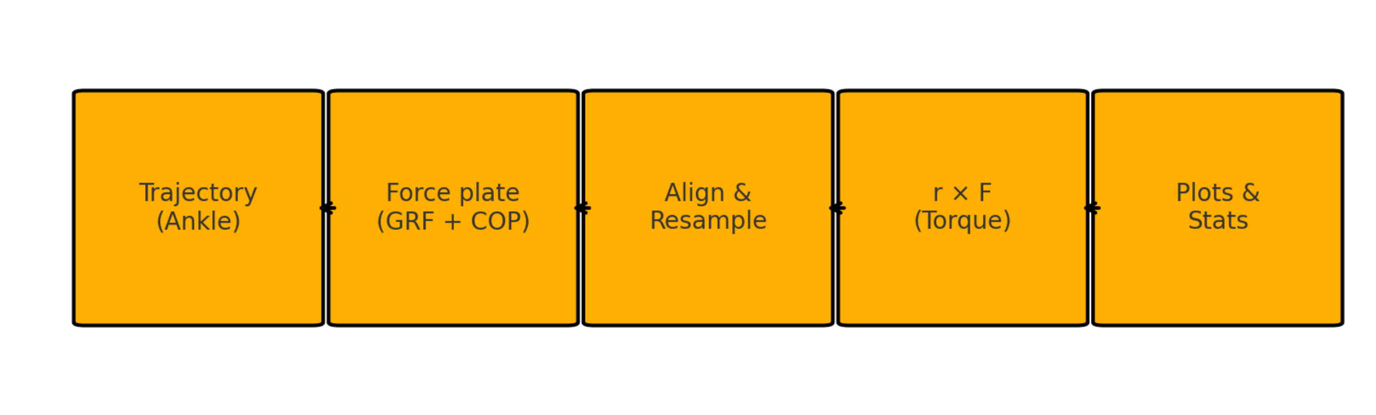
BIOMECHANICAL ANALYSIS OF ANKLE TORQUE IN HUMAN MOVEMENT

ANALYSIS OF HUMAN GAIT BIOMECHANICS BY CALCULATING THE ANKLE TORQUE

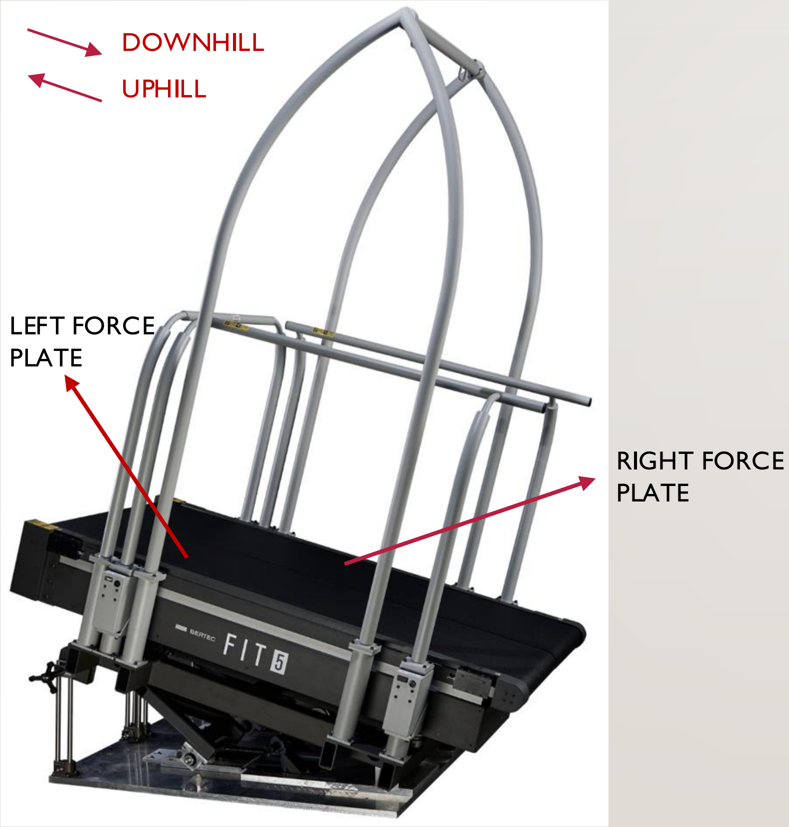
DATA FLOW AT A GLANCE

1. Marker trajectories → ankle centre (global frame)
2. Force-plate file → GRF + COP (local frame)
3. Align, rotate, resample
4. Cross-product $\tau = r \times F$
5. Visualise & export stats



FORCE PLATE: BERTEC TREADMILL FIT GEN 5

FORCE PLATE: BERTEC TREADMILL FIT GEN 5



Since the force plates are switched during uphill and downhill walking, the left force plate data in downhill walking is used as the right foot data, and the right force plate data is used as the left foot data.

FORCE-PLATE META (DUAL-BELT TREADMILL)

- Bertec | 1200 Hz | Length 1778 mm | Width 558.8 mm
- 4 corner markers
- Output forces already in local (+Z = upward GRF)

- Right belt
  Local-origin offset → X = +279.4 mm,
  Y = +889.0 mm, Z = 0 mm

- Left belt
  Local-origin offset → X = –279.4 mm,
  Y = +889.0 mm, Z = 0 mm

CODE STRUCTURE

Cell	Purpose	Output
1	Load libs + helpers + file paths	variables in memory
2	Process one trial (core function)	numpy arrays
3	Ankle-position plots	4-panel figure
4	Force & torque plots	4-panel figure
5	Batch loop / sanity check	console log

DYNAMIC DISPATCH

For subject 2-10

T01 is a trial at 0 degree walking with the boot actuated

T02 is a trial at 0 degree walking with the boot unactuated

T04 is a trial at 5 degrees

T05 is a trial at -5 degrees

T06 is a trial at 10 degrees

T07 is a trial at -10 degrees

- Subject_02
  - T01
    - S02_T01.tsv
    - S02_T01_a.tsv
    - S02_T01_f_6.tsv
    - S02_T01_f_7.tsv
  - T02
  - T05
  - T04
  - T06
  - T07

- Automate file identification and data extraction with code
- Automatically match files based on experiment orientation

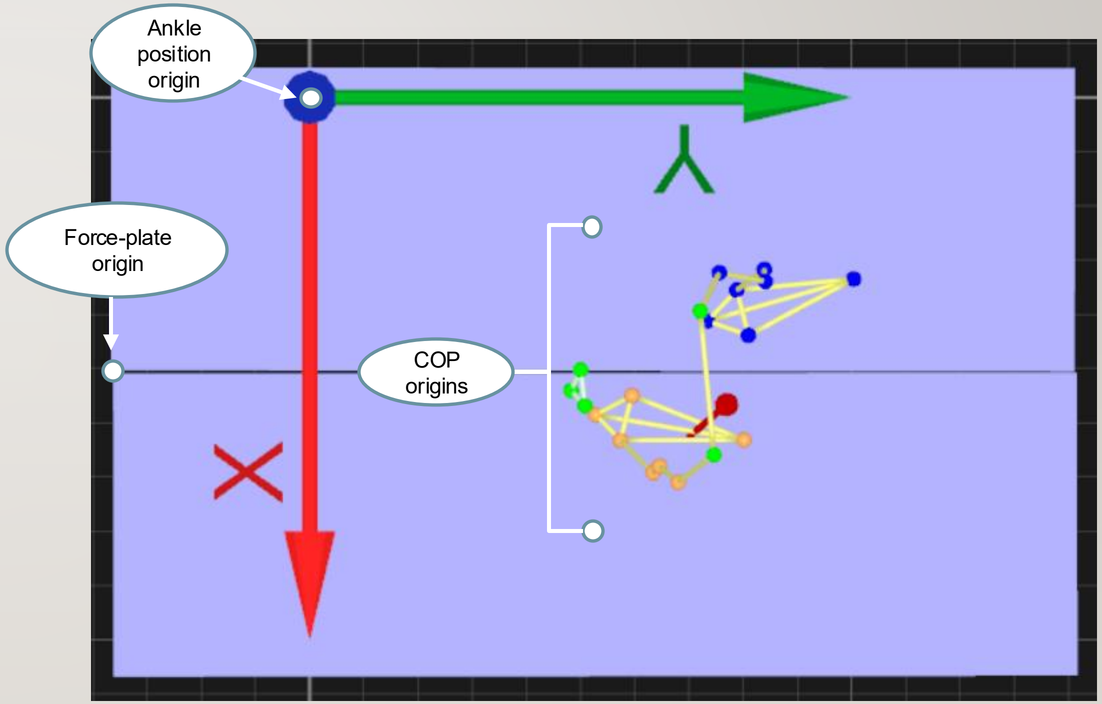
DYNAMIC DISPATCH

```
# 1  loop through all files
for file in root_dir:
    if file == S??_T??.tsv :          # trajectory file
        prefix  = file without ".tsv"
        trial   = number after T

        # choose force filenames
        if trial in (5,7):             # swapped case
            R = prefix + "_f_7.tsv"
            L = prefix + "_f_6.tsv"
        else:                          # normal case
            R = prefix + "_f_6.tsv"
            L = prefix + "_f_7.tsv"

        # add existing pairs
        if R exists:  add (pos , R , "RIGHT")
        if L exists:  add (pos , L , "LEFT")

print total pairs
```

