Squat Jump Analysis:
Automating Analysis To Improve Patient Health

Andrew Ba, Walker Azam, Seoyeon Hong, John Cheng

Abstract

- The purpose of this project was to configure a web application that allows users to analyze force data derived from squat jumps. These squat jumps were done by subjects recovering from ACL-reconstruction surgery.
- Users are able to interact with the analysis program through the dashboard in Streamlit.
- Jump data in the form of a CSV can be dragged and dropped in the dash.
- Metrics, such as peak force, jump height, and timestamps are calculated in the back-end and returned to the user via multiple visualizations.
- The calculation results can be downloaded as a CSV for further analysis/comparision.

A Squat Jump

Motivation

- The key motivation was to make an analysis program that is constructive and accessible for users who can diagnose ACL injuries or work with people recovering from ACL reconstruction.
- These users most likely do not have enough knowledge or time to compute the jump metrics themselves, or are not able to create an automated program from scratch.
- Another motivation was to create a program that could accurately identify timestamps of jump events in order to automate calculations without user input.

Metric Calculations

Data preprocessing

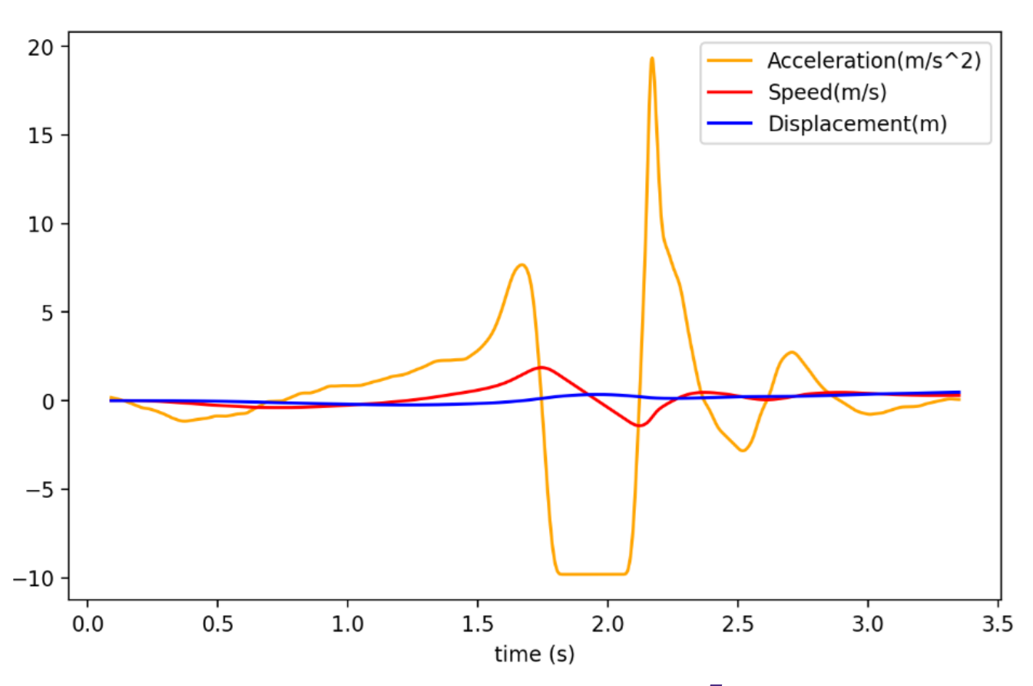
- Filter raw data to reduce noise using a low-pass butter filter.
- Integrate values in each frame to get the acceleration, velocity and position lists.
- Apply algorithms to extract the timestamps of each phase in one jump, and generated an index table.

