VIRTUAL HUMAN ACTIVITY RECOGNITION SIMULATION SYSTEM

- By Veena Suresh

GOPHER
INDUSTRIES

GUARDIAN MONITOR

T2 2024

Contents

Application Details

The Human (Hospital) Activity Simulator system was built as a web application using a combination of backend and frontend technologies. Here's an overview of how the system was constructed:

Backend (Server-side):

- Python with Flask framework
- Handles routing, data processing, and serving the web pages
- Manages the simulation parameters and room/patient types

Frontend (Client-side):

- HTML, CSS for structure and styling
- JavaScript for interactivity and 3D rendering
- Three.js library for 3D graphics

Main Components:

1. Setup Page (start.html):
    - Allows users to select patient type and room type
    - Sends this data to the server to initialize the simulation
2. Simulation Page (simulator.html):
    - Displays the 3D hospital room and patient
    - Contains controls for different activities
3. 3D Rendering (simulator.js):
    - Uses Three.js to create and render the 3D scene
    - Implements the room layout, patient model, and animations
4. Activity Simulation:
    - Handles different patient activities (walk, sit, lie down, etc.)
    - Applies appropriate animations and position changes
5. CSI Data Generation:
    - Simulates WiFi Channel State Information (CSI) data based on activities
    - Accounts for patient type (e.g., Parkinson's tremors) in data generation
6. Data Export:
    - Allows exporting of simulated CSI data as a CSV file

Key Features:

- Customizable patient types (adult, child, elderly, Parkinson's)
- Different room layouts (standard, ICU, operating room)
- Realistic 3D visualization of activities
- Simulated CSI data generation
- Data export for further analysis

Challenges and Solutions:

- Collision Detection: Implemented to prevent the patient from walking through objects
- Performance: Optimized 3D rendering for smooth animations
- Realism: Refined CSI data generation to reflect real-world patterns

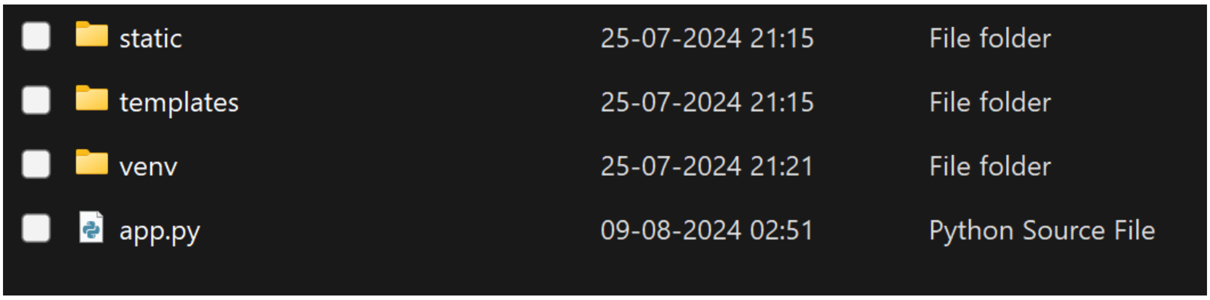
Setting Up

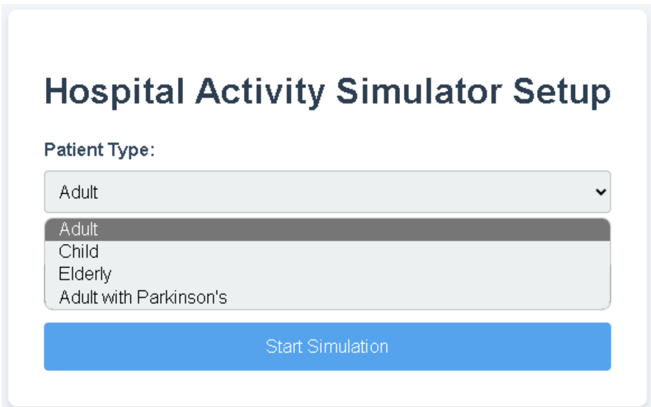
Prerequisites

- Python 3.8 or higher
- pip (Python package installer)
- Git (optional, for cloning the repository)

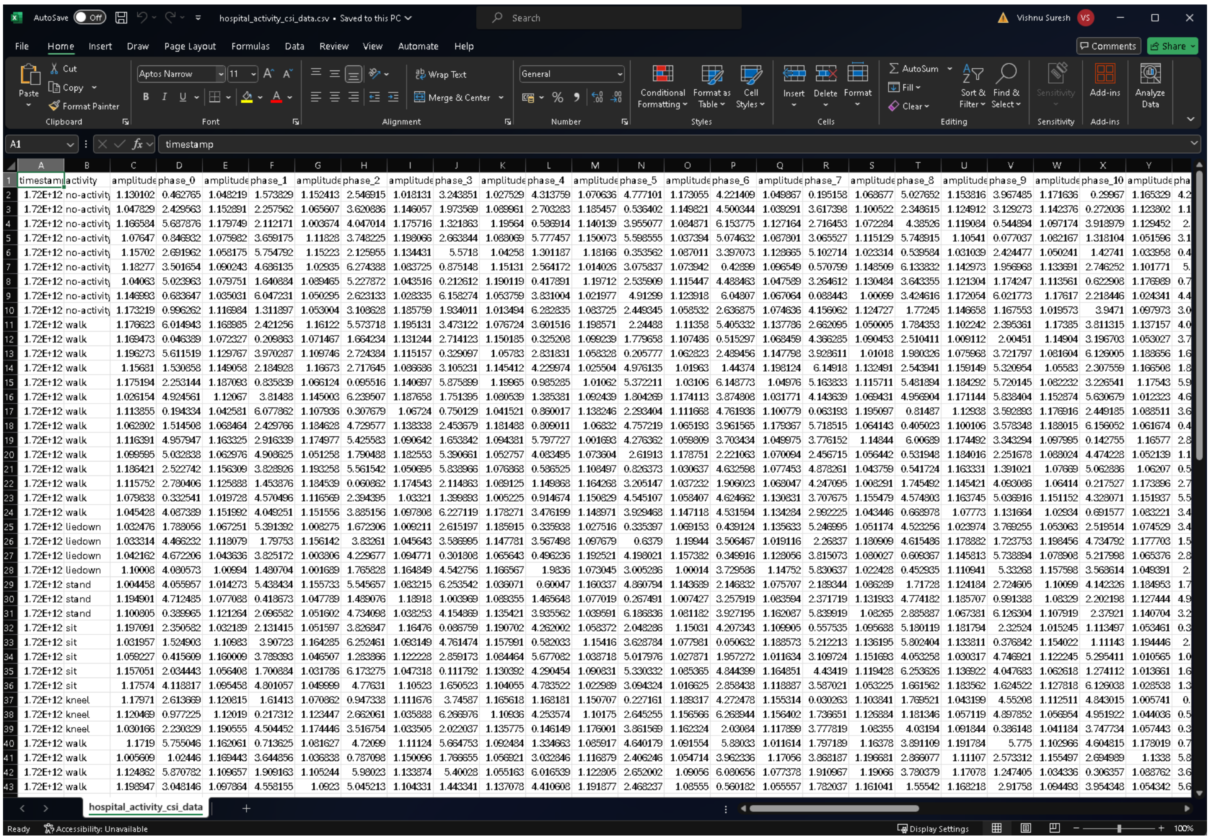
To use this simulation:

