User Manual: IMU-based Gait Analysis Program (main_.m)

2025-12-22 Huiseok Moon

1. Overview

This program automates the processing of IMU sensor data (.mtb files) for gait analysis. It performs the following main tasks:

① **Data Loading:** Imports raw sensor data.

② **Sensor Reordering:** Maps specific sensors to anatomical locations (Waist, Thigh, Shank, Foot).

③ **Data Segmentation:** Separates calibration phases from walking trials.

④ **Gait Processing:** Calculates joint angles, detects gait events (heel strike/toe off), and computes spatiotemporal parameters.

⑤ **Reporting:** Generates cyclic gait plots and exports a summary CSV file.

2. System Requirements & Setup

2.1. Software

MATLAB: R2020b or later is recommended.

Toolbox: Signal Processing Toolbox, Statistics and Machine Learning Toolbox, Robotics System Toolbox.

Xsens: ver. 2022or later is recommended.

2.2. Directory Structure (Crucial)

The program relies on a specific folder structure. Ensure your project directory looks like this:

```
[Project_Root_Folder]
  ├── main_.m                    % This main script
  ├── Functions/                 % Folder containing analysis functions (processGaitData, etc.)
  └── Dataset/
      └── controle/
          └── [SubjectID]/       % e.g., '01-T-CC'
              ├── File1.mtb      % 1. Barefoot Calibration
              ├── File2.mtb      % 2. Barefoot Fast Walking
              ├── File3.mtb      % 3. Barefoot Preferred Walking
              ├── File4.mtb      % 4. Shoe Calibration
              ├── File5.mtb      % 5. Shoe Fast Walking
              └── File6.mtb      % 6. Shoe Preferred Walking
```

3. Configuration

Before running the script, you must modify the following variable in main_.m:

Line 8: subName

```
subName = '01-T-CC'; % Change this to the target subject's folder name
```

Line 20: fs (Optional)

```
fs = 100; % Sampling frequency (Hz). Change only if sensor settings were different.
```

Line 55: newOrder (Optional)

```
newOrder = [7, 5, 6, 4, 3, 2, 1];
```

- ✓ This array maps the sensor IDs to body parts.
- ✓ Default Mapping: Waist(7) → NonAff Thigh(5) → NonAff Shank(6) → NonAff Foot(4) → Aff Thigh(3) → Aff Shank(2) → Aff Foot(1).
- ✓ If you changed the physical attachment order of the sensors, update these numbers.

4. Execution Steps

Step 1: Run the Script

Open main_.m in MATLAB and press F5 (Run).

Step 2: Data Loading (First Run vs. Repeated Runs)

- ✓ First Run: The script will detect that the .mat file does not exist. It will parse all .mtb files in the subject folder. This may take some time. Once finished, it saves a [SubjectID]_outputData.mat file.
- ✓ Repeated Runs: If the .mat file exists, the script loads it instantly, skipping the parsing process.

Tip: If you update the raw .mtb files, delete the .mat file to force a reload.

Step 3: Data Indexing (Segmentation)

- ✓ The script calls divideDataIntoSections. Depending on the function's implementation:
- ✓ It may automatically split the data based on time.

1. Or it may pop up a graph requiring you to manually select start/end points for:
2. Calibration poses (8 sections).
3. Walking trials (1 section).

Step 4: Analysis Processing

The script processes four specific conditions automatically:

1. Barefoot - Fast
2. Barefoot - Preferred
3. Shoe - Fast
4. Shoe - Preferred

5. Outputs

Upon completion, the following outputs are generated:

5.1. Command Window Output

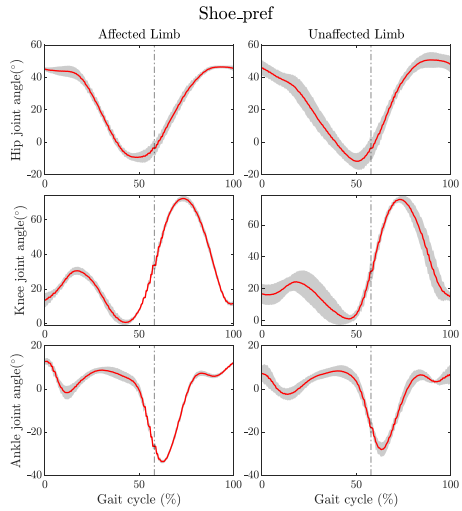
The script prints the sensor mapping and a summary of gait parameters for each

```
==========================================
Device Orders:
==========================================

waist_IMU: [DeviceID]

NonAff Thigh: [DeviceID]

...
```

condition.