- Scan folders → find every Sxx_Txx.tsv trajectory file
- Derive trial number and decide which force file belongs to each foot
    - normal trials → ?_f_6.tsv = RIGHT, ?_f_7.tsv = LEFT
    - trials T05 / T07 → mapping is swapped
- Verify files exist; discard missing pairs
- Output list [(trajectory, force, "RIGHT/LEFT"), …] for batch processing

FORCE-PLATE HELPER FUNCTIONS

- Translate and rotate the ankle joint trajectory to the same coordinate system as the force plate
- Correctly superimpose COP and GRF and calculate the force arm and joint torque.



FORCE-PLATE HELPER FUNCTIONS

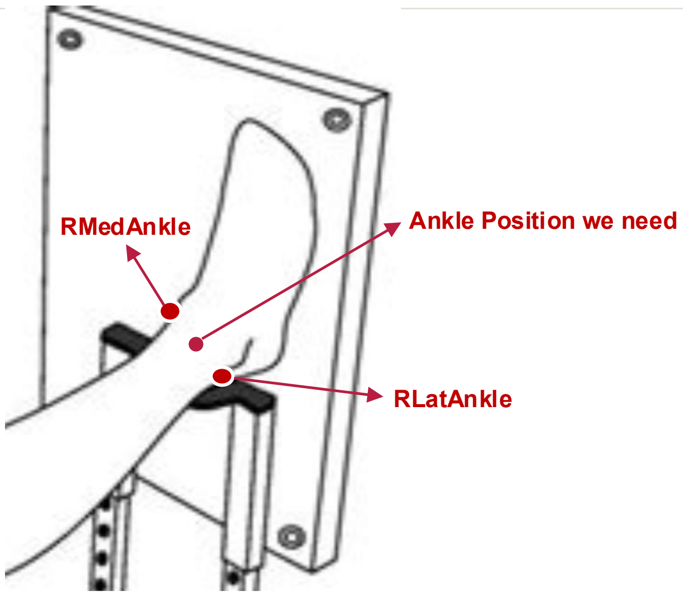
S02_T04_f_6.tsv

```
NO_OF_SAMPLES    161272
FREQUENCY        1200
TIME_STAMP       2023-04-25, 16:09:12.729        15722.59718610
FIRST_SAMPLE     1
DESCRIPTION      Force data in local (force plate) coordinates
DATA_INCLUDED    Force
FORCE_PLATE_TYPE         Bertec
FORCE_PLATE_MODEL        Unknown
FORCE_PLATE_NAME         Right belt
FORCE_PLATE_CORNER_POSX_POSY_X   507.676721
FORCE_PLATE_CORNER_POSX_POSY_Y   1415.532112
FORCE_PLATE_CORNER_POSX_POSY_Z   -1.608300
FORCE_PLATE_CORNER_NEGX_POSY_X   1066.476464
FORCE_PLATE_CORNER_NEGX_POSY_Y   1416.136622
FORCE_PLATE_CORNER_NEGX_POSY_Z   -1.755400
FORCE_PLATE_CORNER_NEGX_NEGY_X   1068.399668
FORCE_PLATE_CORNER_NEGX_NEGY_Y   -361.862302
FORCE_PLATE_CORNER_NEGX_NEGY_Z   -2.971600
FORCE_PLATE_CORNER_POSX_NEGY_X   509.599984
FORCE_PLATE_CORNER_POSX_NEGY_Y   -362.466902
FORCE_PLATE_CORNER_POSX_NEGY_Z   -2.824600
FORCE_PLATE_OFFSET_X     279.400000
FORCE_PLATE_OFFSET_Y     889.000000
FORCE_PLATE_OFFSET_Z     0.000000
FORCE_PLATE_LENGTH       1778.000000
FORCE_PLATE_WIDTH        558.800000
```

Function / Action	Why it matters
extract_force_plate_corners(): • Read lines 10-21 of each *.tsv*	Get the exact geometry of **each belt plate**
calculate_rotation_matrix(corners): • X = front_right − front_left • Y = front_left − back_left • Z = X × Y (right_hand rule)	Converts “plate local” vectors into the **global lab frame**

FORCE-PLATE HELPER FUNCTIONS

- calculate_plate_center_offsets()
  1. Call both helpers for left & right plate.
  2. Average 4 corners → plate geometric centre.
  3. offset = −centre → shifts origin to plate centre.
  4. Return 4 keys ready for pipeline:
     - right_offset, left_offset (mm → m later)
     - right_rotation, left_rotation (3 × 3 each)

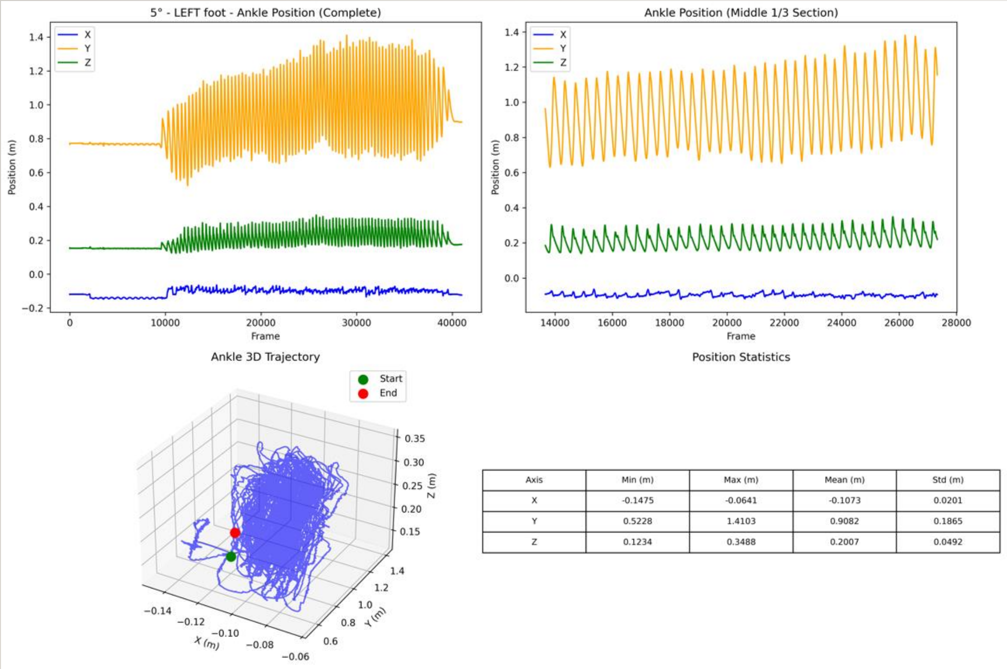



```python
# 4. Calculate the center position of ankle
AnklePosition = (RLatAnkle + RMedAnkle) / 2  # (3, Frames)

AnklePosition /= 1000  # mm → m
```

