

ECE 276A - Project 1

Saqib Azim (A59010162)

sazim@ucsd.edu

UC San Diego

Feb 10, 2023

1 Introduction

In this project, we are given a setup consisting of an IMU sensor and a camera jointly mounted together such that they form a rigid body. The joint setup is able to rotate about its axes and we can measure the linear acceleration and rotational velocity from the readings obtained from the accelerometer and gyroscope. The IMU sensor is configured with a MEMS-based ADXL335 accelerometer and an LPR530AL gyroscope motion sensor. The raw IMU measurements consist of noise due to various factors such as measurement error, environmental conditions, experimental error, etc. There is also bias present in the raw acceleration and rotational velocity data due to the device configuration. We have preprocessed the raw data to mitigate existing bias and to convert the data into acceleration (in m/s^2) and rotational velocity (in rad/s). Using this, we can estimate the positional trajectory as well as the rotational trajectory of the body.

In addition to IMU measurements, the setup also captures images using the attached camera

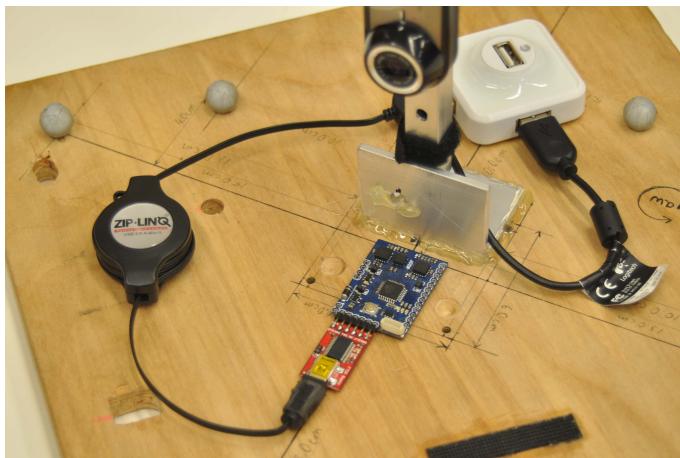


Figure 1: IMU and Camera Setup

module. The precise and accurate orientation of the body is also measured using a VICON motion capture system. The measurement rates of the camera, the IMU sensor, and the VICON system are different. The setup of IMU and camera can be seen in figure 1.

In the first part of the project, we are required to estimate the 3D orientation of the rotating body using only the IMU measurements (angular velocity ω_t and linear acceleration a_t). First, we estimated the bias in the measured raw IMU data by computing the mean of initial samples as we observe that the body is nearly at rest initially for a few seconds. We also observe this behavior by plotting the rotational euler angles (roll, pitch, yaw) measured using the VICON system. We convert the IMU data (measured in millivolts) to its corresponding SI units by multiplying a scale factor computed based on the reference voltage (V_{ref}) and the sensitivity of the accelerometer and gyroscope used. To track the orientation of the body over time, we use the quaternion kinematics motion model to predict the body orientation and the observation model to predict the linear acceleration at each time step. We estimate orientation trajectory by optimizing a cost function over all quaternion orientation variables that tries to minimize the error between the predicted orientation and orientation variables. Finally, we present results comparing our estimated orientation trajectory with the ground-truth orientation from VICON data.

In the second part of this project, we are required to use camera data to create a panorama image by stitching together RGB frames using body orientations $q_{1:T}$. To stitch together multiple overlapping images, we need to project them onto a common space and we have used a cylindrical space for this. We first overlay the image on a sphere and compute the spherical coordinates of each pixel. Using spherical coordinates, we convert it to cartesian coordinates which gives the position of each pixel in the 3D body frame. Now, we convert these pixel locations from the body frame to the world frame using the appropriate rotation matrix obtained from the VICON ground-truth measurements. Then, we reversed the rotated cartesian coordinates back to the spherical coordinates. In order to project these rotated pixel locations onto a common image, we project these pixel locations to a cylinder by imagining an image wrapped around the cylinder of height π and width 2π . Finally, we unwrap the cylindrical image and assign the RGB intensity values to the corresponding pixels in stitched image. Further details are explained in the technical approach section of this report.

2 Problem Formulation

2.1 Orientation Tracking

In the first part of the project, we are required to track the 3D orientation of the body and estimate the orientations (in terms of rotation matrices or quaternions) that the body undergoes for each time step. We measured the raw IMU data for linear acceleration $\{A_t\}_{t=0}^T$ and rotational velocity $\{\Omega_t\}_{t=0}^T$ in terms of voltages and convert it to linear acceleration $\{a_t\}_{t=0}^T$ and rotation velocity $\{\omega_t\}_{t=0}^T$ in SI units. Using this data, we estimate the orientation estimates q_1, q_2, \dots, q_T given the initial orientation of the body $q_0 = [1, 0, 0, 0]^T$. We use the cost function $c(q_1, q_2, \dots, q_T)$ as defined below:

$$c(q_1, q_2, \dots, q_T) = \frac{1}{2} \left[\sum_{t=0}^{T-1} \|2 \log(q_{t+1}^{-1} \circ f(q_t, \tau_t, \omega_t))\|_2^2 + \sum_{t=1}^T \|a_t - h(q_t)\|_2^2 \right] \quad (1)$$