Metric Computations

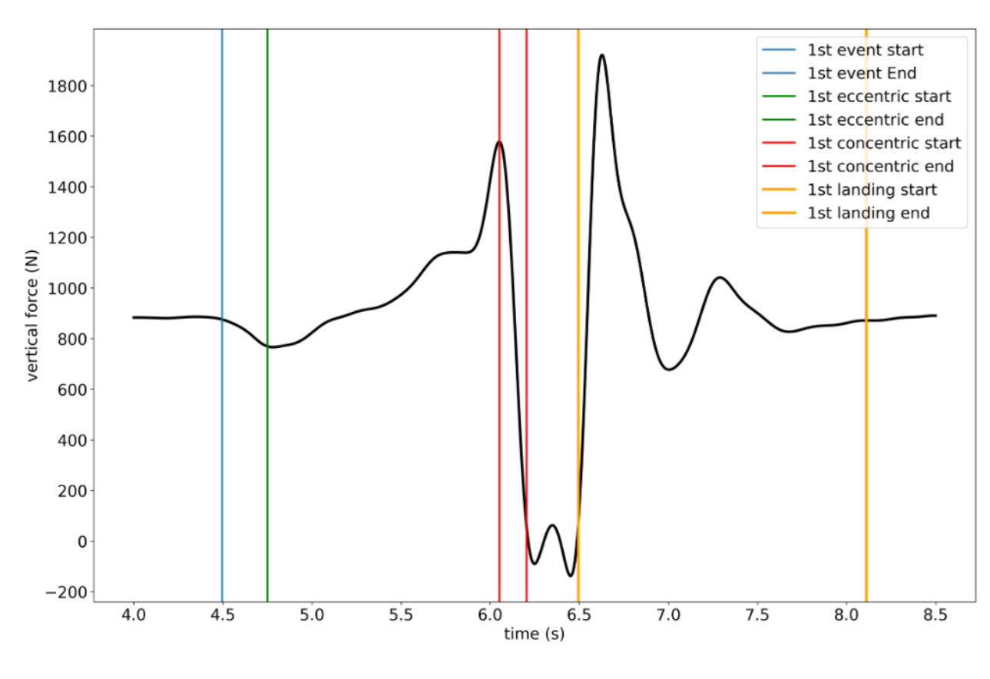
- Implement numerical functions such as linear regression and approximation to calculate metrics (peak power, jump height, center of pressure displacement, etc.) .
- Create a class object to store values and to make sure every procedures are visited during initialization.

Data Delivery

- Three data-frames, processed data, calculated metrics, and index table are delivered to the front-end.



Data Integrations



Timestamps

Technical Use Case – Center of Pressure Plot

Objective

- To visualize how the subject is loading during the squat jump through each foot.

Design Considerations

- The interactive plot using the Plotly Slider allows the viewer to play, pause, and reiterate back to different timings, making it easier to interpret and communicate to the patients of their jump performance.
- The viewer can select which jump to visualize.
- Position of two feet are plotted separately and placed side by side for comparison.

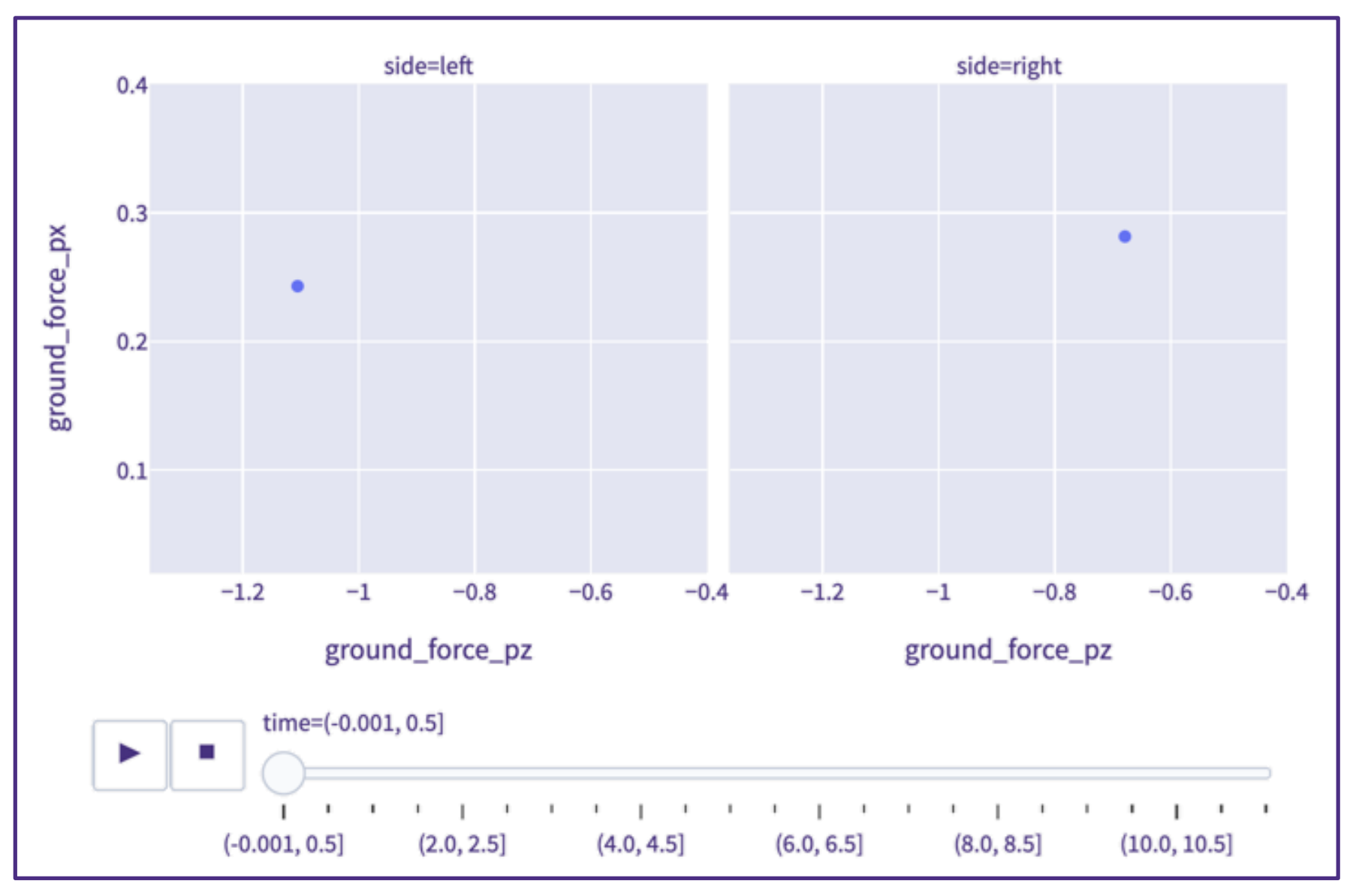
Technical Use Case – 3D Animation For Force