MARKER → ANKLE CENTRE

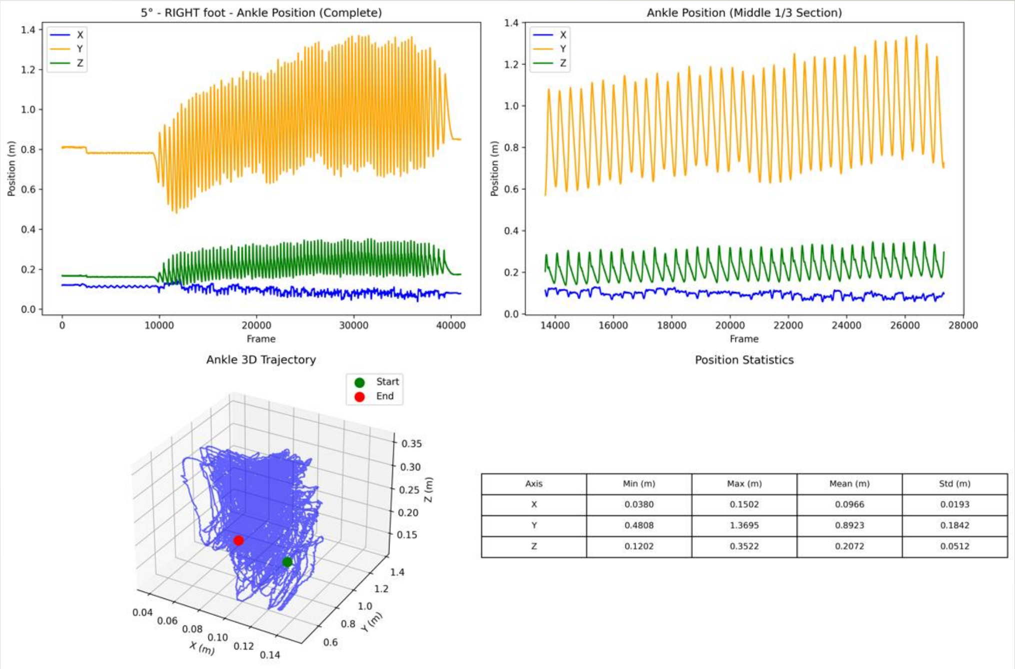
- Read TSV, pick
  LatAnkle + MedAnkle
- Average & convert mm→m
- Apply:
  1. plate offset
  2. plate rotation R_plate^{T}
  3. add global offset
     (±279.4 mm, 889 mm)
  4. **optional** slope-angle rotation
     (±5 °, ±10 °)

ANKLE CENTRE POSITION PLOT



Axis	Min (m)	Max (m)	Mean (m)	Std (m)
X	-0.1475	-0.0641	-0.1073	0.0201
Y	0.5228	1.4103	0.9082	0.1865
Z	0.1234	0.3488	0.2007	0.0492

S04_T04_LEFT

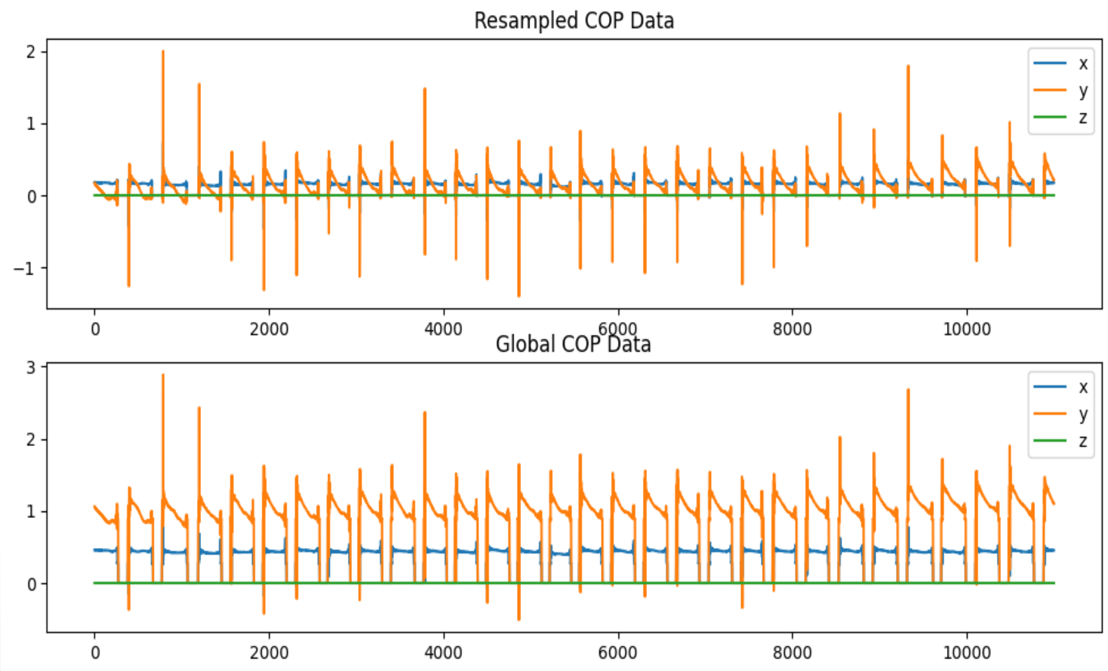


Axis	Min (m)	Max (m)	Mean (m)	Std (m)
X	0.0380	0.1502	0.0966	0.0193
Y	0.4808	1.3695	0.8923	0.1842
Z	0.1202	0.3522	0.2072	0.0512