$\tau_t = t_{k+1} - t_k$ represents the time difference between consecutive samples measured by IMU sensor.

q_t is the quaternion representing the orientation of the body wrt the world frame at time t

$f(q_t, \tau_t, \omega_t) = q_t \circ \exp([0, \tau_t \omega_t / 2])$ is the quaternion kinematics motion model that predicts the orientation quaternion at next time step q_{t+1}

$h(q_t) = q_t^{-1} \circ [0, 0, 0, -g] \circ q_t$ denotes the observation model that predicts the linear acceleration of the body $\{a_t\}_{IMU}$ in IMU reference frame using the orientation q_t at each time step. Since, in this project, the body is undergoing pure rotation, the acceleration in the world frame $\{a_t\}_W$ should be constant at all times $= [0, 0, -g]$.

The 1st term in the cost function measures the motion model error based on relative rotation $q_{t+1}^{-1} \circ f(q_t, \tau_t, \omega_t)$ between the predicted orientation $f(q_t, \tau_t, \omega_t)$ and the estimated orientation q_{t+1} . The 2nd term in the cost function measures the observation model error between the measured linear acceleration $\{a_t\}_{t=1}^T$ and predicted acceleration $h(q_t)$.

We estimate the orientation trajectory by minimizing the cost function wrt variables $q_{1:T} = \{q_1, q_2, \dots, q_T\}$. We also enforce the constraint that the orientation quaternions $\|q_t\|_2 = 1$ for all $t \in \{1, 2, \dots, T\}$ to ensure that they represent rotations in 3D space.

We state our optimization problem as:

$$\min_{q_1, q_2, \dots, q_T} c(q_{1:T}) \quad \text{s.t.} \quad \|q_t\|_2 = 1 \quad \forall \quad t \in 1, 2, \dots, T \quad (2)$$

2.2 Panorama Stitching

Given orientation estimates of the IMU $q_{1:T}$, and a sequence of RGB camera images $c_{1:N}$, sampled or captured at different frequencies, we are required to stitch the given sequence of images together into a single panoramic image.

3 Technical Approach

3.1 Orientation Tracking

- In my approach, the body frame coordinates convention is taken as shown in figure 2

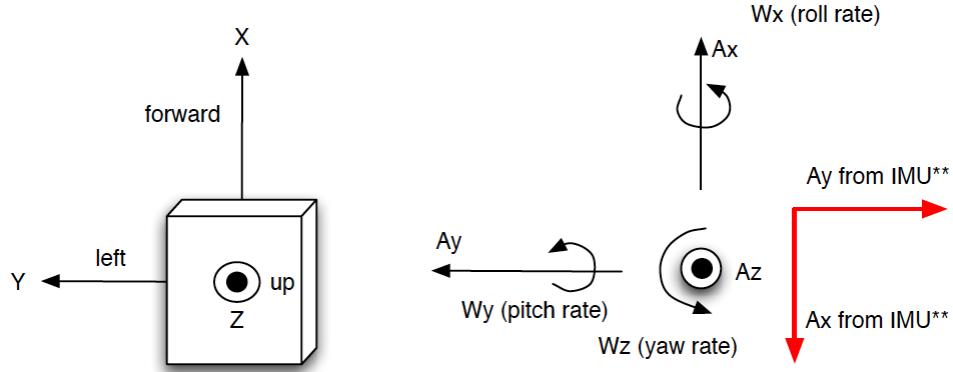


Figure 2: IMU and Camera Setup

- The IMU sensor measures linear acceleration $\{A_t\}_{t=1}^T$ and rotational angular velocity $\{\Omega_t\}_{t=1}^T$ in terms of voltages. We first convert from raw data to data in SI units, namely $\{a_t\}_{t=1}^T$ and $\{\omega_t\}_{t=1}^T$. We compute bias present in the IMU data for A_t and Ω_t by averaging initial raw samples $t = 1 : T_r$ (when the body is at rest).

$$A_{bias} = \frac{1}{T_r} \sum_{t=1}^{T_r} A_t \quad \text{and} \quad \Omega_{bias} = \frac{1}{T_r} \sum_{t=1}^{T_r} \Omega_t \quad (3)$$

where $A_{bias} \in \mathbb{R}^3$ and $\Omega_{bias} \in \mathbb{R}^3$

- T_r is estimated by plotting raw data and observing the time from 0 to T_r upto which the values are approximately constant. $T_r = 500$ initial samples (for dataset = 1,2,3,4,8,9), 400 initial samples (for dataset 6,7,10), and 350 initial samples (for dataset 5, 11).
- For the acceleration measured in the z-direction ($A_t)_z$, the body experiences gravity in the world frame and therefore, the body's acceleration is non-zero but equal to $-g$ or 9.8 m/s^2 in the vertical downward (negative z) direction. The initial sample mean calculated for the z-component of acceleration $(A_{bias})_z$ has a contribution due to actual system bias $(A_{bias})_{sys}$ and a contribution due to gravity $(A_{bias})_g$.

