ॐ

CS698F Final Demo

TEAM 1:

ASHWIN SHENAI	(180156)
ISHANH MISRA	(180313)

Gesture Recognition

Basic DTW implemented for recognizing simple gestures.

Decision for "Distance" metric d(x_i, y_j):

- Cosine Similarity Metric:

$$d[i][j] = (1 - scale * \frac{Acc_i * Acc_j}{\|Acc_i\| * \|Acc_j\| + 1 * 10^{-6}}) * Norm$$

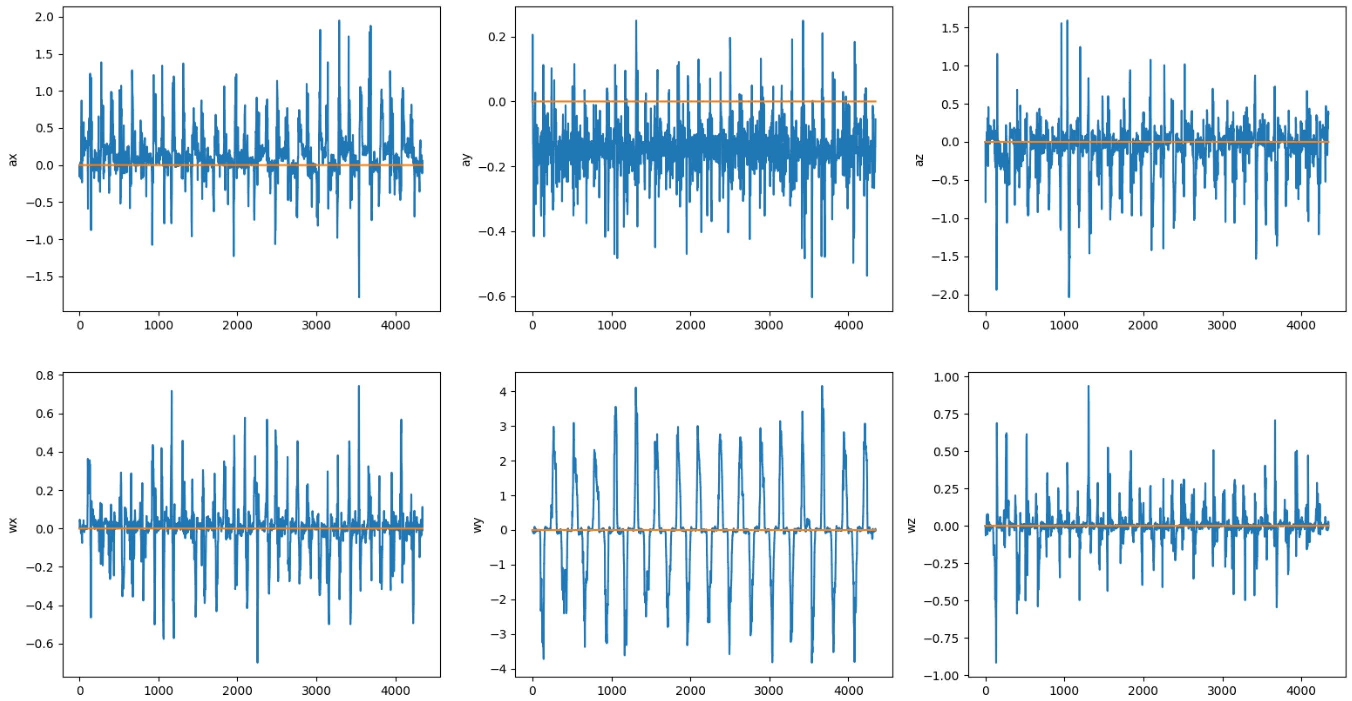
Gestures we've worked with:

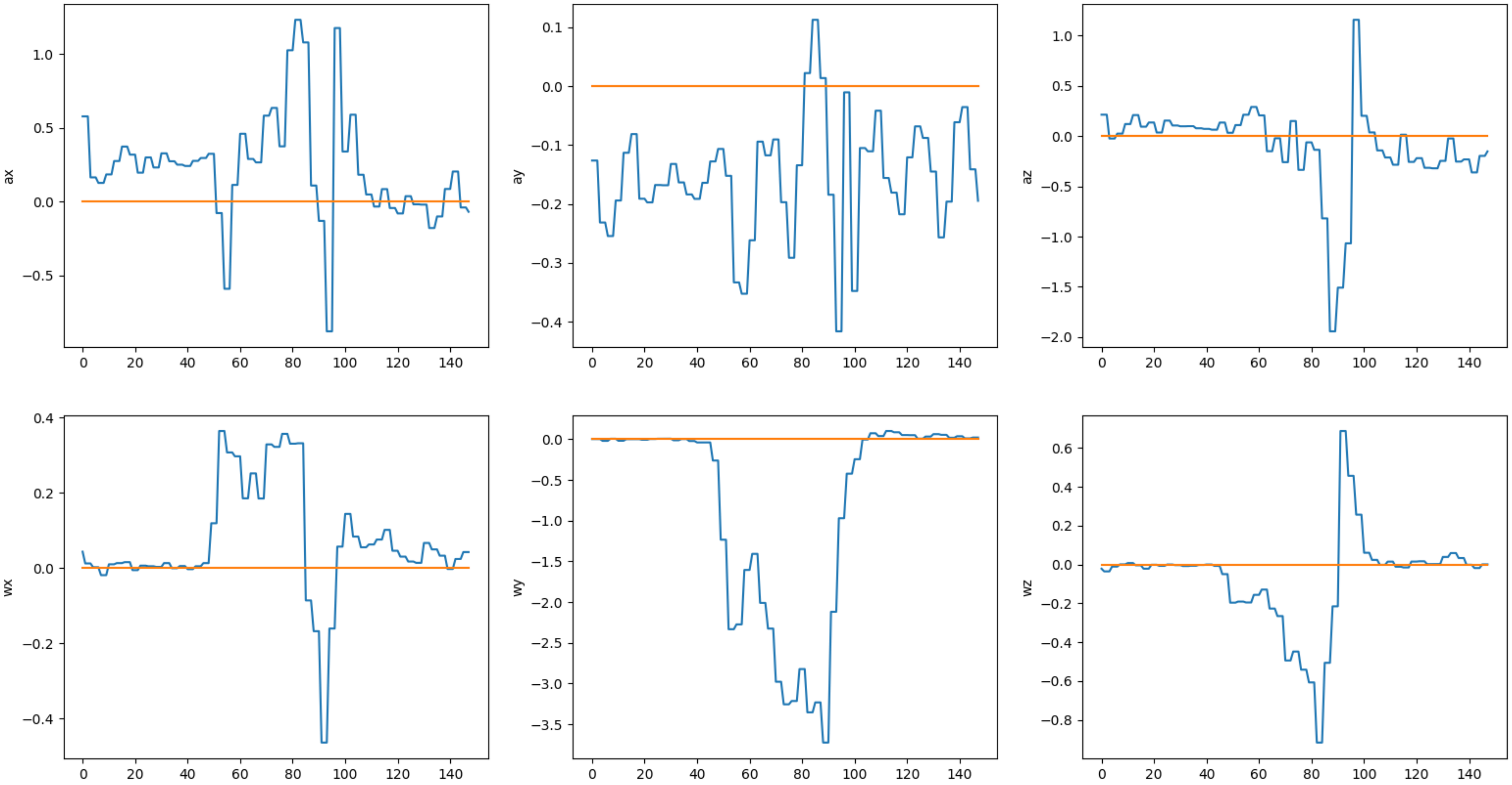
- Hand going forward/backward, making a cross and a wave.
- up/down, right/left, rotational hand movement