5.2. Figures (Plots)

Four figures will be generated (one for each walking condition). These plots visualize the Cyclic Gait Data (Joint angles averaged over the gait cycle).



5.3 Print gait summary

The script prints summary of gait results for each

```
==========================================
Gait Evaluation Summary for: Bare_fast
==========================================

* Primary Evaluation Criterion:
- Peak knee flexion (paretice limb, swing phase): 73.83 ± 1.27 deg
**Secondary Evaluation Criteria:
- Peak hip flexion (paretice limb, swing phase): 37.55 ± 1.62 deg
- Peak hip extension (paretice limb, stance phase): -26.40 ± 1.34 deg
- Stride length (non-paretic limb): 1.45 ± 0.01 m
- Walking speed: 0.94 ± 0.00 m/s...
```

5.4. CSV Export

A file named [SubjectID]GaitSummary.csv is created in the root folder. This file contains the calculated gait parameters for further statistical analysis.

	A	B	C	D	E
	TCCGaitSummary				
	File	Parameter	Mean	Std	Unit
	범주	범주	숫자	숫자	범주
1	File	Parameter	Mean	Std	Unit
2	Bare_fast	WalkingSp...	0.94466912...	0	m/s
3	Bare_fast	Affected str...	1.34948316...	0.02476349...	m
4	Bare_fast	Affected str...	0.746	0.03049590...	s
5	Bare_fast	Affected st...	0.412	0.02588435...	s
6	Bare_fast	Affected s...	0.334	0.00547722...	s
7	Bare_fast	Affected st...	55.1884411...	1.26544284...	s
8	Bare_fast	Non-pareti...	1.45342625...	0.00681681...	m
9	Bare_fast	Non-pareti...	0.734	0.02509980...	s

6. Critical Parameter Tuning: Swing Detection

⚠ **IMPORTANT:** The accuracy of gait event detection (Heel Strike / Toe Off) depends heavily on the Swing Detection Algorithm. Since gait patterns vary by individual (especially in pathological gait), you must tune the detection parameters in processGaitData.m for best results.

6.1. Swing Detection Parameters (set_params)

Inside the processGaitData function, you will find the set_params vector. Adjust these values if the algorithm misses steps or detects false steps.

```
% Found in processGaitData.m (approx. line 52)
set_params = [100, 30, -300, 30, 10]; % Format: [ Dh, Dl, Ds, Tm, Td ]
```

Parameter	Symbol	Description	Tuning Advice
High Threshold	Dh	Threshold for the filtered difference signal (Df) to start a swing phase.	Increase if the algorithm falsely detects swings during stance. Decrease if it misses the start of a swing.
Low Threshold	Dl	Threshold for Df to identify the stance phase.	Signals below this value are considered stance.
Derivative Threshold	Ds	Threshold for the derivative (Df_dot). Controls the sensitivity of the swing-to-stance transition.	A more negative value makes the transition detection stricter.
Min Swing Timer	Tm	Minimum duration (in samples) before the algorithm checks for a stance transition.	Prevents the swing phase from ending too early due to signal noise.
Stance Timeout	Td	Maximum continuous stance counter.	Resets the internal state machine if stance lasts too long (e.g., standing still).

6.2. Automatic Artifact Removal

The script includes a safety feature to remove short, invalid swing artifacts (noise) that may appear as brief spikes in the detection signal.

- ✓ **Function:** remove_short_pulses(Sw, min_length)
- ✓ **Default Setting:** min_length = 30
- ✓ **Behavior:** Any detected swing phase that lasts fewer than **10 samples** (approx. 300ms at 100Hz) is automatically deleted (set to 0).