Objective

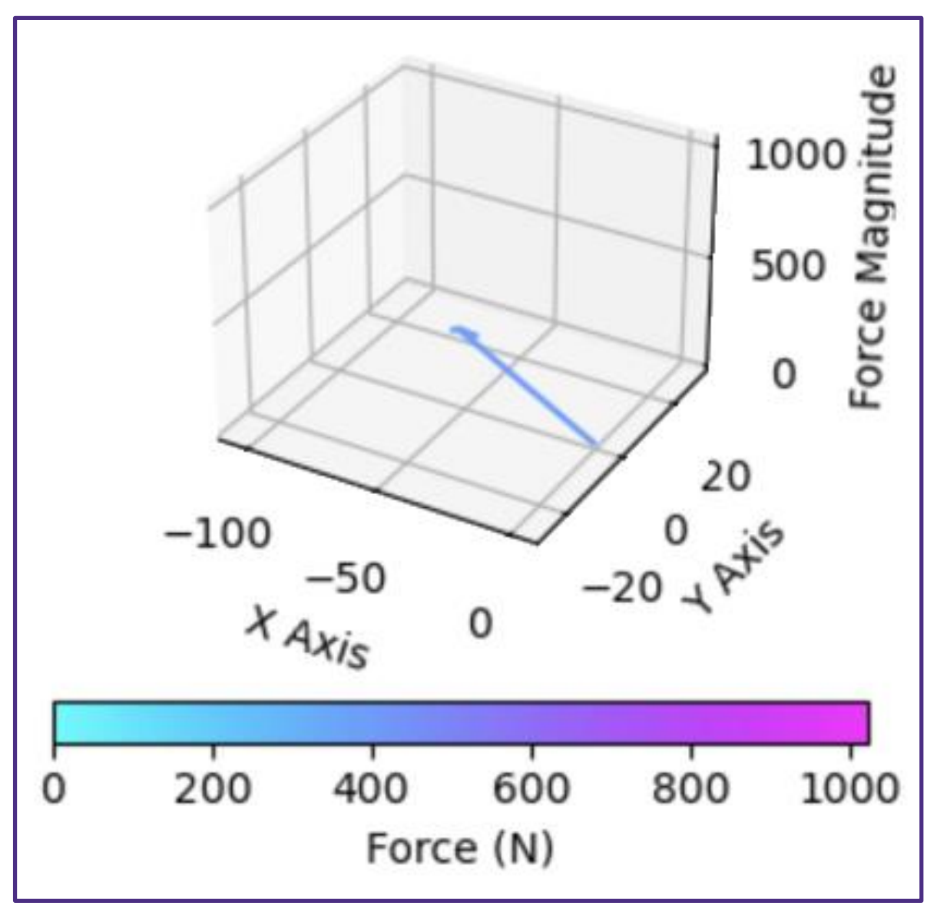
- To compare magnitude and angle of the left and right leg for each jump.

Design Considerations

- The color of the arrow changes based on force magnitude for easier interpretation.
- Plots are visualized using Matplotlib FuncAnimation and is saved as a JSHTML file to display on Streamlit