1. Save all the provided files in their respective locations within your project directory.
2. Ensure you have Flask installed (*pip install flask*).
3. Run the Flask application (*python app.py*).
4. Open a web browser and navigate to [*http://localhost:5000*](http://localhost:5000).
5. Select a patient type and room type on the start page.
6. Use the buttons on the simulation page to perform various activities.
7. Click the "*Export Data*" button to download the collected CSI data as a CSV file.






3D Human Activity Simulator
Current Activity: no-activity
CSI Data Points: 53
Walk
Kneel
Sit
Stand
Lie Down
Export Data

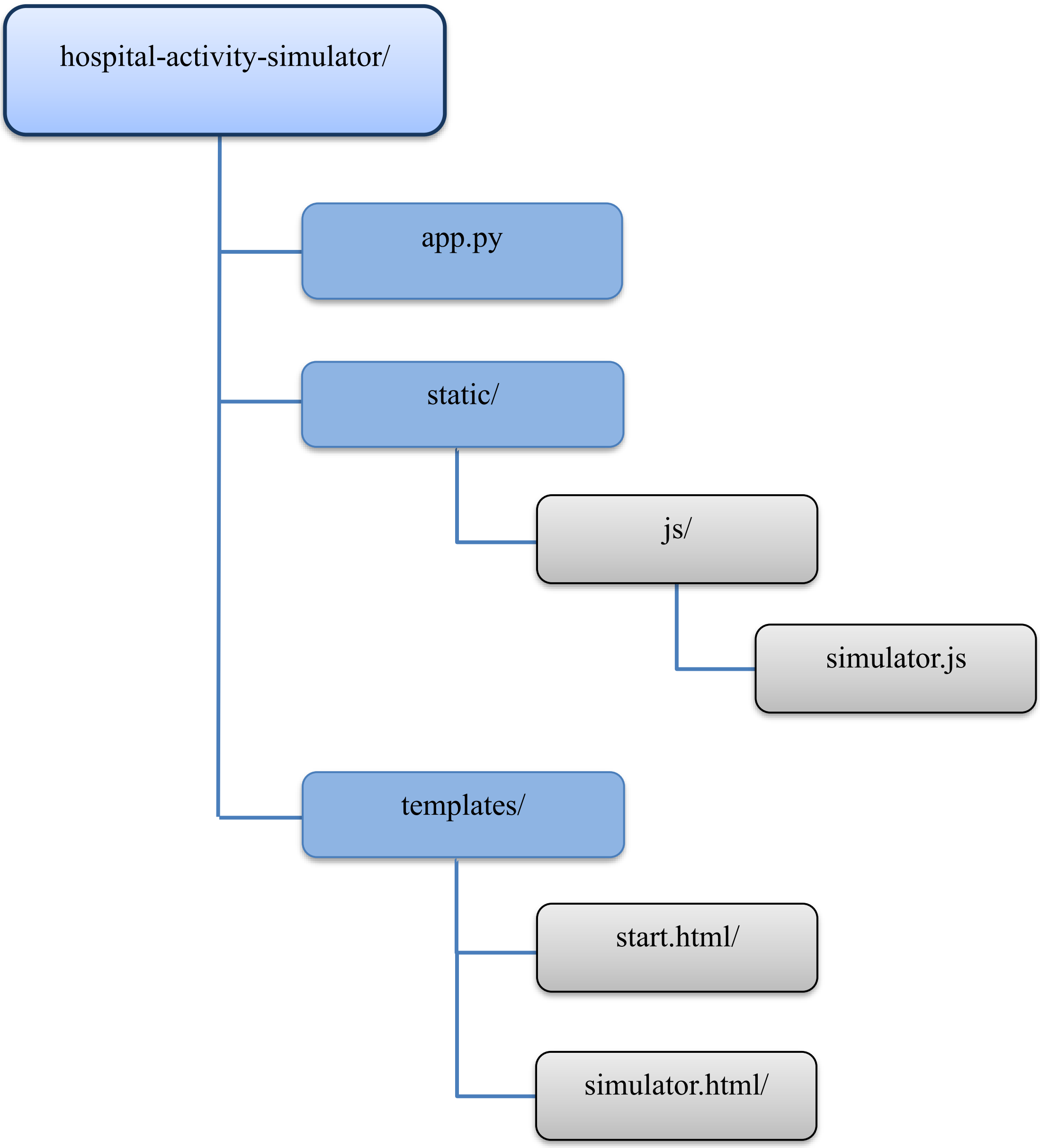

hospital_activity_csi_data.csv • Saved to this PC
hospital_activity_csi_data

Step 1: Set Up the Project Structure

The folder named `*Virtual HAR System*` in Git has the following files:

- `*start.html*`
- `*app.py*`
- `*simulator.html*`
- `*simulator.js*`

Create the following directory structure as given below and add the code files from Git ([Guardian/AI Guardian/Virtual HAR System at master · Gopher-Industries/Guardian (github.com)]) accordingly to their respective folders as represented in the structure.