S04_T04_RIGHT

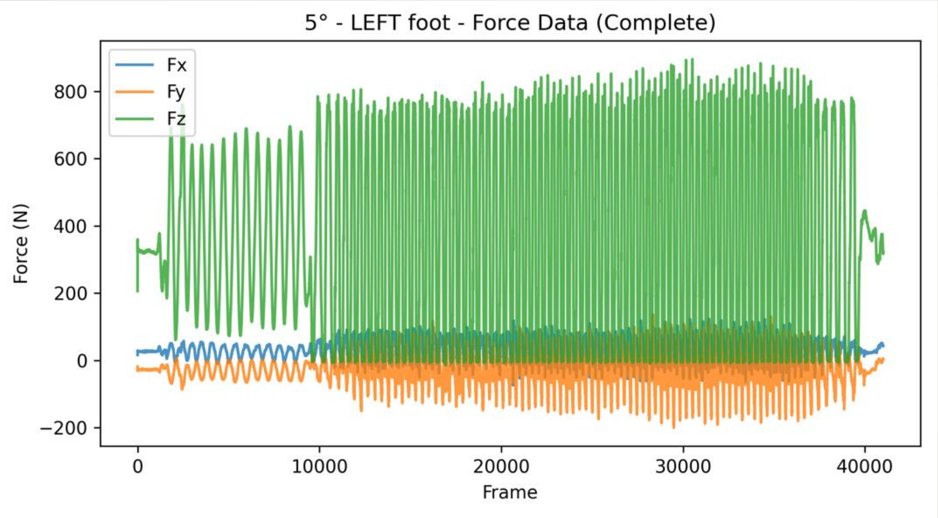
COP DATA PROCESSING

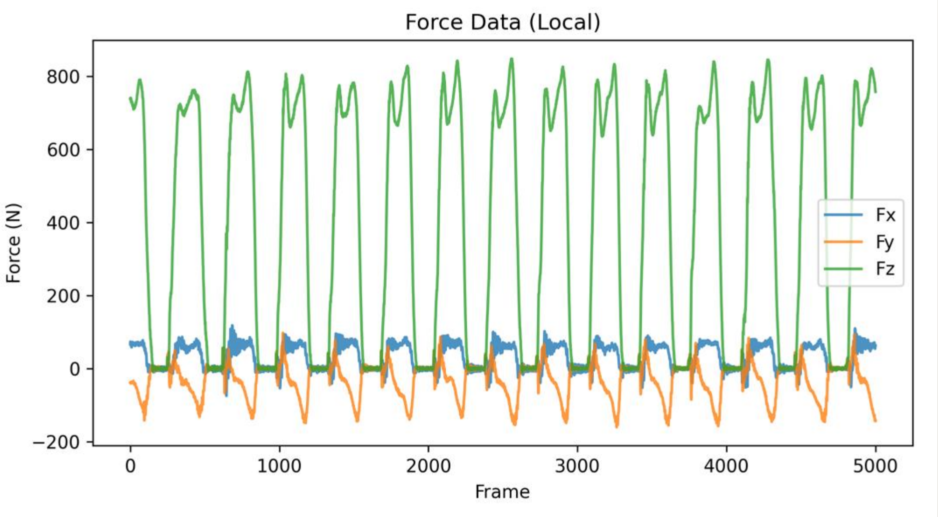
- COP data before and after adding global offset.
- Resample 1200 Hz → 300 Hz
- Offset is applied only to **valid (non-zero) frames**.
- The COP in the Y direction shows a clear forward-to-backward pattern during stance phase. This reflects the natural gait cycle, where the foot initially contacts the ground in front of the body and then rolls back behind as the step progresses.



- ↑ Forward (early stance)
- ↓ Backward (late stance)

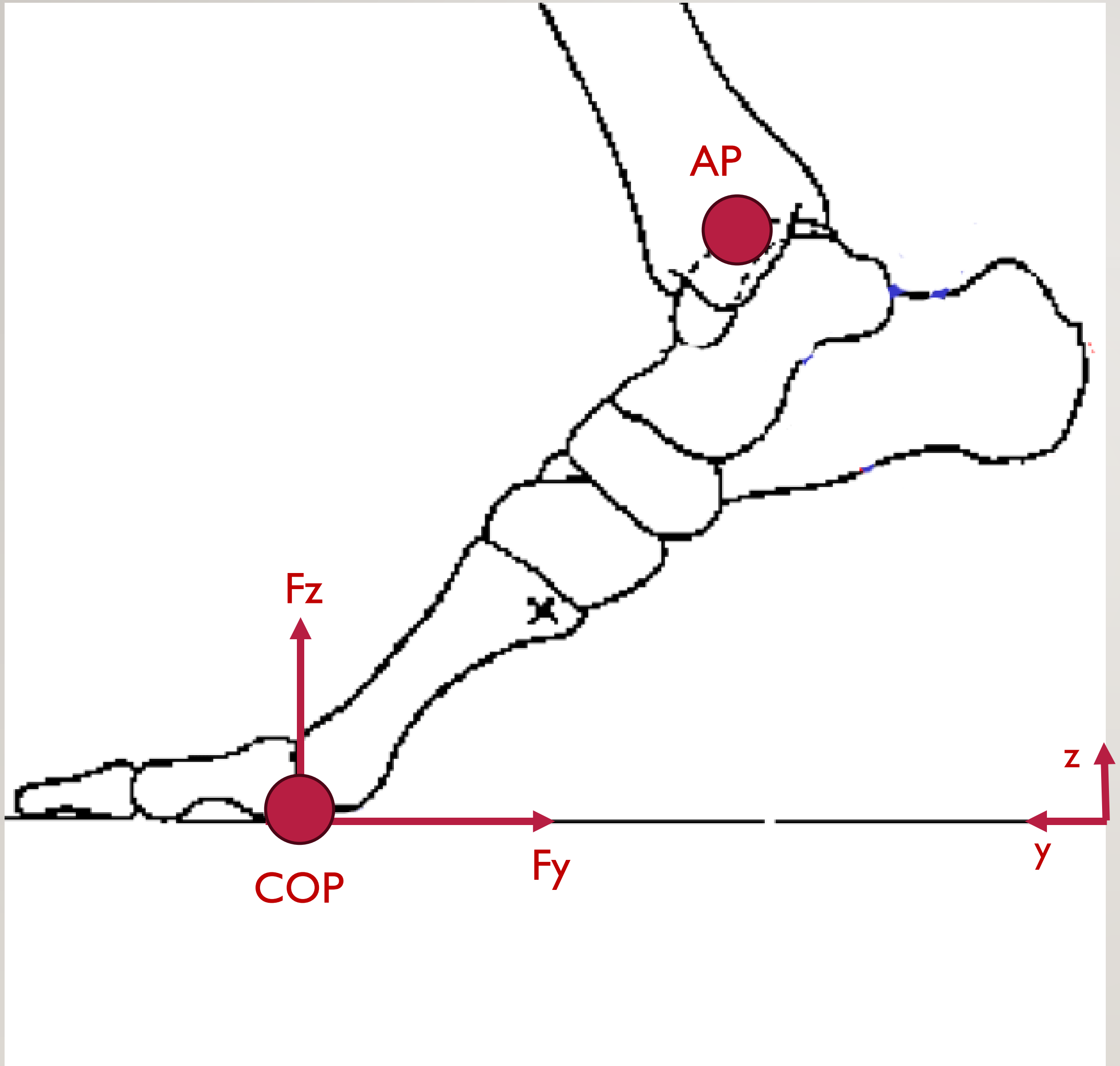
FORCE DATA PLOT





Force Diagram Analysis

- **Axes & Ranges**
  - **F_z (Green):** Dominant vertical force; high range indicates weight/load.
  - **F_y (Orange):** Forward shear; lower range shows side loads.
  - **F_x (Blue):** Lateral sway; lower range shows side loads.
- **Key Features**
  - **Peaks:** Sudden impacts or abrupt load changes.
  - **Slopes:** Gradual rises/falls indicating smooth loading/unloading.
  - **Oscillations:** Periodic motion or sensor noise.
  - **Zero:** When the left leg is off ground.
- **Observations**
  - Clean gait cycles can be seen in the zoomed-in force data (bottom).