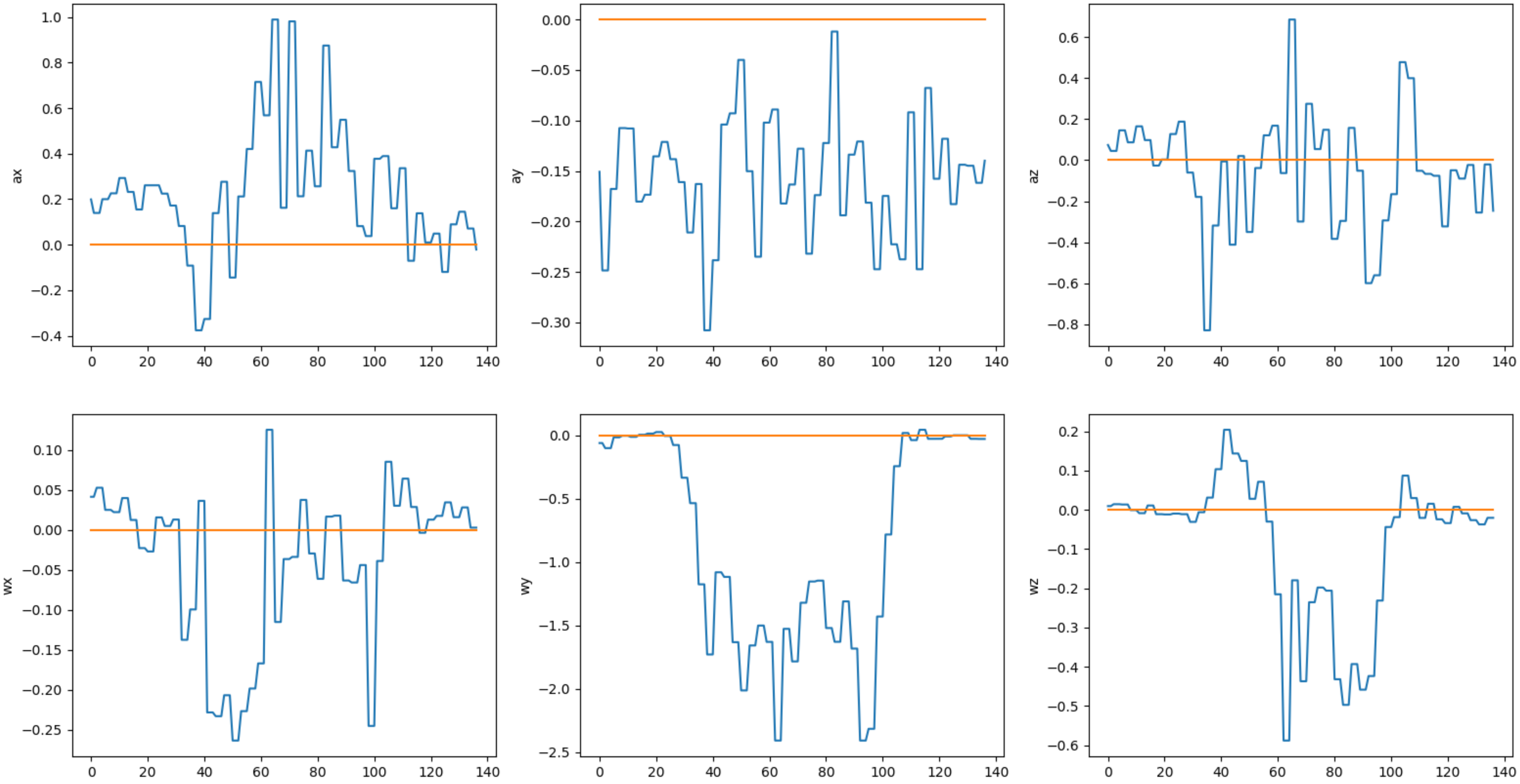
Data Collection

- Used Physics Toolbox to collect accelerometer data for the following gestures:
  - Forward, Backward, Wave, Cross
  - Bow up, Bow down, Flip, Flop
  - Up, Down, Left, Right
- Gestures were recorded at low speed – each gesture spans about 5 seconds
- Data was recorded at minimum possible sampling rate – 100 Hz frequency
- Multi Record feature used to simultaneously record accelerometer data

Data: Flip / Flop




ax
1.0
0.5
0.0
−0.5
ay
0.1
0.0
−0.1
−0.2
−0.3
−0.4
az
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0 20 40 60 80 100 120 140
wx
0.4
0.2
0.0
−0.2
−0.4
wy
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
wz
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0 20 40 60 80 100 120 140

