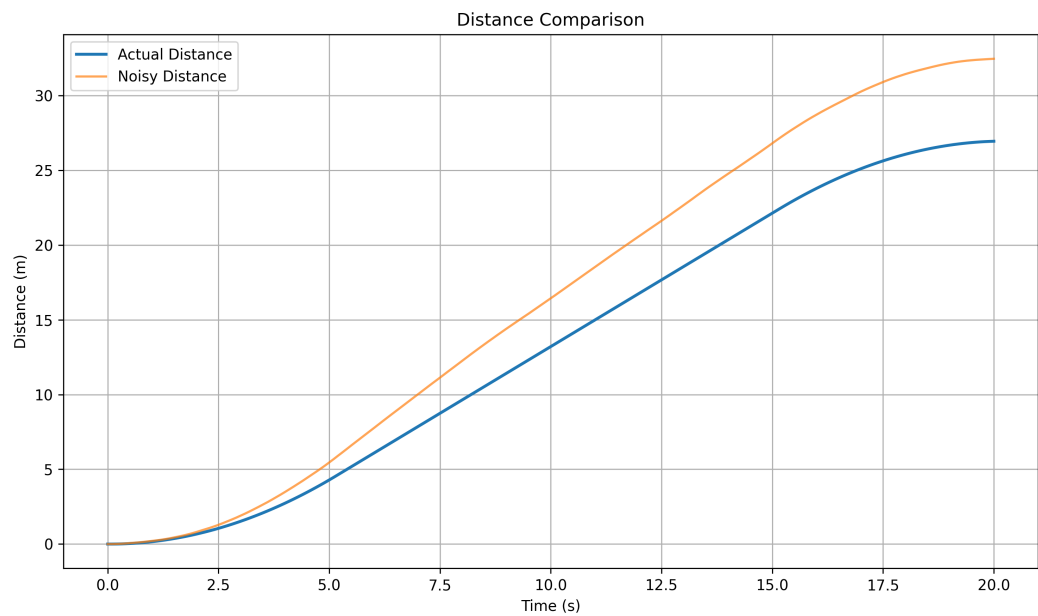
The acceleration plot shows the actual acceleration pattern (a step function with values 0.35, 0, and -0.35 m/s<sup>2</sup>) compared to noisy sensor measurements. The noise introduces random variations around the true values.

**Figure 2: Velocity Comparison**



Integrating acceleration produces velocity. The actual velocity follows a clear triangular pattern, while the noisy velocity shows cumulative error that grows over time due to noise integration.

**Figure 3: Distance Comparison**



Distance traveled shows the most significant error accumulation. The actual distance follows a smooth curve reaching the final value, while the noisy estimate diverges substantially due to double integration of noise.

## Quantitative Analysis

**Final distance using actual acceleration:** 26.9430 m

**Final distance using noisy acceleration:** 32.4594 m

**Difference between estimates:** 5.5164 m

**Percentage error:** 20.47%

## Discussion

The analysis demonstrates that sensor noise has a compounding effect when integration is performed. While the noise in acceleration appears relatively small, integrating once to get velocity and again to get distance causes errors to accumulate significantly. The 20.47% error in final distance estimation highlights the critical importance of sensor accuracy in pedestrian dead-reckoning applications.

This error accumulation explains why practical PDR systems require additional techniques such as zero-velocity updates, step detection algorithms, and sensor fusion to maintain accuracy over time. Without these corrections, raw integration of noisy accelerometer data quickly becomes unreliable for navigation.

## Conclusion

Noisy accelerometer measurements introduce significant errors in distance estimation through double integration. The 5.5 meter difference over a short movement demonstrates why real-world PDR systems must implement sophisticated filtering and error correction strategies to achieve useful positioning accuracy.

