

ml-balance

Data processing and analysis for the mediolateral balance project.

Pre-Processing

1. Place the mocap-generated data in the `original_data/` directory.
2. Create a `data/` directory parallel to `original_data/`.
3. Place the `pre-processing/` directory under `original_data/`.
4. Inside `Main.m`, modify the `folder` variable to initiate the data pre-processing process.
5. This process will copy all `.forces` files and process all `.trc` files, saving them as `.csv` files in the corresponding folder under `data/`.

Script

`calculate_com/`

The entry point for this section is `Main.m` within this folder.

- `double_integration_com/`
 - Calculate the center of mass using the double-integration method with `.forces` files using the function in `get_double_com.m`.
 - Use `get_mass()` to retrieve a subject's mass.
- `kinematic_method_com`
 - Calculate the center of mass using the kinematic method from `.csv` files via `KinematicMethod.m`.
 - Use `importfile()` to obtain a data table from a `.csv` file.
 - `HJCEstimation.m` calculates the trace of the hip joint center for a single trial.

`calculate_attributes/`

`real_attributes()` calculates various motion traces, such as pelvis and shoulder movement in the x direction, from single trial data.

`theta_relative_regression/`

- `slope_from_trial()` obtains the slope of `theta_vert` (trunk-pelvis relative angle) vs. `theta_com` from a single trial.
- `slopes_from_person()` retrieves the regression slopes from selected trials of a single subject.
- `get_prediction()` completes the calculation of both the old and new models.
- `model_prediction()` acts as an interface and performs preliminary calculations for `get_prediction`, which is used to calculate both old (90-degree assumption) and new (`theta_relative` or `theta_z` revised) predictions.

- `theta_relative_regression()` serves as a pure interface for `model_prediction()`.

theta_z_prediction/

- Utilizes regression of `theta_p` (trunk-earth z angle) vs. `theta_com`, following the same structure as `theta_relative_regression/`.
- `theta_z_regression()` acts as a pure interface for `model_prediction()`.

cross_validation/

The entry point for this section is `Main.m` within this folder.

- `load_model()` loads the trained model from `models/` using a given sequence number.
- `save_model()` saves the trained model in `models/`.
- `train_test_split()` divides subjects into training and test groups; currently, the test group has only one subject.
- `fit()` employs the given subjects to train a second-order regression model.
- `test()` applies the model to calculate errors in shoulder and pelvis positions for each trial of a single subject.
- `cross_validate()` employs Leave-One-Out Cross-Validation (LOOCV) for a set of subjects and utilizes `process_errors()` to calculate errors, selecting the best model with minimal error.

generate_graph/

The entry point for this section is `Main.m` within this folder.

- `Calvin_prediction/` contains the original code for generating comparison graphs created by Calvin.
- `comparison_graph()` generates comparison graphs for pelvis and shoulder positions for a specific trial.