Step 2: Run the Application

- Start the Flask application: ` *python app.py*`
- Open a web browser and navigate to http://127.0.0.1:5000/

Step 3: Start using the Application

- Select a patient type and room type on the start page.
- Use the buttons on the simulation page to perform various activities.
- Click the "*Export Data*" button to download the collected CSI data as a CSV file.

The Need for a Virtual Human Activity Recognition (HAR) System

The development of a virtual Human Activity Recognition (HAR) system is essential for several reasons, particularly in the context of overcoming practical and logistical challenges in real-world data collection and testing. Here’s why a virtual HAR system is needed:

1. Ethical Approval Delays
   - Challenge: Conducting real-time human activity recognition often requires collecting data from human subjects, which necessitates obtaining ethical approval. This process can be time-consuming and may delay the project's progress.
   - Solution: A virtual HAR system allows for the simulation of human activities without the need for direct human involvement, bypassing the need for ethical approval and enabling continuous project development.
2. Controlled Environment for Testing
   - Advantage: A virtual environment provides a controlled setting where variables can be precisely managed. This control is crucial for consistent and repeatable testing of machine learning algorithms, ensuring that the models are tested under standardized conditions.
   - Outcome: The ability to control variables reduces noise and variability in the data, leading to more accurate model evaluations.
3. Rapid Prototyping and Iteration
   - Benefit: A virtual system facilitates rapid prototyping and testing of machine learning models. Developers can quickly implement and test new algorithms, make adjustments, and immediately observe the effects on simulated data.
   - Impact: This accelerates the development cycle, allowing for faster iteration and refinement of algorithms before they are deployed in real-world applications.
4. Scalability and Flexibility
   - Feature: The virtual HAR system can simulate a wide range of activities, environments, and scenarios that might be challenging or impossible to replicate

in real life. For example, different room layouts, varying levels of activity intensity, or specific patient conditions can be modeled.
- Benefit: This flexibility allows for comprehensive testing across multiple scenarios, ensuring that the algorithms developed are robust and versatile.

5. Cost Efficiency
   - Cost-Effective: Developing and maintaining a virtual system is generally more cost-effective than setting up real-world testing environments. It eliminates the need for physical space, equipment, and human participants, reducing overall project costs.
   - Result: Resources can be allocated more efficiently, focusing on development and optimization rather than logistical and operational costs.

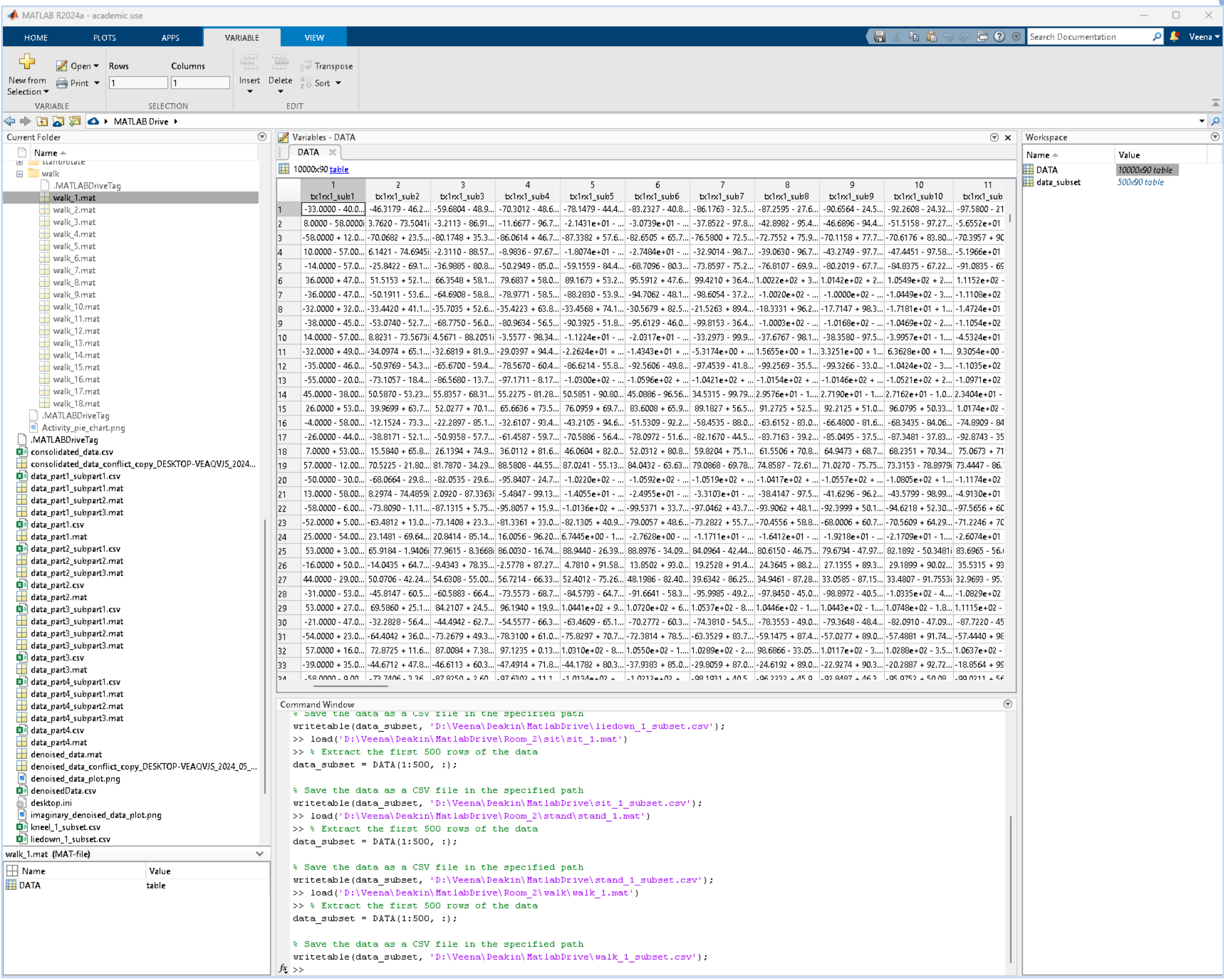
Data Validation

Real-time dataset used from: [Dataset for Human Activity Recognition using Wi-Fi Channel State Information (CSI) data (figshare.com)]