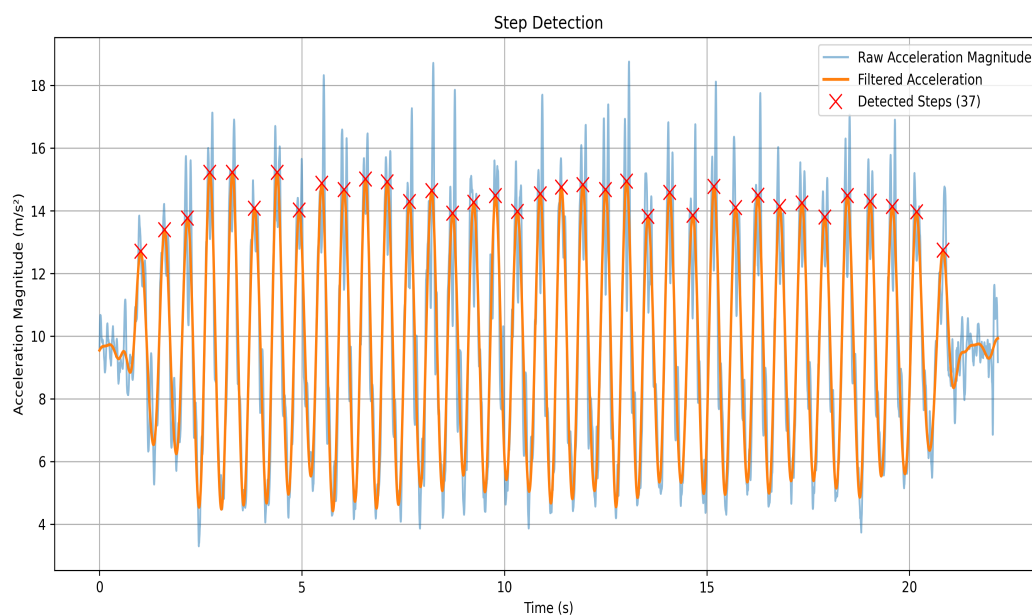
## Part 2: Step Detection

Step detection is crucial for pedestrian dead-reckoning systems. By detecting individual steps, the system can estimate distance traveled by multiplying step count by estimated step length.

## Methodology

The step detection algorithm processes accelerometer data from all three axes by calculating the acceleration magnitude. A low-pass Butterworth filter (cutoff frequency 3 Hz) smooths the signal to reduce noise. Peak detection identifies local maxima in the filtered signal that exceed a threshold based on mean and standard deviation, with a minimum time spacing between detections to prevent false positives from a single step.

**Figure 4: Step Detection Results**



The algorithm successfully detected 37 steps in the WALKING.csv dataset. Each peak in the filtered acceleration magnitude corresponds to one complete step cycle.

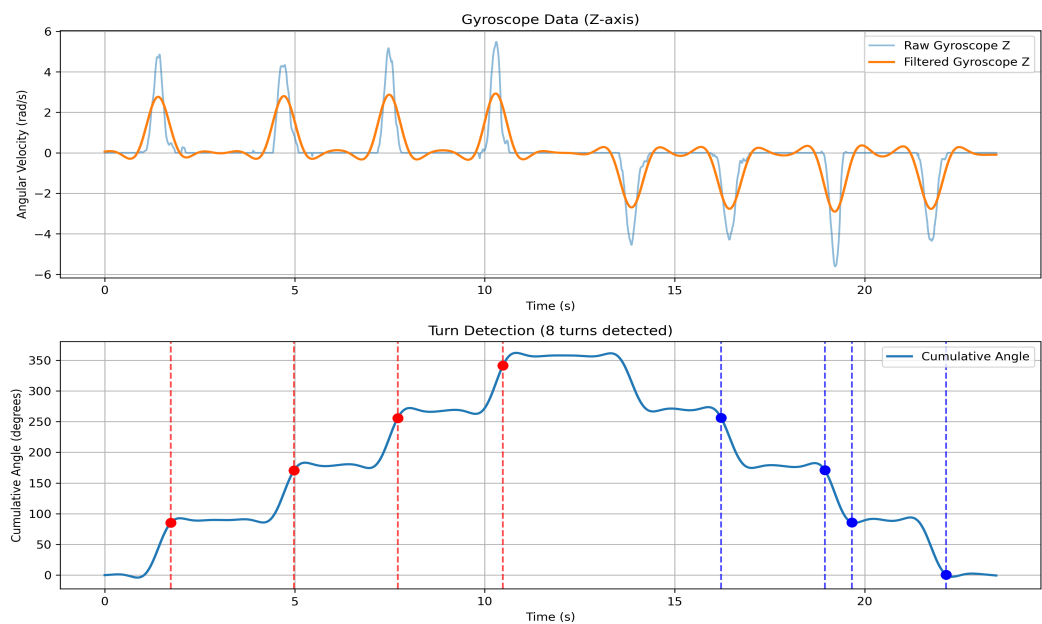
## Part 3: Turn Detection

Direction detection complements step detection by tracking changes in heading. This implementation uses gyroscope data to detect rotational movements and identify 90-degree turns.