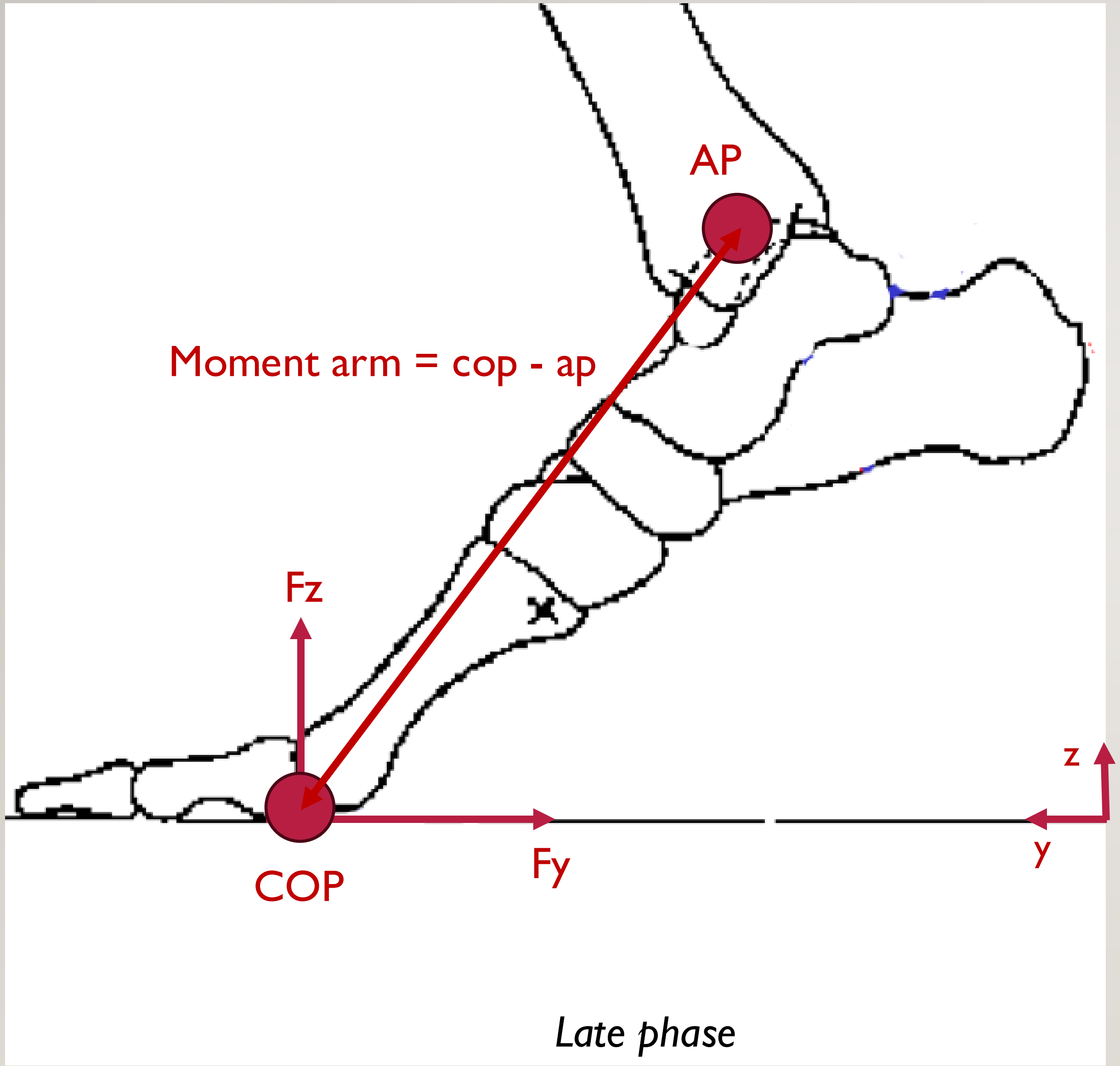


FORCE DATA ANALYSIS

View: Y-Z plane

X axis: out of plane (medial-lateral)

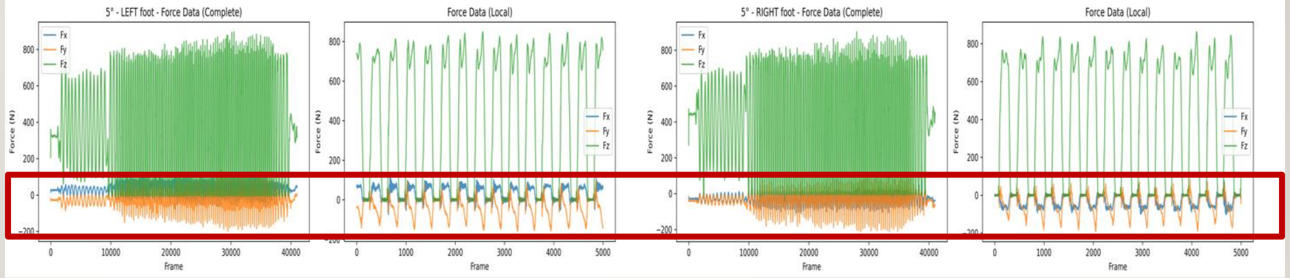
The force vector shown represents the **ground reaction force acting on the foot.**

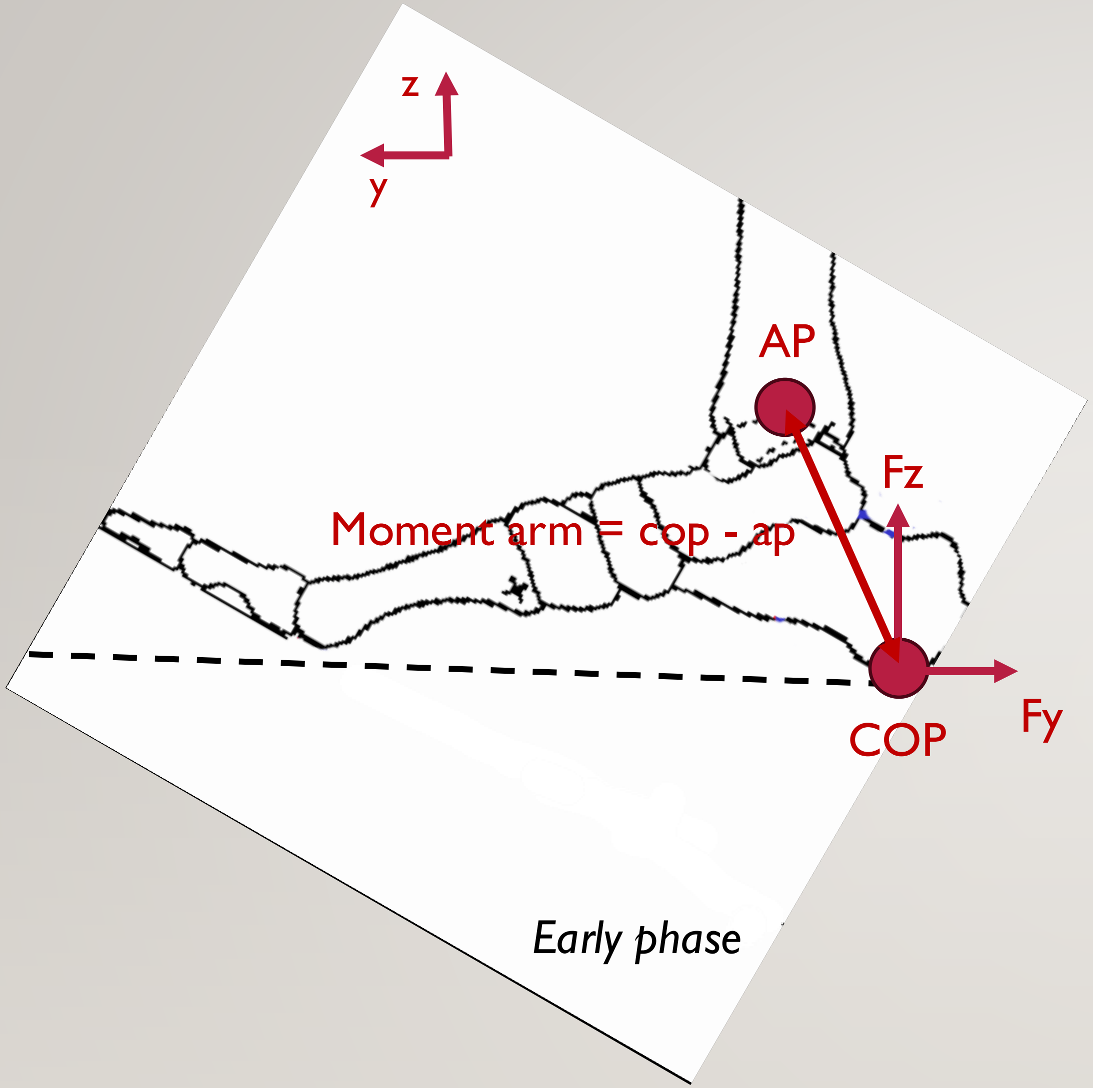


WHY DO WE USE THIS FORMULA?