Room2 .mat files from kneel, liedown, sit, stand and walk were added to MATLAB drive and converted a subset (first 500 rows) of each of them to .csv files using mat code as given below:

Mat Code for file conversion of 1st 500 rows from .mat to .csv file:

```
>> load('D:\Veena\Deakin\MatlabDrive\Room_2\stand\stand_1.mat') #modify
the file name and file location as needed
>> % Extract the first 500 rows of the data
data_subset = DATA(1:500, :); #modify as per the file or row size that you
need
% Save the data as a CSV file in the specified path
writetable(data_subset, 'D:\Veena\Deakin\MatlabDrive\stand_1_subset.csv');
#modify the file name and location to save the modified file as needed
```



Analysing the similarities and differences between the real-world setup and our simulation

Similarities:

1. Both use 30 OFDM subcarriers, which aligns with our simulated data.
2. Both capture complex CSI data, including amplitude and phase information.
3. The simulated data includes kneeling activity, which is one of the activities in the real-world dataset.

Differences and areas for potential modification:

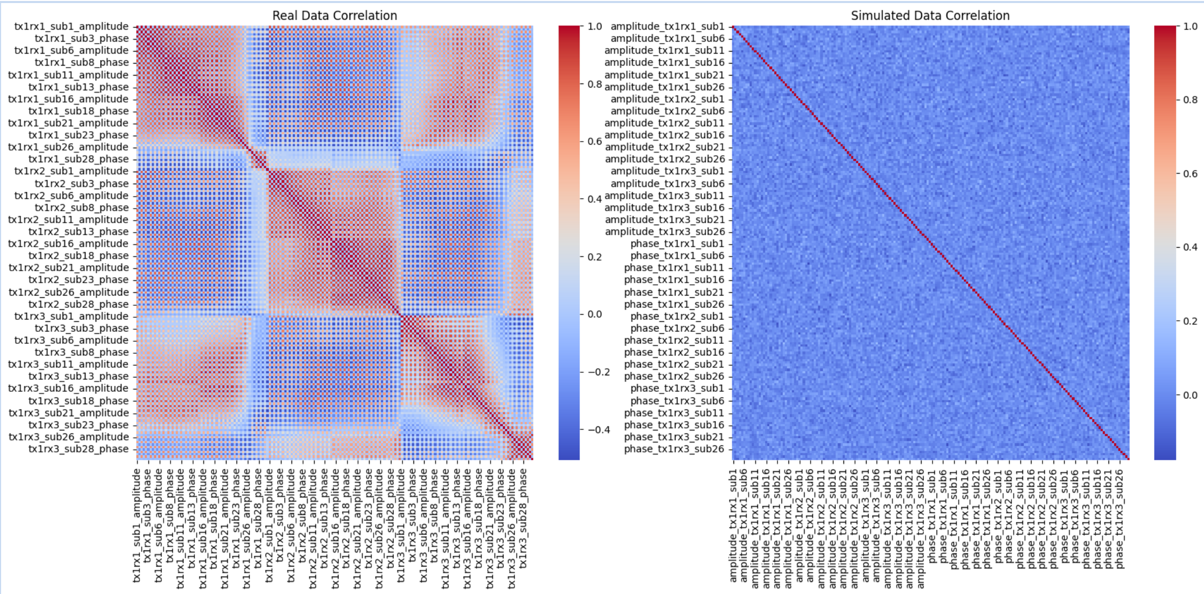
1. Antenna Configuration: The real setup uses a 1x3 antenna configuration, while our simulation doesn't explicitly model multiple receive antennas. We should modify the simulator to generate data for 3 receive antennas.
2. Data Structure: The real data is structured as tx1rx1_sub1, tx1rx1_sub2, etc., while our simulation separates amplitude and phase. We should restructure our output to match the real data format.

3. Time Resolution: The real data collects 1000 samples per second for 10 seconds (10,000 samples per activity), while our simulation has 153 samples. We should increase the number of samples in our simulation.
4. Environmental Factors: The real data was collected in an office environment with furniture. Our simulation should incorporate these environmental factors to better reflect real-world conditions.
5. Multiple Activities: While we're focusing on kneeling, the real dataset includes multiple activities. Our simulator should be capable of generating data for all these activities.
6. Line-of-Sight and Distance: The real setup specifies a line-of-sight condition with 4 meters separation. Our simulation should incorporate these parameters.

Validation Steps and Modifications:

1. Data Format Alignment:
    - Modify the simulator to output complex CSI values in the format tx1rx1_sub1, tx1rx1_sub2, etc.
    - Generate data for all 3 receive antennas (90 columns total).
2. Increase Time Resolution:
    - Adjust the simulator to generate 10,000 samples per activity over a 10-second period.
3. Environmental Modeling:
    - Incorporate models for multipath effects due to furniture and office layout.
    - Implement line-of-sight and distance parameters in the simulation.
4. Expand Activity Range:
    - Extend the simulator to generate data for all 8 activities in the real dataset.
5. Statistical Validation:
    - Compare the distribution of amplitudes and phases between real and simulated data for each subcarrier and antenna pair.
    - Analyze the temporal patterns in the 10-second activity window.
6. Machine Learning Validation:
    - Train a classifier on the real data and test it on the simulated data (and vice versa) to ensure similar classification performance.