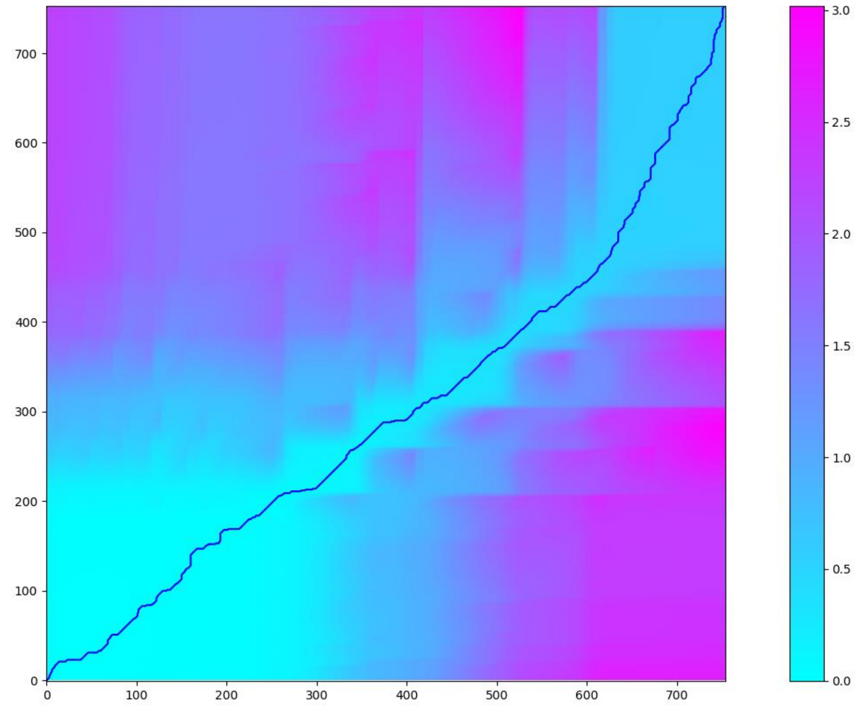


ax
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
20
40
60
80
100
120
140
ay
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
az
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
wx
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
wy
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
wz
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6

Basic DTW

- O(MN) runtime
- All combinations need not be checked
- Impose global constraints
- Approximate optimal solutions acceptable

$$d[i][j] = (1 - scale * \frac{Acc_i * Acc_j}{\|Acc_i\| * \|Acc_j\| + 1 * 10^{-6}}) * Norm$$

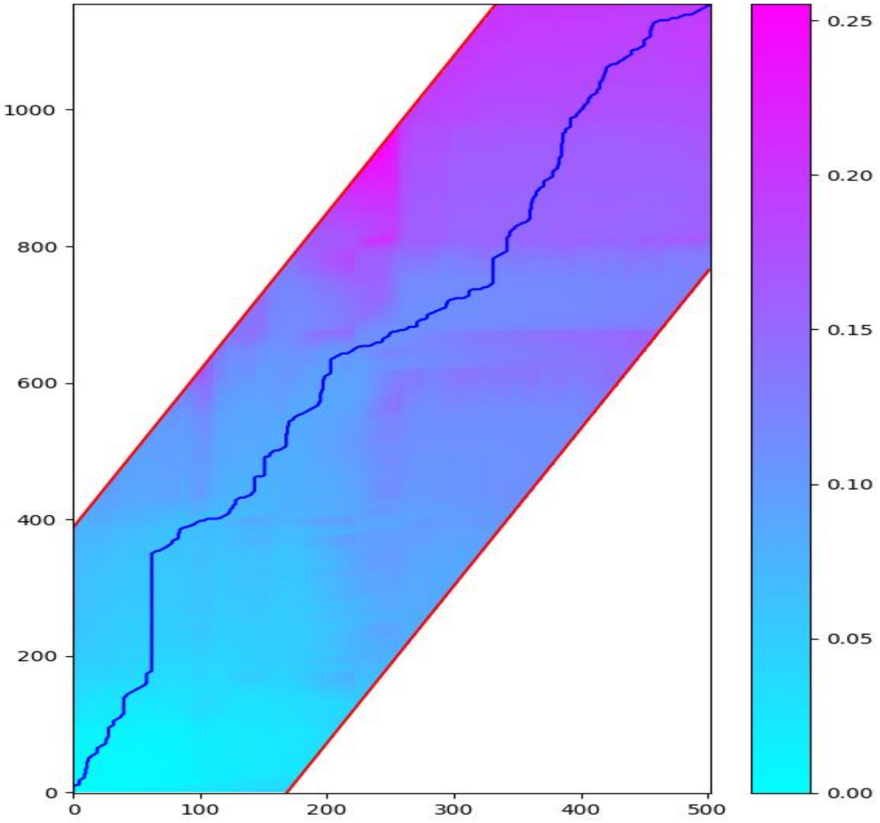


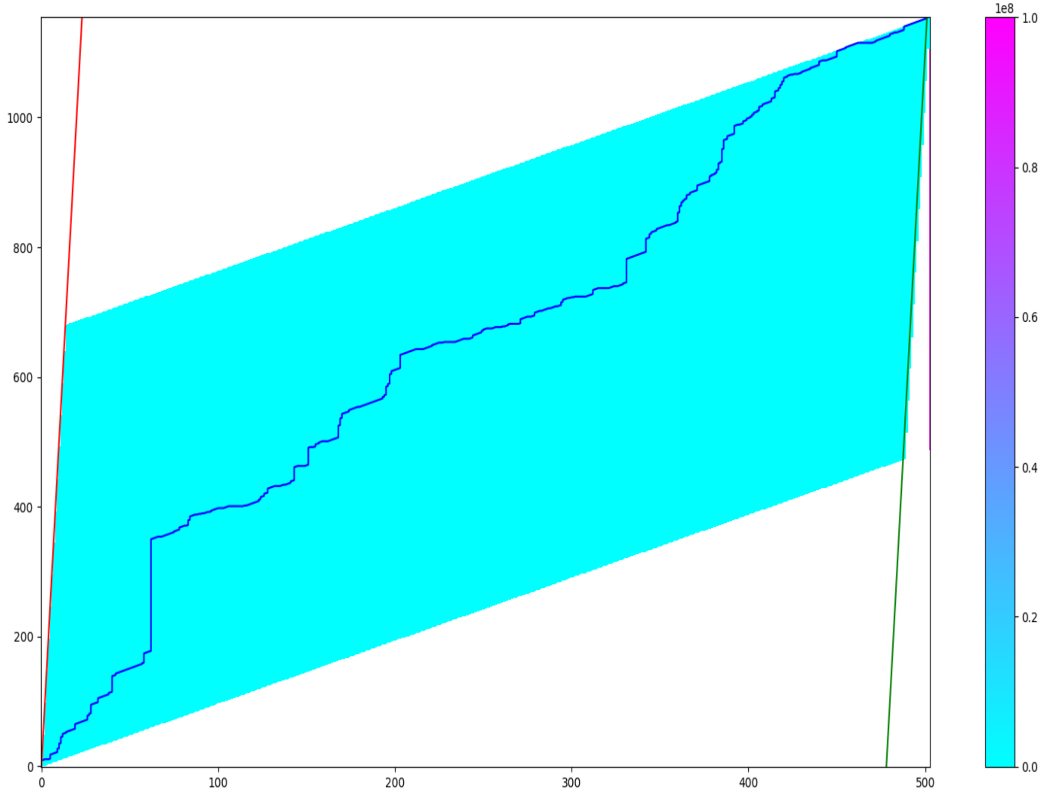
DTW path (matching signal) from Cosine Metric