F_y: mostly negative

- Because the force sensor is the belt, to drag the feet going backward, the force should be negative y.





WHY DO WE USE THIS FORMULA?

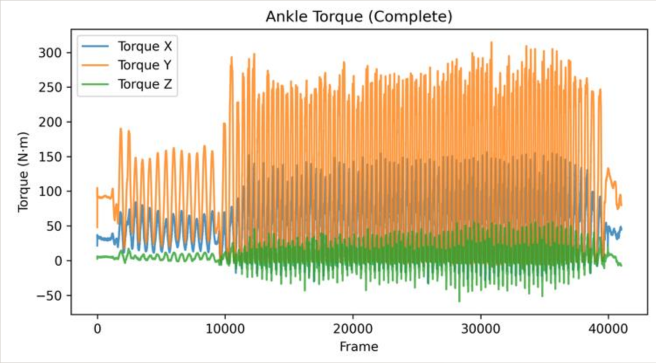
In this diagram, we calculate the moment arm as:

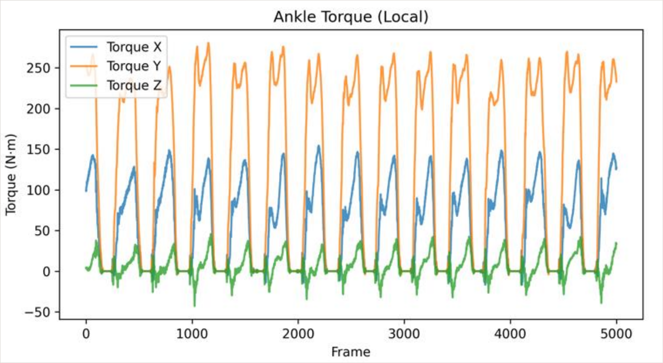
Moment = COP – Ankle

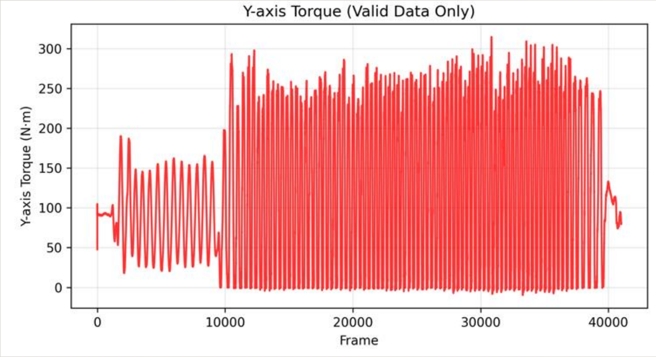
(only with valid data, not [0,0,0])

This vector points from the ankle to the point where the ground force is applied.

The resulting torque describes the rotational effect of that force on the ankle joint.







Torque Statistics (Valid Data)

Axis	Min (N·m)	Max (N·m)	Mean (N·m)	Std (N·m)
X	-32.24	156.47	53.42	41.24
Y	-9.72	314.68	133.22	92.92
Z	-59.05	55.33	6.25	12.53

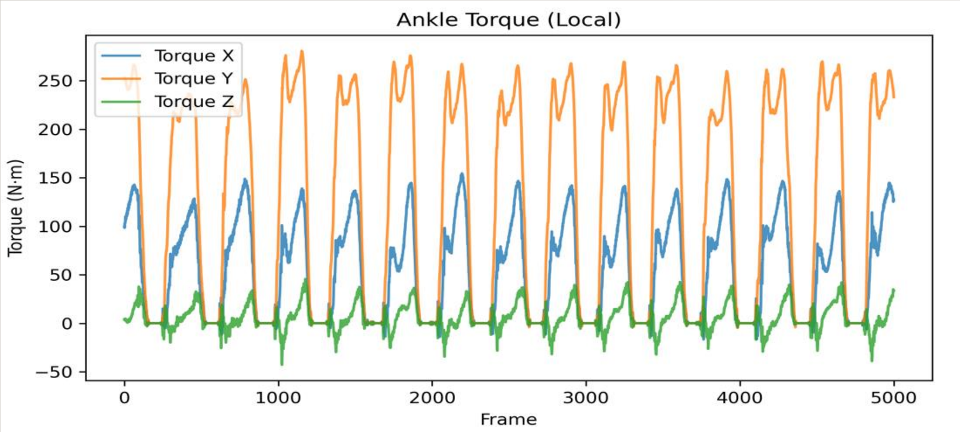
WHY DO WE USE THIS FORMULA?

In this diagram, we calculate the torque as:

$$\boldsymbol{\tau = r \times F}$$

r = COP – Ankle (moment arm)

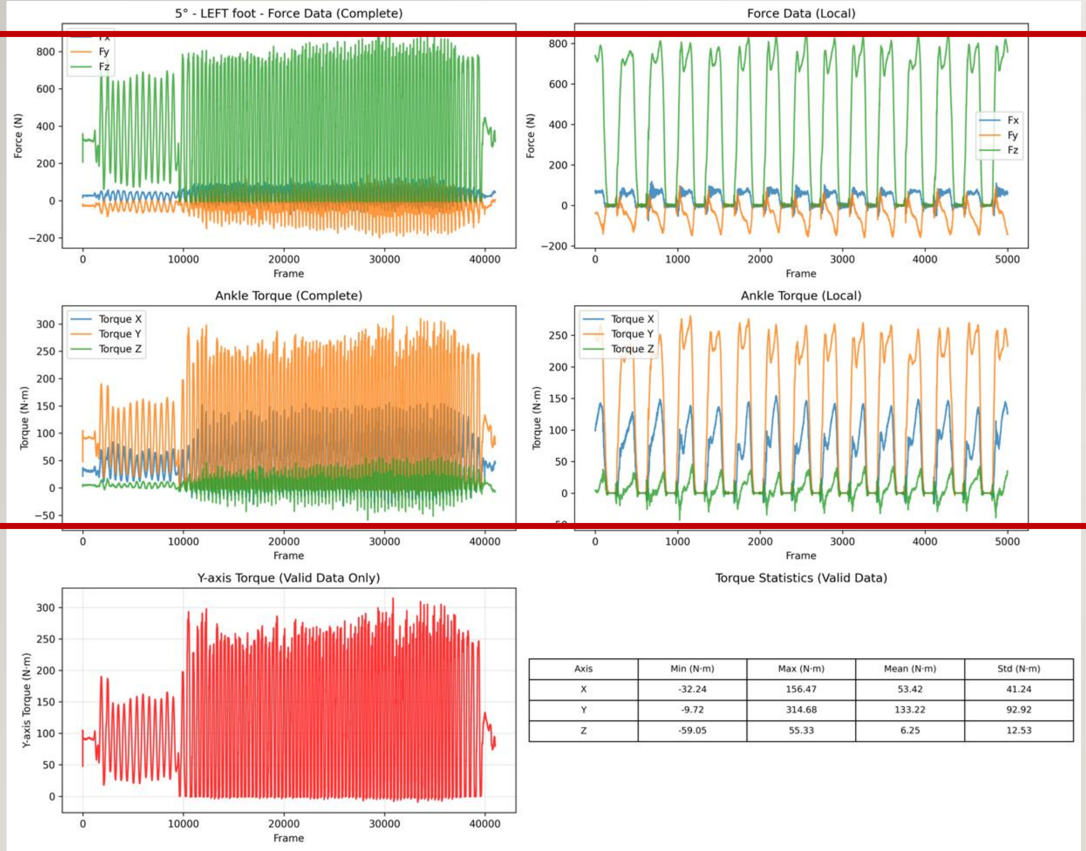
F = GRF (plate → foot)