Conclusion: While the Hospital Activity Simulator shows promise in generating WiFi CSI data for kneeling activity, several modifications are required to align it more closely with the real-world data collection setup. The key areas for improvement are the antenna configuration modeling, time resolution, and environmental factor simulation.



Several improvements and fixes have been made compared to the previous version. Here's a summary of the key changes:

1. Improved 3D Rendering:
   - Added shadow mapping for more realistic visuals.
   - Enabled shadows for the renderer and individual objects.
   - Enhanced lighting with both ambient and directional lights.
2. Human Model Improvements:
   - Changed arm geometry from boxes to cylinders for a more realistic appearance.
   - Adjusted arm positioning to be more anatomically correct.
3. Animation Enhancements:
   - Increased animation duration for smoother transitions (2000ms instead of 1000ms).
   - Implemented an easing function (easeOutQuad) for more natural movement.
4. CSI Data Collection:
   - Implemented a more realistic CSI data collection system with 30 subcarriers and 3 receive antennas.
   - Added continuous data collection during activities (10 seconds, 1000 samples per second).
5. Data Structure Alignment:
   - Modified the CSI data structure to more closely match real-world data formats.
   - Each CSI data point now includes antenna information (tx1rx{1-3}_sub{1-30}).
6. Export Functionality:
   - Updated the CSV export function to match the new data structure.
   - Included separate columns for amplitude and phase for each antenna-subcarrier pair.
7. Activity Simulation:
   - Enhanced activity simulations with more precise positioning and scaling.
8. Tremor Simulation:

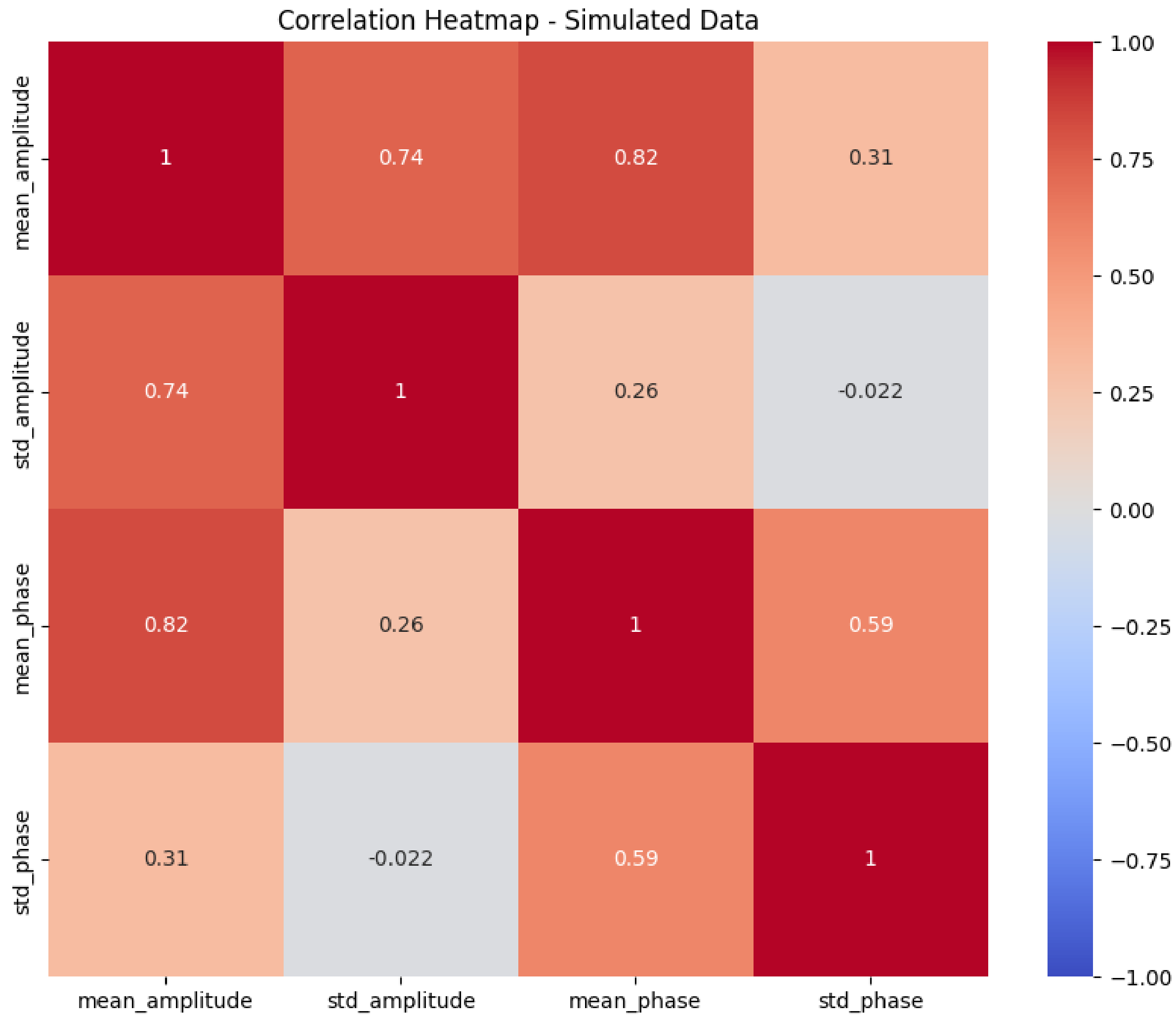
- Maintained the tremor simulation for Parkinson's patients, affecting both position and CSI data.