## Methodology

The turn detection algorithm processes gyroscope Z-axis data (rotation around the vertical axis). After applying a low-pass filter (cutoff frequency 1 Hz), the angular velocity is integrated over time to compute cumulative rotation angle. When the angle crosses an 85-degree threshold (allowing for sensor noise), a turn is detected. The algorithm distinguishes between clockwise and counter-clockwise rotations based on the sign of the angle change.

Figure 5: Turn Detection Results



The algorithm detected 8 turns in TURNING.csv: 4 clockwise turns (completing one full 360-degree rotation) and 4 counter-clockwise turns (completing another full rotation in the opposite direction). Each detected turn corresponds to approximately 90 degrees of rotation.

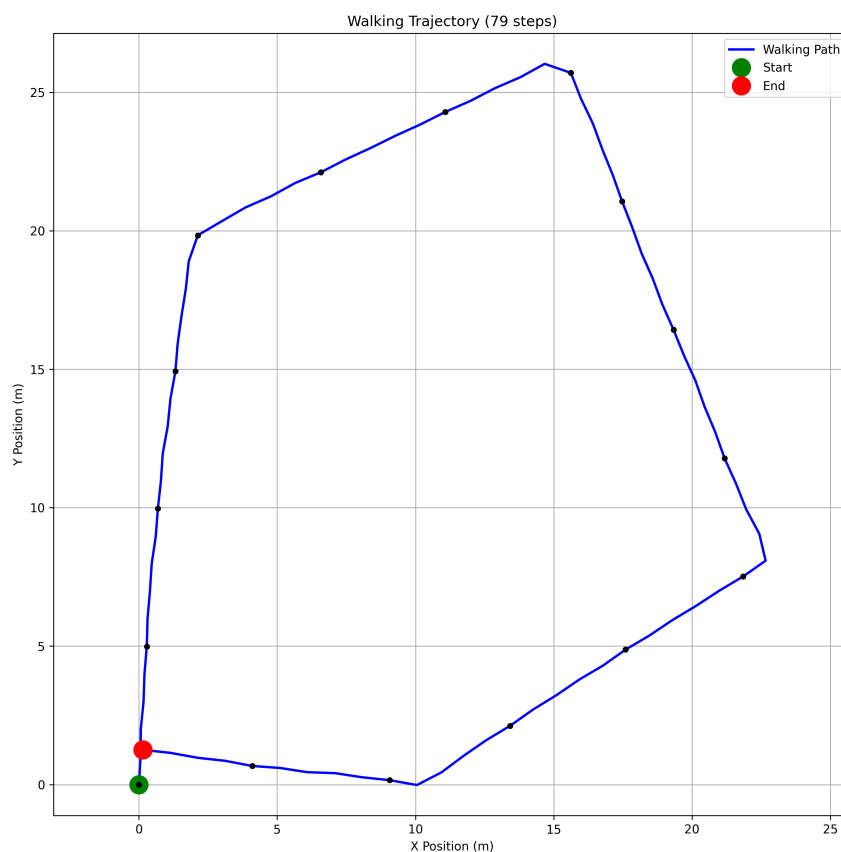
## Part 4: Trajectory Reconstruction

Combining step detection and turn detection enables reconstruction of the complete walking trajectory. This demonstrates how PDR systems can track position over time without GPS.

### Methodology

The trajectory reconstruction algorithm applies both step detection and turn detection to the WALKING\_AND\_TURNING.csv dataset. For each detected step, the algorithm advances the position by 1 meter in the current heading direction. The heading is continuously updated based on the integrated gyroscope data. Starting from an origin at (0,0) facing north, the algorithm tracks position changes throughout the walking session.

**Figure 6: Reconstructed Trajectory**



The reconstructed trajectory shows 79 detected steps forming a complete walking path. The green marker indicates the starting position, while the red marker shows the ending position. The close proximity of start and end points suggests the walking path was designed to return near the origin.

## **Overall Conclusion**

This lab demonstrated the fundamental components of pedestrian dead-reckoning: understanding sensor error accumulation, detecting steps, identifying turns, and reconstructing trajectories. These techniques form the basis of indoor navigation systems and fitness tracking applications. The analysis showed that while raw sensor data contains significant noise, proper filtering and peak detection algorithms can reliably extract meaningful motion information. Future improvements could include adaptive step length estimation, zero-velocity updates to reduce drift, and sensor fusion with magnetometer data for improved heading accuracy.