Key Focus: Optimizing DTW

Sakoe Chiba Band

Itakura Parallelogram





Runtime Comparison

Variant	Length of X	Length of Y	Average Runtime (s)
Vanilla	502	1154	5.65
Vanilla	753	1154	7.99
Sakoe Chiba	502	1154	3.43
Sakoe Chiba	753	1154	4.75
Itakura	502	1154	5.61
Itakura	753	1154	7.17

Can try to analyze/graph asymptotic behaviour – will attempt to include in the report

FastDTW

- Approximate DTW in linear time and space
- Recursively applies DTW at different granularities
- Overall recursion is bounded in O(N)
- Additionally applies global constraints for more speedup
- Alignment is not straightforward to compute, only total cost is readily accessible

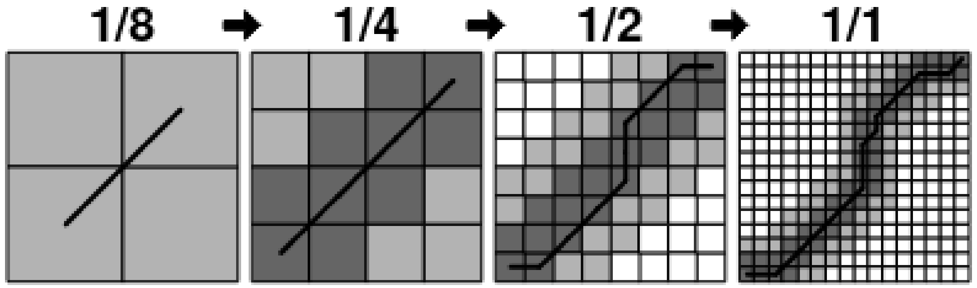
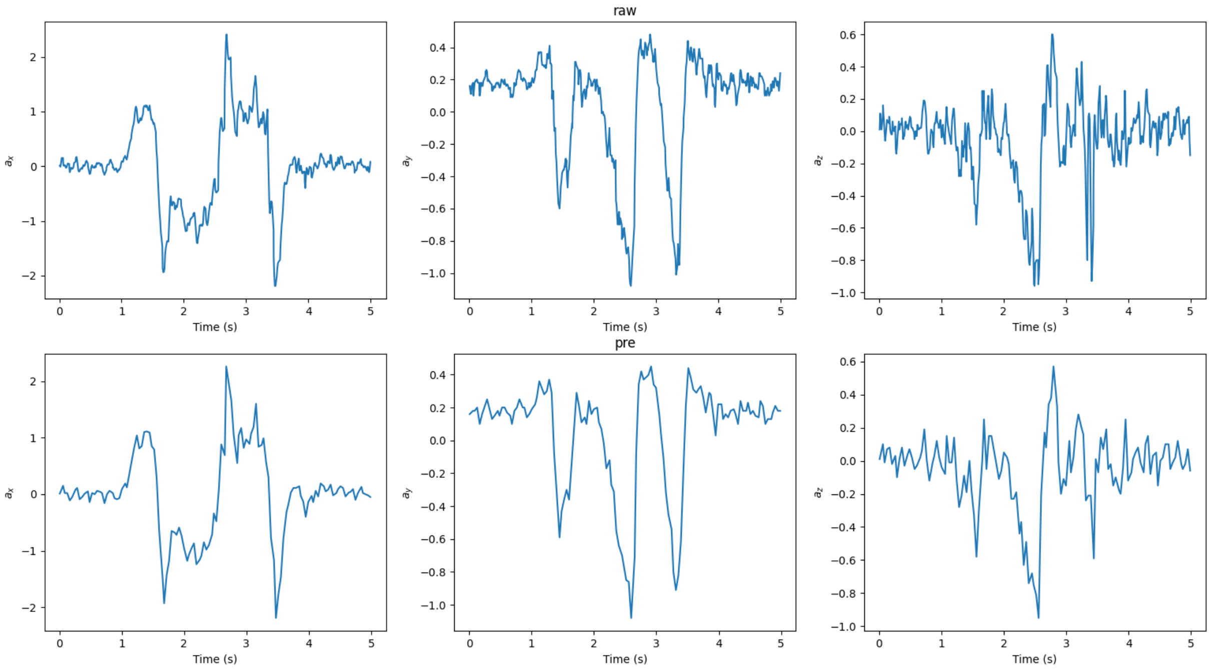


Figure 6. The four different resolutions evaluated during a complete run of the FastDTW algorithm.