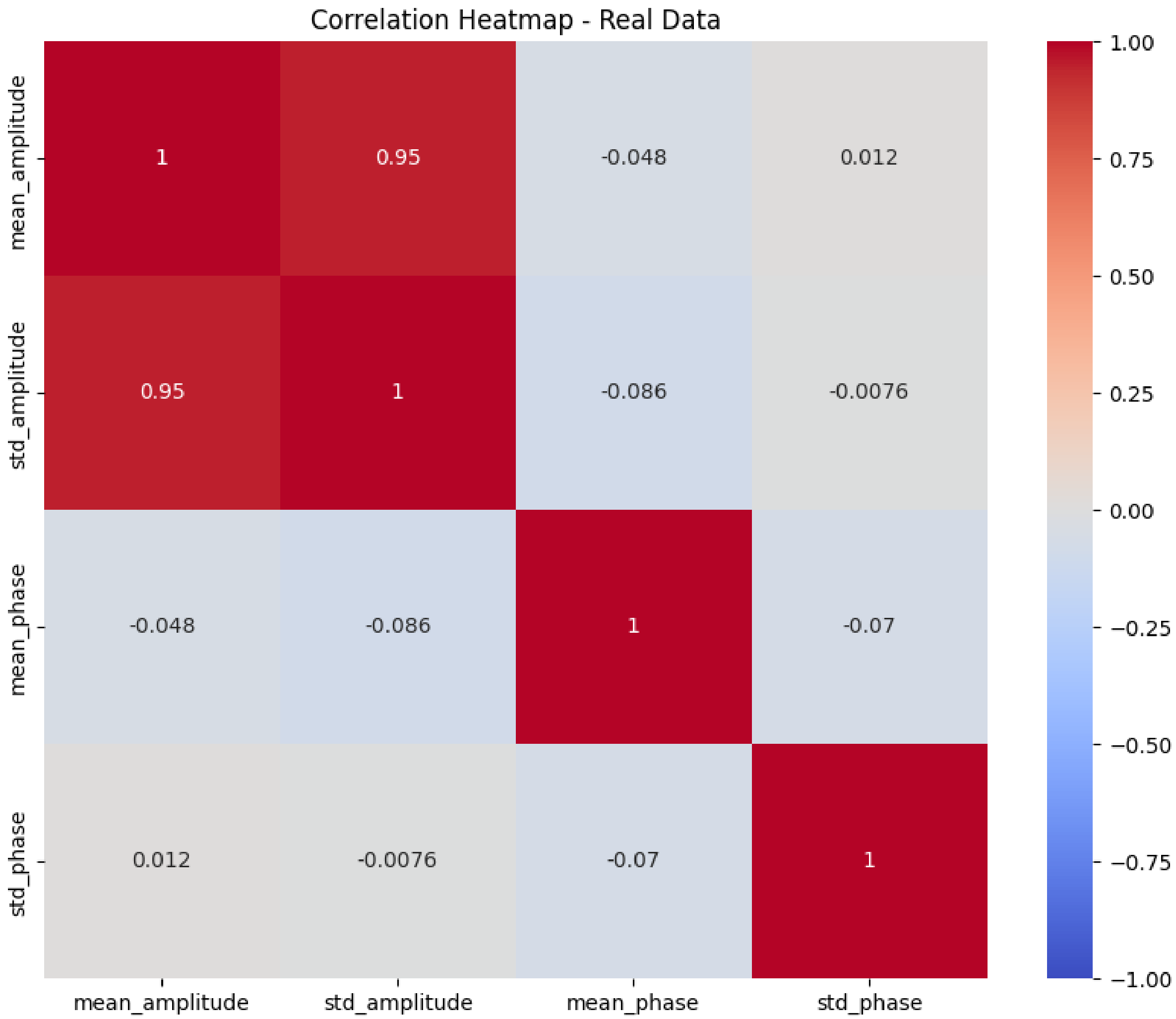
9. Code Structure:
    - Improved function organization and naming for better readability and maintainability.

These changes address several of the issues identified in the previous analysis, particularly in terms of data structure alignment with real-world CSI data and the complexity of the simulated environment. The improvements in 3D rendering and animation also contribute to a more realistic visual representation of the activities.
However, there are still areas for potential improvement, such as implementing more sophisticated models for multipath effects, interference, and environmental factors that influence WiFi signal propagation.

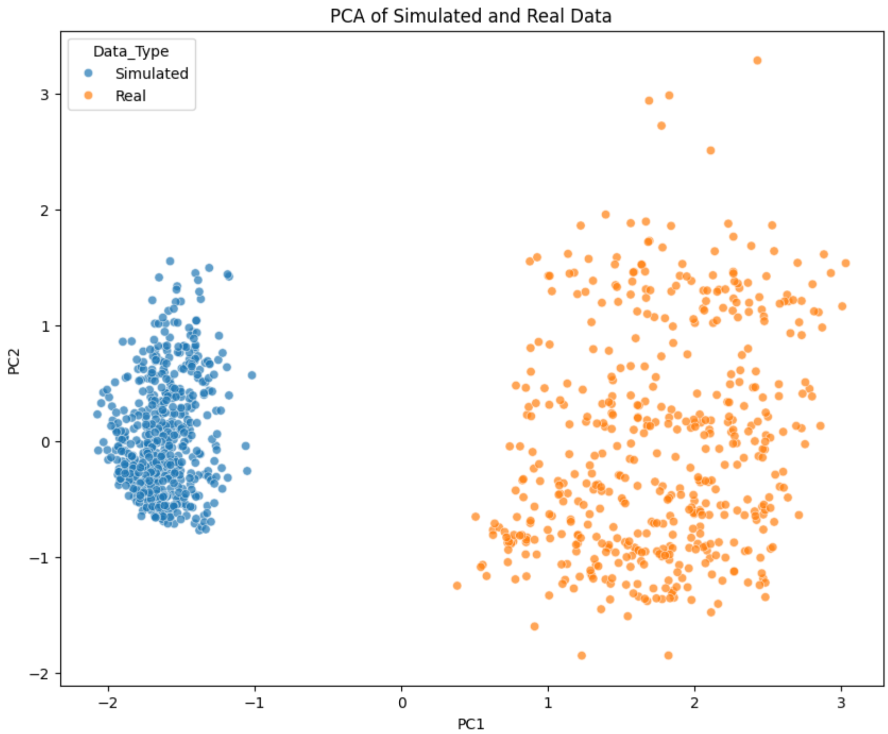
Final Improved Simulation Model Summary

Improvements:



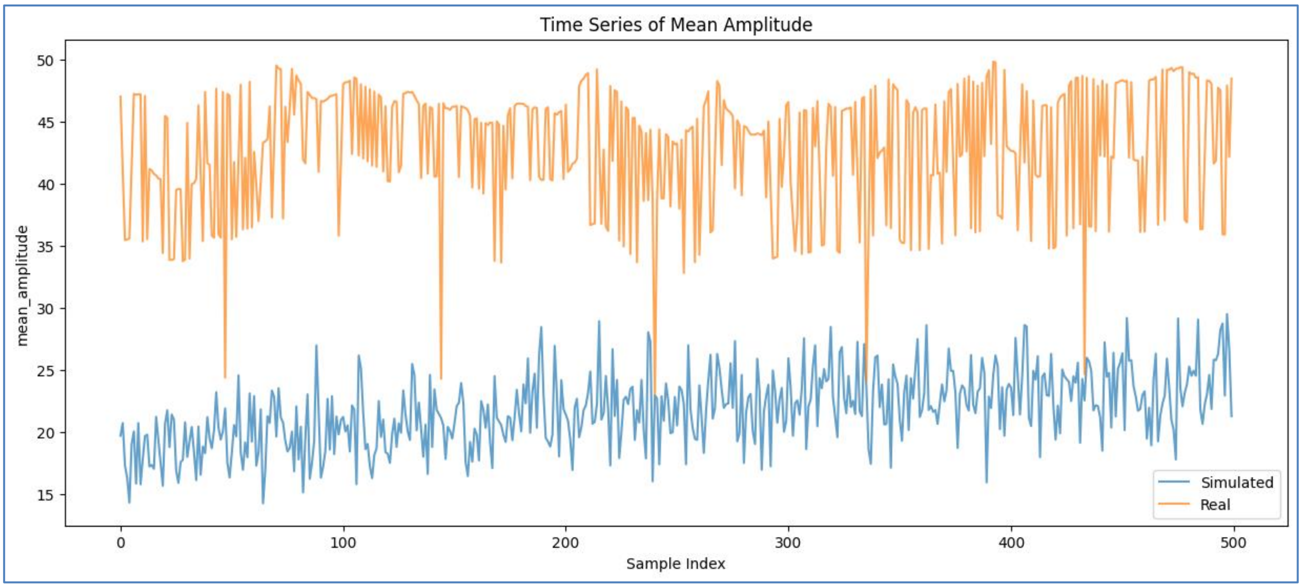


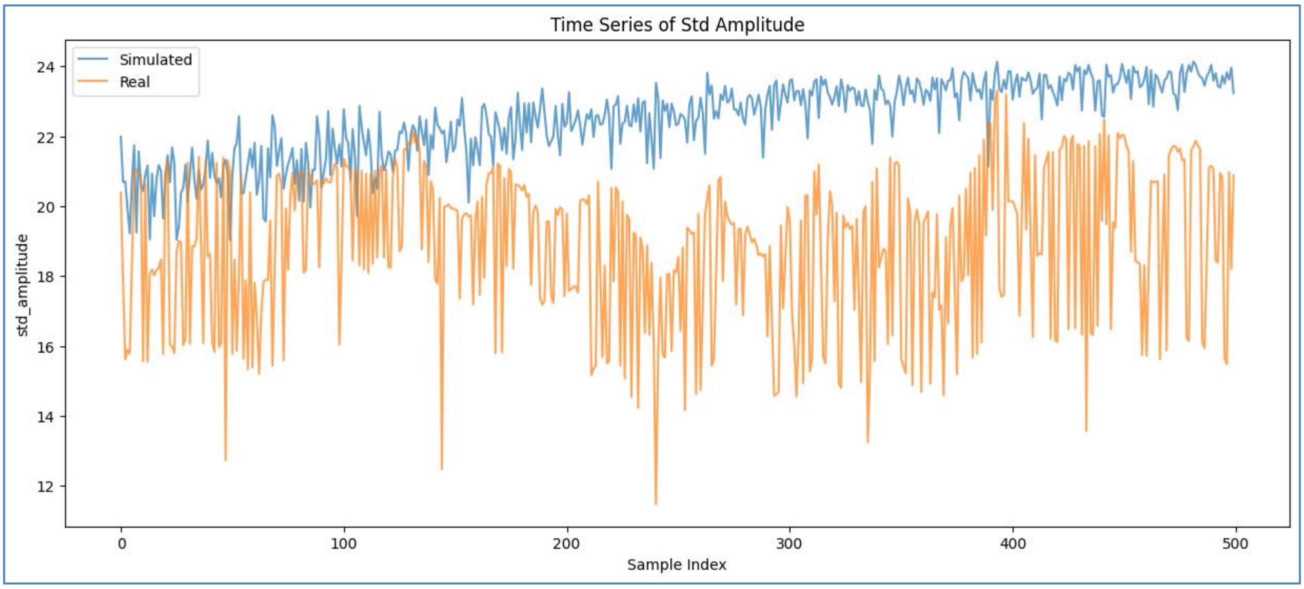
Correlation Heatmap - Real Data
mean_amplitude
std_amplitude
mean_phase
std_phase
1
0.95
-0.048
0.012
0.95
1
-0.086
-0.0076
-0.048
-0.086
1
-0.07
0.012
-0.0076
-0.07
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