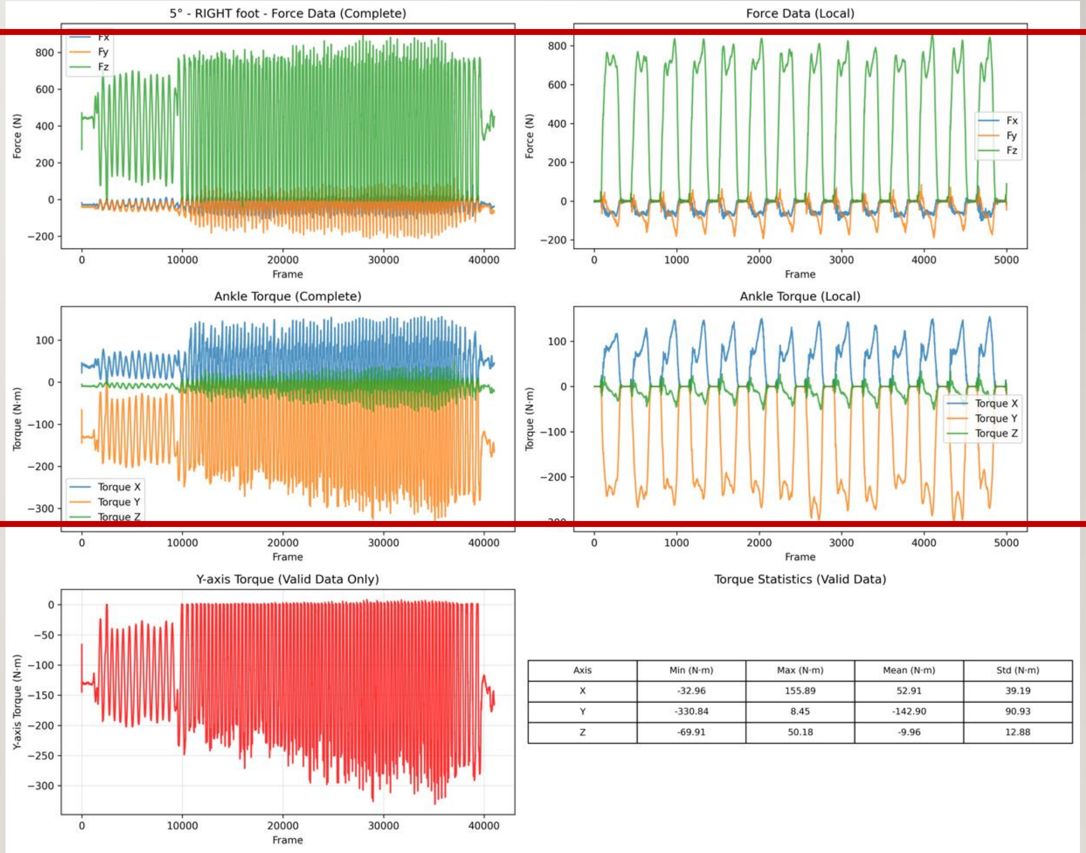
WHY DO WE USE THIS FORMULA?

Quantity	Typical value	Comment
Peak **F_z**	≈ 800 N	80 kg subject × 9.8 m/s², plus impact
Horizontal moment arm **r_y** (medial-lateral)	0.10 – 0.12 m	ankle ↔ COP sideways
Horizontal moment arm **r_x** (anterior-posterior)	0.28 – 0.35 m	toe-out places COP far forward
Cross-product ⇒ **τ_Y peak**	r_x × F_z ≈ 0.32 m × 800 N ≈ 256 N·m	plot shows **314 N·m** → within 20 %
Cross-product ⇒ **τ_X peak**	r_y × F_z ≈ 0.11 m × 800 N ≈ 88 N·m + smaller (–r_z F_y) term → ~150 N·m	plot shows **156 N·m**
τ_z	only from horizontal forces (≤ 70 N) × short arms → < 60 N·m	plot max **55 N·m**

WHY IS THE TORQUE POSITIVE OR NEGATIVE?

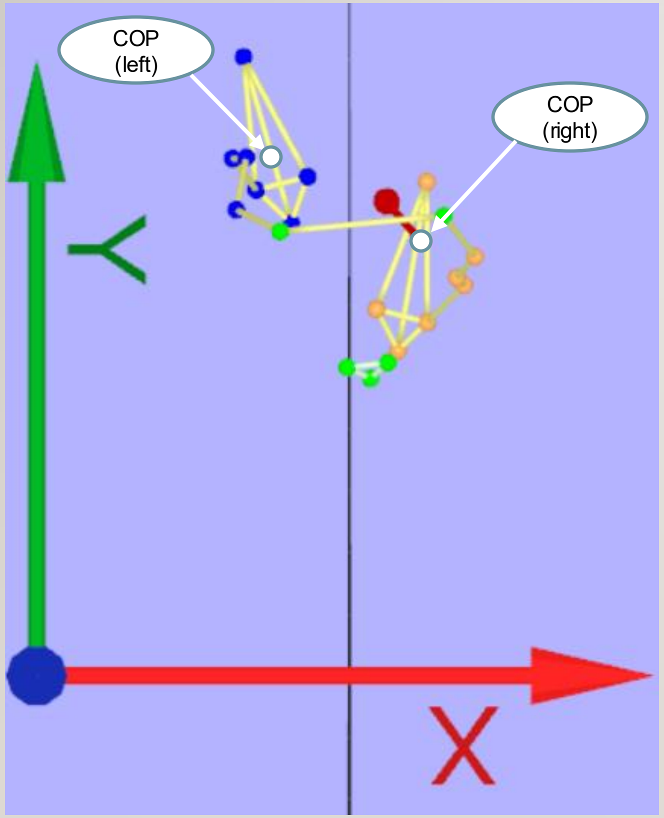


Axis	Min (N·m)	Max (N·m)	Mean (N·m)	Std (N·m)
X	-32.24	156.47	53.42	41.24
Y	-9.72	314.68	133.22	92.92
Z	-59.05	55.33	6.25	12.53

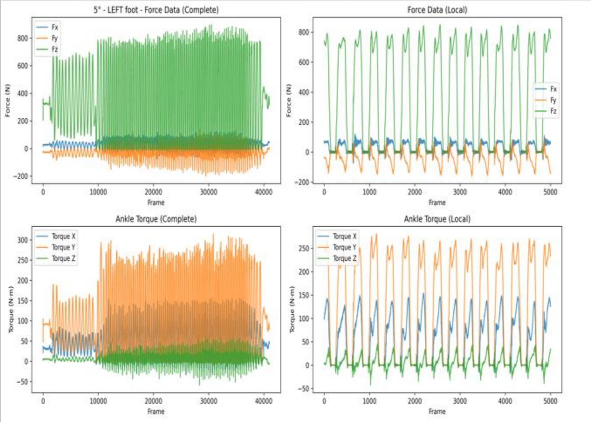


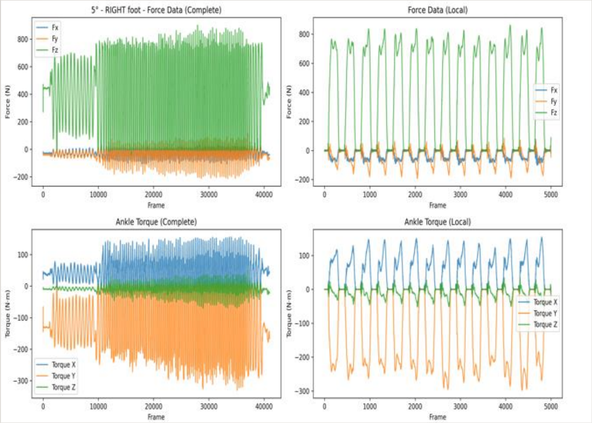
Axis	Min (N·m)	Max (N·m)	Mean (N·m)	Std (N·m)
X	-32.96	155.89	52.91	39.19
Y	-330.84	8.45	-142.90	90.93
Z	-69.91	50.18	-9.96	12.88

WHY IS THE TORQUE POSITIVE OR NEGATIVE?