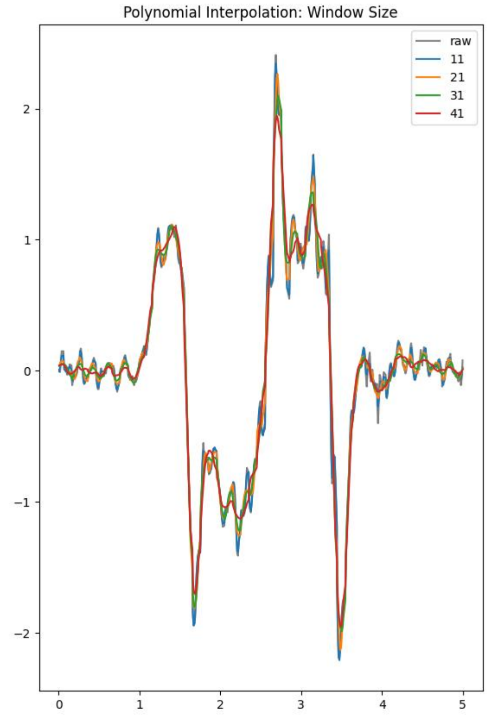
Improving Data

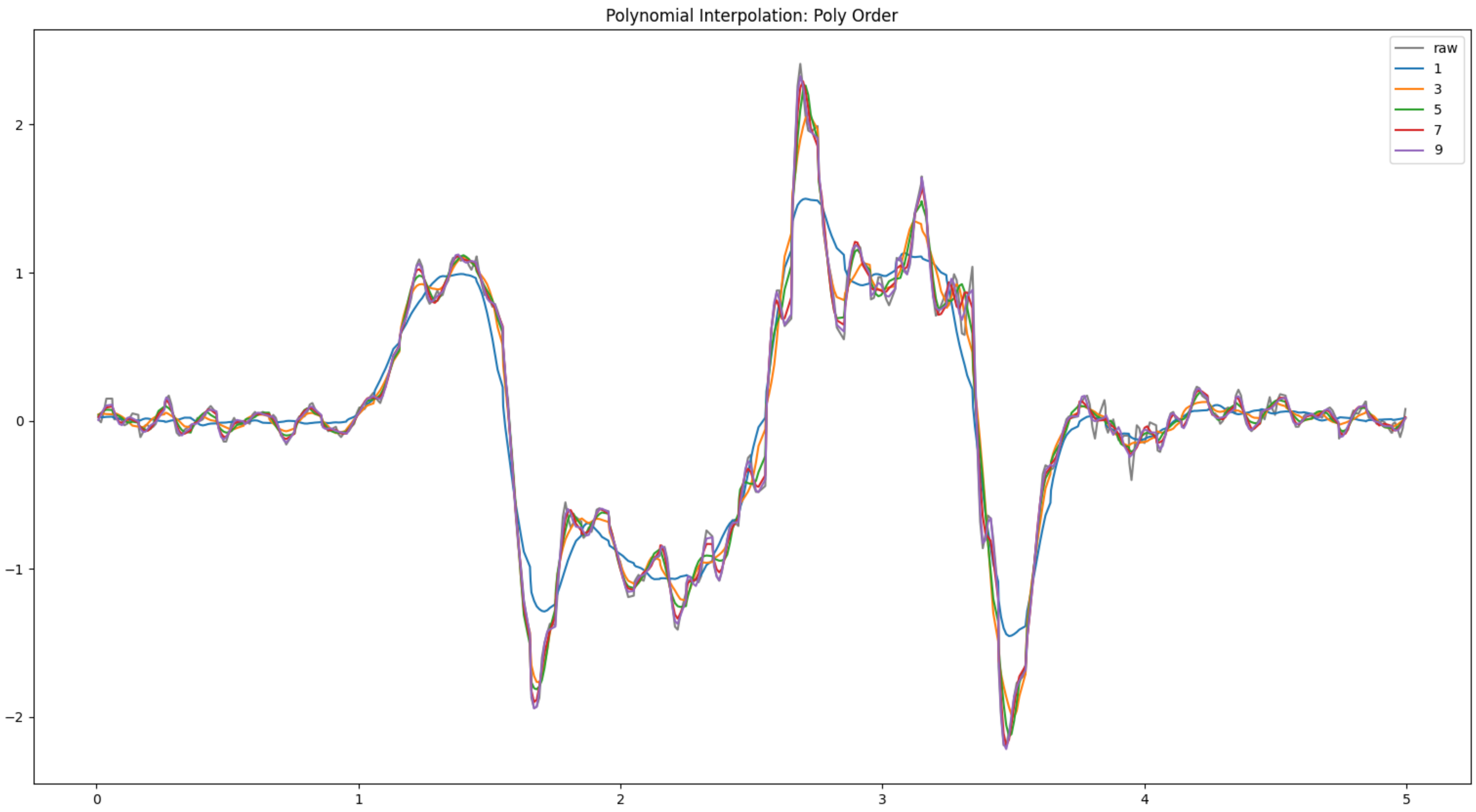
Downsampling



Polynomial Interpolation

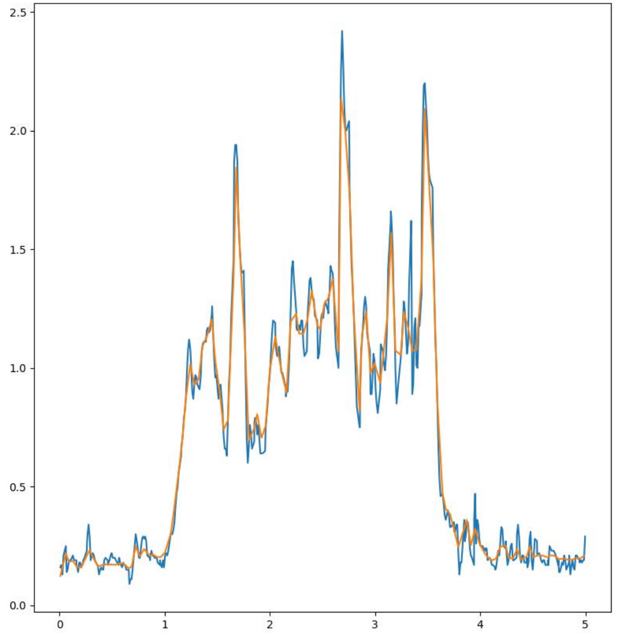
- Savitzky-Golay Filter
  - Essentially polynomial interpolation
- Fits successive subsets of adjacent data points with a low-degree polynomial by the method of linear least squares
  - Window and polynomial order must be tuned
- We apply moving average smoothing in addition to this