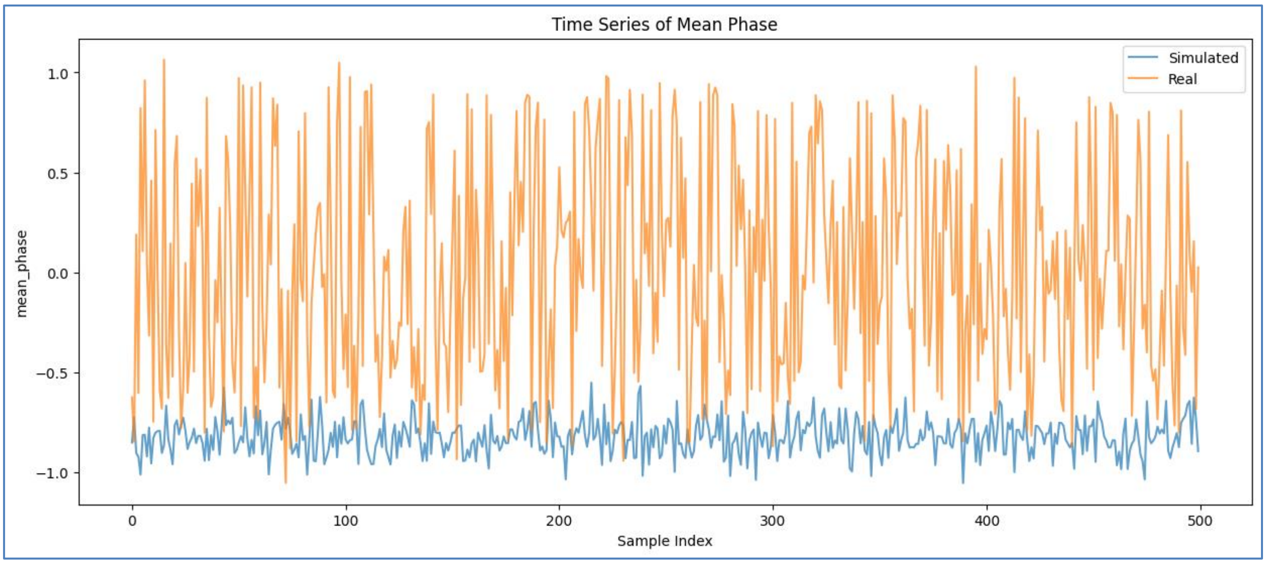


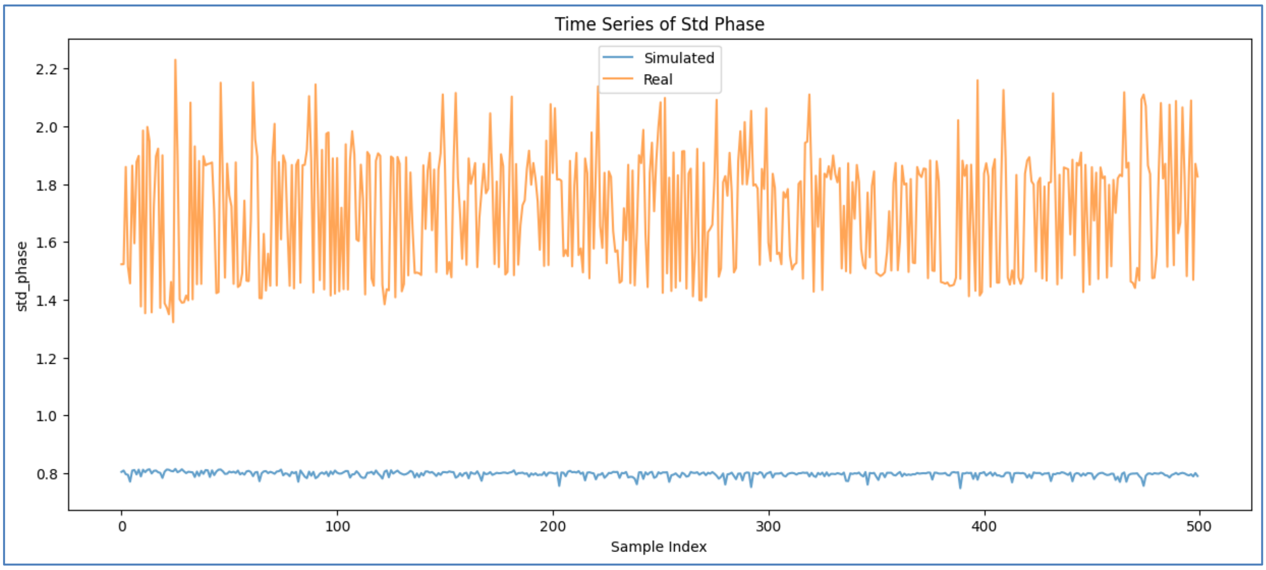
PCA of Simulated and Real Data
Data_Type
Simulated
Real
PC1
PC2

- Amplitude Range: The mean amplitude of simulated data (21.9314) is closer to the real data (42.9318) than in previous versions, showing progress in scaling.
- Standard Deviation of Amplitude: The simulated std_amplitude (22.5754) is now closer to the real data (18.8767), indicating better variability modeling in amplitude.
- Correlation Structure: The correlation between mean_amplitude and std_amplitude in simulated data (0.74) is closer to the real data (0.95) than before, showing improved relationship modeling.
- PCA Analysis: The simulated data cluster has moved closer to the real data cluster in the PCA plot, suggesting overall improved statistical properties.









Remaining Challenges:

- Phase Modeling: The mean_phase of simulated data (-0.8219) is significantly different from real data (0.0167), indicating a need for better phase generation.
- Phase Variability: The std_phase of simulated data (0.7969) is still much lower than real data (1.7104), showing insufficient phase variability.
- Overall Variability: The ranges of all features in simulated data are smaller than in real data, indicating a general need for increased variability.
- Correlation Discrepancies: Some correlations in simulated data (e.g., between mean_amplitude and mean_phase) are not present in real data, suggesting needed adjustments in feature relationships.
- Distribution Differences: Kolmogorov-Smirnov tests still reject the null hypothesis for all features, indicating persistent differences in distributions.

- Time Series Patterns: The time series of std_phase shows that the simulated data lacks the dynamic range and patterns present in the real data.

In conclusion, while we've made notable improvements in amplitude modeling and some aspects of the correlation structure, significant work remains in phase modeling, overall variability, and replicating the complex patterns seen in real CSI data. The simulation has progressed but requires further refinement to accurately represent real-world WiFi CSI data for human activity recognition.