Vector	Sign (left foot)	Sign (right foot)
r = Ankle → COP	r_x **< 0** ← COP lateral to ankle r_y **> 0** ← ankle is medial to COP r_z **< 0** ← ankle above plate (-Z)	r_x **> 0** ← (same) r_y **> 0** ← COP medial but on opposite side r_z **< 0** ← (same)
F = GRF (belt → foot)	F_x **> 0** ← horizontal force points **inward** F_y **< 0** ← the belt drags the feet backward F_z **> 0** ← upward load	F_x **< 0** ← (same) F_y **< 0** ← (same) F_z **> 0** ← (same)





$$\boxed{\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}} \implies \begin{cases} \tau_x &= r_y F_z - r_z F_y \\ \tau_y &= r_z F_x - r_x F_z \\ \tau_z &= r_x F_y - r_y F_x \end{cases}$$

WHY LEFT_Y_TORQUE "+", BUT RIGHT "-" ?

Sign (left foot)

τ_x (sagittal)

- r_y × F_z → (+)(+) = **positive**
- –(r_z × F_y) → –[(-)(-)] = **positive**
- **Result:** $\tau_x > 0$ (plantar-flexion direction)

τ_y (frontal)

- r_z × F_x → (-)(+) = **negative**
- –(r_x × F_z) → –[(-)(+)] = **positive**
- r_x & r_z are tiny, |F_z| is much greater than |F_x|, so the result is **positive**
- **Result:** $\tau_y > 0$

τ_z (transverse)

- r_x × F_y → (-)(-) = **positive**
- –(r_y × F_x) → –[(+)(+)] = **negative**
- r_x & r_z are tiny, mostly **|F_y| > |F_x|**, so most of the result is **positive;** when **|F_y| < |F_x|**, the result is **negative**
- **Result:** $\tau_z > 0$, a small part of $\tau_z < 0$

Sign (right foot)

τ_x (sagittal)

- r_y × F_z → (+)(+) = **positive**
- –(r_z × F_y) → –[(-)(-)] = **positive**
- **Result:** $\tau_x > 0$ (plantar-flexion direction)

τ_y (frontal)

- r_z × F_x → (-)(-) = **positive**
- –(r_x × F_z) → –[(+)(+)] = **negative**
- r_x & r_z are tiny, |F_z| is much greater than |F_x|, so the result is **negative**
- **Result:** $\tau_y < 0$ (external inversion)

τ_z (transverse)

- r_x × F_y → (+)(-) = **negative**
- –(r_y × F_x) → –[(+)(-)] = **positive**
- r_x & r_z are tiny, mostly **|F_y| > |F_x|**, so most of the result is **negative;** when **|F_y| < |F_x|**, the result is **positive**
- **Result:** $\tau_z < 0$, a small part of $\tau_z > 0$

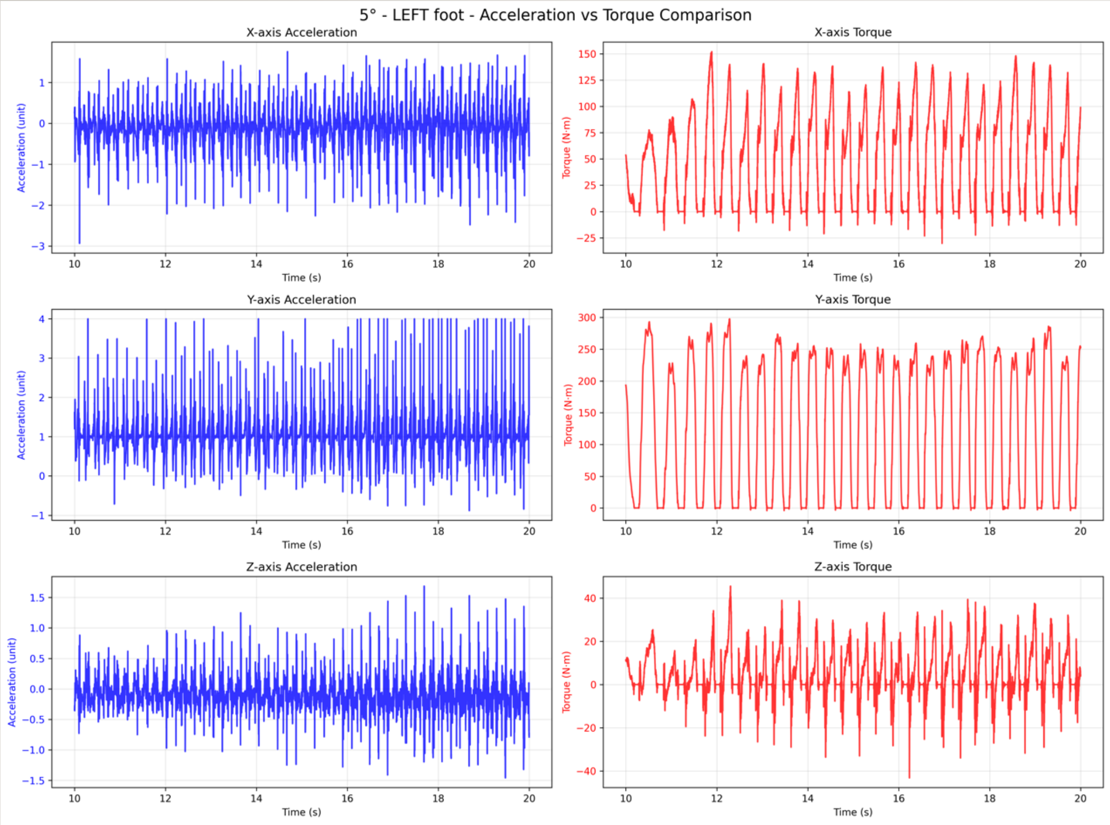
TORQUE DATA INSERTION

sync
1
1
1
1
0
0
0
0

- Torque data is inserted into the corresponding columns starting from the point where the sync value changes from 1 to 0.

						FORCE_Z									TORQUE_X/ Y/ Z	
1	3	0	0	0	0		0	0	0	0	1	0	-1			
1	3	0	0	0	0		0	0	0	0	1	0	-1			
1	3	0	0	0	0		0	0	0	0	1	0	-1			
0	3	0	0	0	0	461.2798703	0	0	0	0	1	0	-1	124.348113	95.2162666	12.9880782
0	3	0	0	0	0	460.6931904	0	0	0	0	1	0	-1	203.093058	139.953134	20.9490236
0	3	0	0	0	0	460.3075503	0	0	0	0	1	0	-1	184.00781	129.272954	19.3567379
0	3	0	0	0	0	460.0812757	0	0	0	0	1	0	-1	196.33661	136.681585	20.1792601
0	3	0	0	0	0	460.0245695	0	0	0	0	1	0	-1	188.695091	132.723541	19.4621563
0	3	0	0	0	0	460.1843807	0	0	0	0	1	0	-1	191.399233	134.828477	19.2769782

COMPARISON OF ACCELERATION AND TORQUE



- Side-by-side layout – left = ankle linear acceleration; right = computed joint torque. Three stacked rows → X, Y, Z axes.
- Step-cycle synchrony
  - Spiky acceleration bursts (heel-strike / push-off) recur ≈ 1 Hz.
  - Corresponding torque curves show matching periodic ramps.
- Axis specifics
  - X-axis ±3 m/s² jitter co-occurs with 0 → 150 N·m plantar-flexion pulses.
  - Y-axis largest swings: 4 m/s² spikes align with 0 → 300 N·m eversion/ inversion moments.
  - Z-axis vertical impacts < 1.5 m/s² pair with modest ± 40 N·m transverse torque.

THANK YOU!