Polynomial Interpolation: Poly Order
raw
1
3
5
7
9
2
1
0
−1
−2
0
1
2
3
4
5

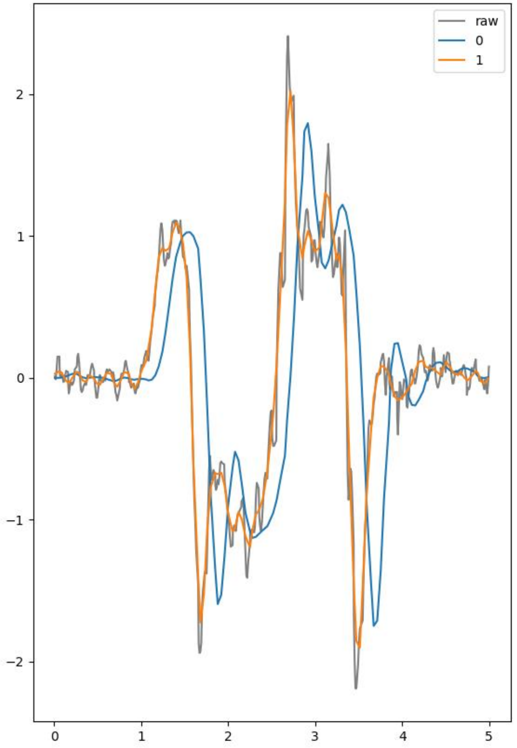
Wavelet Denoising

- Uses the Wavelet transform
  - STFT-like operation – convolve signal with shifted, scaled wavelets
- Better at preserving time-related features
- Relatively expensive to compute
- Denoising – take the transform, threshold coefficients and invert

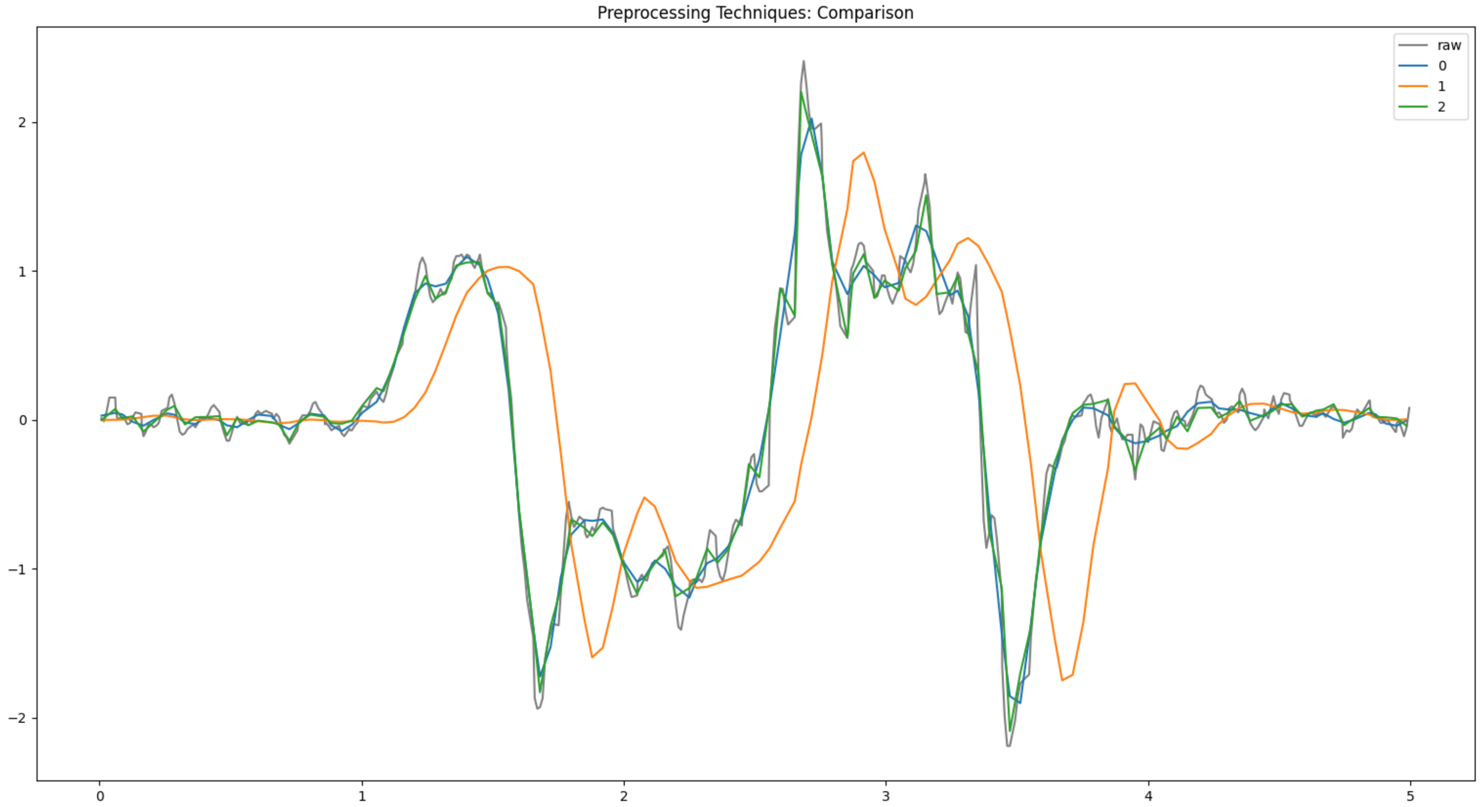


Low Pass Filtering

- Butterworth Filter
  - Very widely used for denoising in signal processing
  - Designed to have a flat response in the passband
  - Cutoff frequency and order can be controlled
- Used in low pass mode - high frequency noise attuned out
- Applying the filter inevitably causes a time shift
  - Increases with filter order



Comparisons



Runtime Comparison

Variant	Length of X	Length of Y	Average Runtime (s)
Vanilla	753	1154	7.99
Vanilla (DS)	188	289	1.45
Sakoe Chiba	753	1154	3.43
Sakoe Chiba (DS)	188	289	0.285
Itakura	753	1154	5.61
Itakura (DS)	188	289	0.387
FastDTW	753	1154	0.266
FastDTW (DS)	188	289	0.069