Center of Pressure Plot

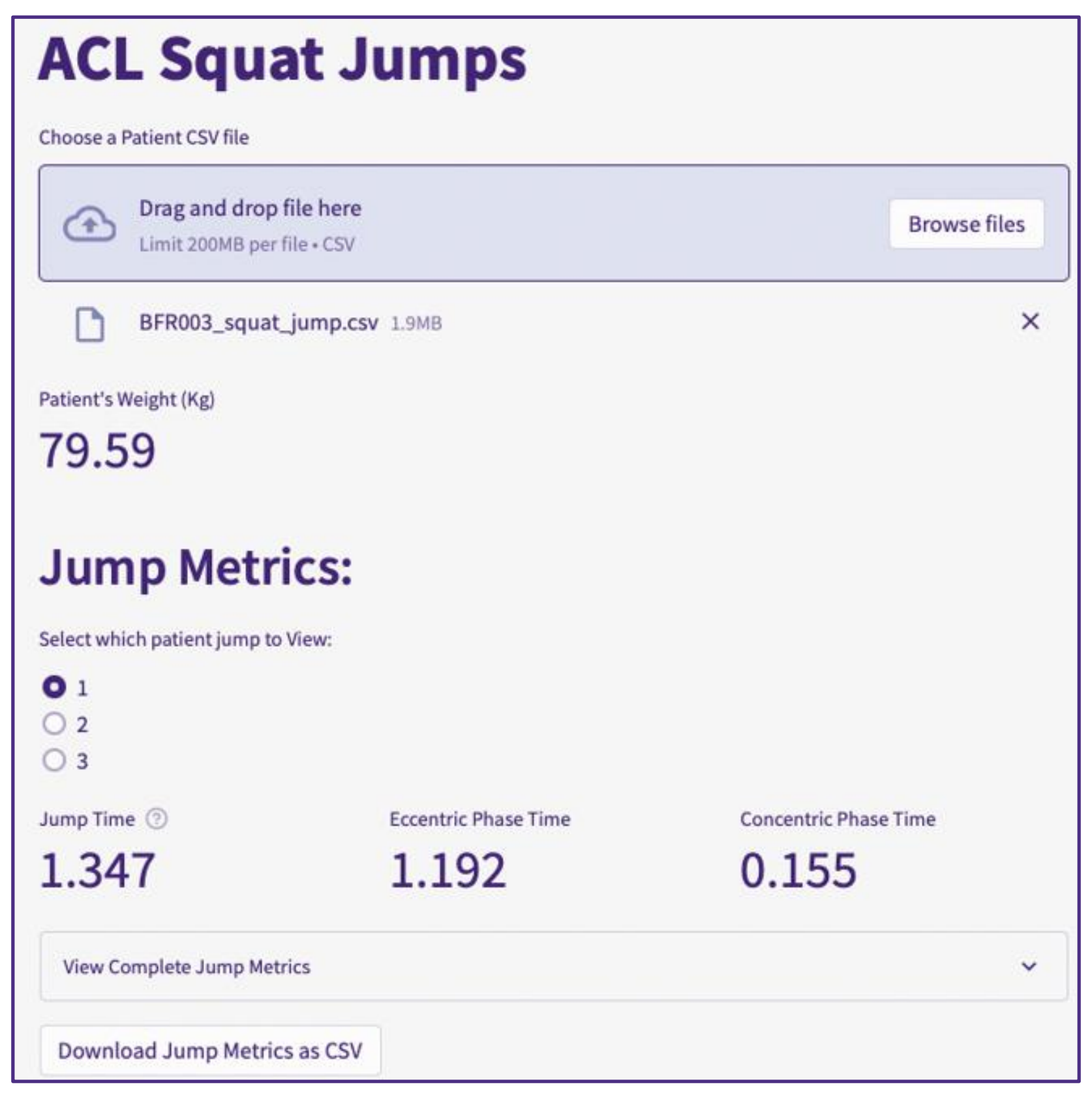


3D Animation For Force

Technology Used

How To Use the Dashboard

- The web application was made using the Streamlit Python framework.
- To use this dashboard, you can navigate to the web address and directly upload CSV force jump data gathered from force plates An about page also exists to provide more details on use cases.


ACL Squat Jumps

Choose a Patient CSV file

Drag and drop file here
Limit 200MB per file • CSV
Browse files

BFR003_squat_jump.csv 1.9MB

Patient's Weight (Kg)
79.59

Jump Metrics:

Select which patient jump to View:
[✓ 1] 2 3

Jump Time	Eccentric Phase Time	Concentric Phase Time
1.347	1.192	0.155

View Complete Jump Metrics

Download Jump Metrics as CSV

The Dashboard

Tool Functionalities

- After uploading the jump data CSV, back-end processing generates key metrics for the patient that is presented in the dashboard. Metrics are presented in a table view, but can also be downloaded as a CSV.
- Visualizations can also be selected to be viewed based on what the user is seeking to understand from the patient data.

Technologies Considered

- Streamlit Python framework was chosen since it is well documented, quick to deploy and iterate. It supports visualization libraries used, such as matplotlib, seaborn, and plotly. Documentation is also extensive, making this the primary framework utilized.
- To visualize our data, we primarily used a mix of matplotlib and plotly. Plotly allowed for interactive visualizations, whereas the quiver plot and funcAnimation options provided by matplotlib allowed us the most flexibility to show dynamic 3D data such as jump unit vectors.

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

python™ plotly Streamlit