SoftDTW

Current DTW techniques ; discrete in nature (signal values as discrete times)

- What about continuous case? (Is this unrealistic?)
- ML and NN techniques prefer **differentiable** loss functions

Soft DTW computes value and gradient of this loss in quadratic time and quadratic space

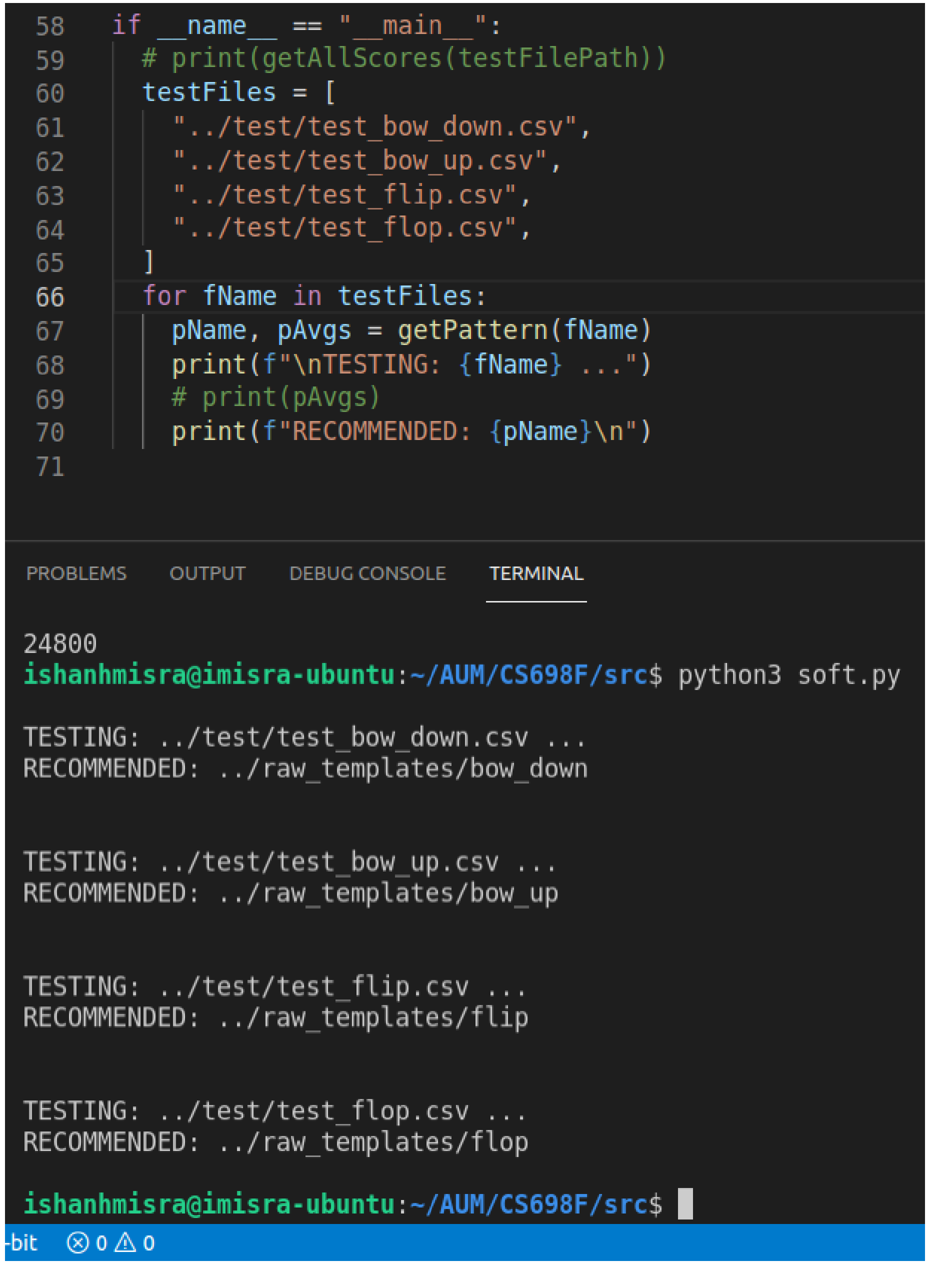
- Original paper: Soft-DTW: a Differentiable Loss Function for Time-Series (arxiv.org)

Code picked from https://github.com/Maghoumi/pytorch-softdtw-cuda as is (quite well-written)

- Offers CUDA (parallel programming library for NVIDIA GPUs) support

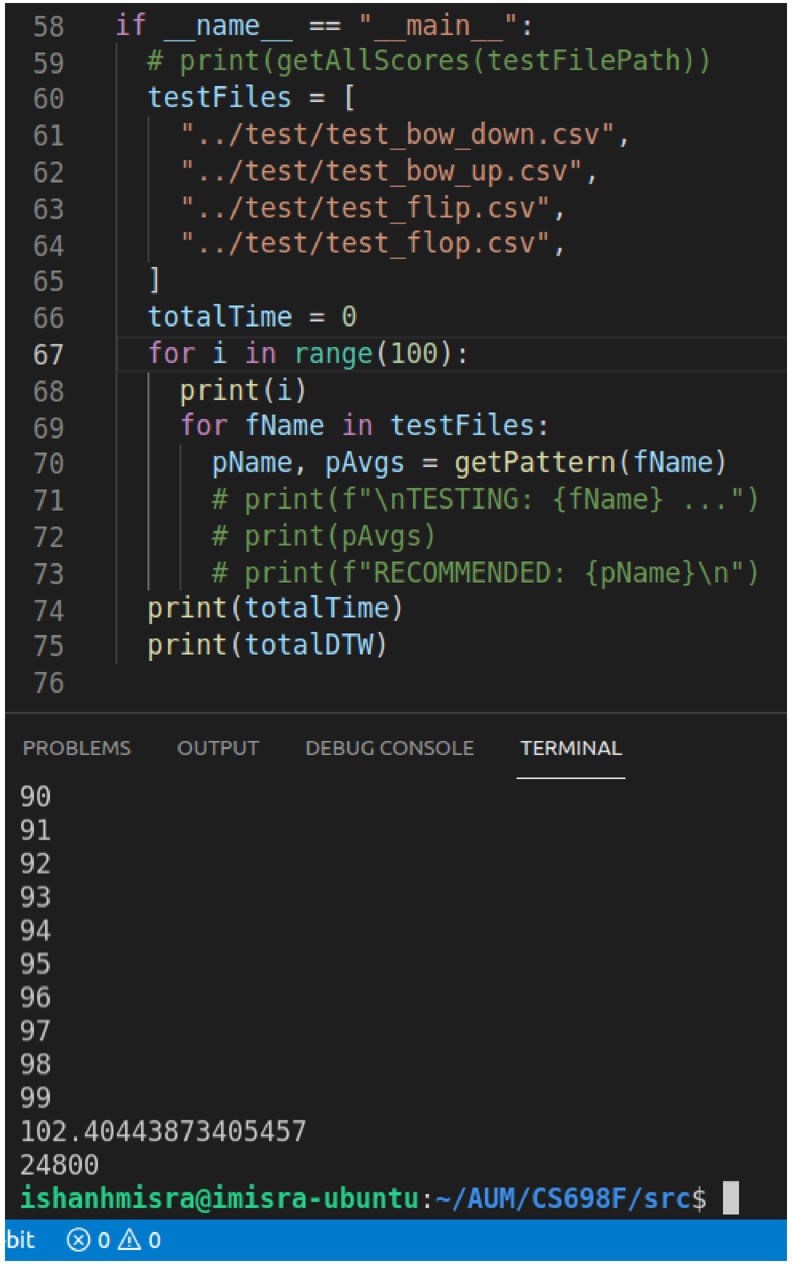
Tried computing SoftDTW for our project : remarkably fast code

- 24800 loss computations in 102.4 seconds
- Drawback: original data from other member of 3 dimensions, while softDTW tested on 6 dimensions
- Three extra dimensions for gyroscope (could not devote much time to compute bias)



```python
if __name__ == "__main__":
    # print(getAllScores(testFilePath))
    testFiles = [
        "../test/test_bow_down.csv",
        "../test/test_bow_up.csv",
        "../test/test_flip.csv",
        "../test/test_flop.csv",
    ]
    for fName in testFiles:
        pName, pAvgs = getPattern(fName)
        print(f"\nTESTING: {fName} ...")
        # print(pAvgs)
        print(f"RECOMMENDED: {pName}\n")

```

```
24800
ishanhmisra@imisra-ubuntu:~/AUM/CS698F/src$ python3 soft.py

TESTING: ../test/test_bow_down.csv ...
RECOMMENDED: ../raw_templates/bow_down


TESTING: ../test/test_bow_up.csv ...
RECOMMENDED: ../raw_templates/bow_up


TESTING: ../test/test_flip.csv ...
RECOMMENDED: ../raw_templates/flip


TESTING: ../test/test_flop.csv ...
RECOMMENDED: ../raw_templates/flop

ishanhmisra@imisra-ubuntu:~/AUM/CS698F/src$
```



```python
if __name__ == "__main__":
    # print(getAllScores(testFilePath))
    testFiles = [
        "../test/test_bow_down.csv",
        "../test/test_bow_up.csv",
        "../test/test_flip.csv",
        "../test/test_flop.csv",
    ]
    totalTime = 0
    for i in range(100):
        print(i)
        for fName in testFiles:
            pName, pAvgs = getPattern(fName)
            # print(f"\nTESTING: {fName} ...")
            # print(pAvgs)
            # print(f"RECOMMENDED: {pName}\n")
    print(totalTime)
    print(totalDTW)

```

```
90
91
92
93
94
95
96
97
98
99
102.40443873405457
24800
ishanhmisra@imisra-ubuntu:~/AUM/CS698F/src$
```

DEMO:
Windowed DTW

Overall System Limitations

No single representative template for gestures – match with entire dictionary

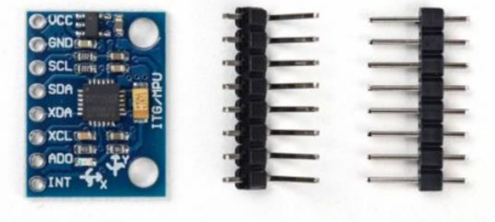
Not user agnostic : separately tested by respective members on their own devices

Tried developing a web application for (near) real time inference : some success

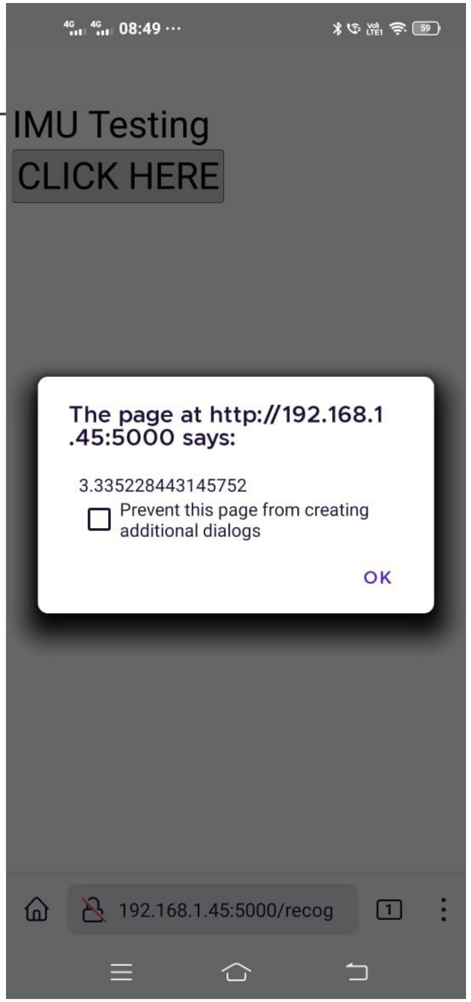
- Flask (Python3) server. Debugging problems on mobile caused limitations

Could have used actual hardware (standalone IMU chips)

- MPU-6050 (not very costly, off-the-shelf)



₹ 239.00



IMU Testing

The page at http://192.168.1.45:5000 says:

3.335228443145752

□ Prevent this page from creating additional dialogs

192.168.1.45:5000/recog