$$(A_{bias})_z = (A_{bias})_g + (A_{bias})_{sys} \quad (4)$$

To compute the actual system bias in the z-direction of accelerometer, we compute $(A_{bias})_g = \frac{g}{S_a}$. Therefore,

$$(A_{bias})_{sys} = (A_{bias})_z - (A_{bias})_g = (A_{bias})_z - \frac{g}{S_a} \quad (5)$$

$$(a_t)_{x,y} = [(A_t)_{x,y} - (A_{bias})_{x,y}] * S_a \quad \text{and} \quad \omega_t = [\Omega_t - \Omega_{bias}] * S_\omega. \quad (6)$$

$$(a_t)_z = [(A_t)_z - (A_{bias})_{sys}] * S_a \quad (7)$$

where S_a and S_ω are scale factors for the IMU accelerometer and gyroscope and are computed as follows:

$$S_a = \frac{V_{ref}}{1023 * \text{sensitivity}_{acclm}} \quad \text{and} \quad S_\omega = \frac{V_{ref}}{1023 \times \text{sensitivity}_{gyro}} \quad (8)$$

$$V_{ref} = 3300 \text{ mV} = 3.3 \text{ V}$$

$$\text{sensitivity}_{acclm} = 300 \text{ mV/g}$$

$$\text{sensitivity}_{gyro} = 3.33 \text{ mV/degree/s} = 190.8 \text{ mV/rad/s.}$$

- In order to verify the correctness of calibrated IMU data, we used motion model to compute $q_{t+1} = f(q_t, \tau_t, \omega_t)$ given initial body orientation $q_0 = [1, 0, 0, 0]$. The motion model performs simple integration of the angular velocity to compute body orientations at successive time steps. We then converted this to roll, pitch and yaw euler angles and compared with the ground-truth data from VICON. For each dataset (from 1 to 9), the comparison plot has been shown below in the results section.
- For optimization implementation, we have used the python autograd library that computes numerical gradients for functions. We tried using the JAX library but faced error since JAX does not allow modifying a JAX numpy variable after it is created. In autograd, we used the grad function to compute gradients wrt each quaternion.
- We used projected gradient descent (PGD) to optimize the orientation trajectory. In the update step, we compute the gradient of the cost function with respect to each quaternion variable and update the orientation quaternions for the entire trajectory $q_{1:T}$. In order to ensure that the updated quaternions represent valid rotation matrix, we normalize each quaternion variable to be of unit norm after each update step. The PGD step size has been kept constant at 0.005 until 250 iterations after which we decrease it to 0.001 for the remaining iterations.

3.2 Panorama Stitching

- First, we iterate over the sequence of camera images $c_{1:N}$ for each image, identify the nearest in the past timestamp at which an IMU sample (or a VICON sample) was captured.
- We used panorama image size to be $(H \times W) = 720 \times 1080$.
- Next, we project the image onto the surface of a unit-radius sphere centered at origin in 3D space. For convention, the x-axis points in forward direction and passes through the image center, y-axis to the left, and z-axis upwards. For spherical coordinate convention (λ, ϕ, r) , where longitude $\lambda \in [-\pi/2, \pi/2]$ and latitude $\phi \in [-\pi, \pi]$.
- Longitude λ is $-\pi/2$ at z-axis and increases to 0 at the x-y plane and further increases to $\pi/2$ at the negative-z axis. Latitude ϕ is positive in the anticlockwise direction and measured wrt positive-x axis.
- For each pixel coordinate (u, v) in the input image of size $(h \times w) = (240 \times 320)$, we estimate the spherical coordinates using equation 9. Here, we have used horizontal field of view = 60° and vertical field of view to be 45° .

$$\lambda = -\frac{(v - \frac{W}{2})}{w} * 60 \quad \text{and} \quad \phi = \frac{(u - \frac{H}{2})}{h} * 45 \quad (9)$$

- From the spherical coordinates, we convert each coordinate to cartesian coordinate using the following relation. For the chosen sphere, $r = 1$.

$$x = r \cos \phi \cos \lambda \quad y = r \cos \phi \sin \lambda \quad z = -r \sin \phi \quad (10)$$

- Now, for each pixel coordinate we have obtained their cartesian coordinate (x,y,z) in 3D space where the image is wrapped around the unit sphere. Next, we rotate these coordinates from the body frame to world frame using the body orientation obtained from IMU (or ground-truth orientation from VICON).

$$s_w = R s_b \quad (11)$$

- Now, we convert the cartesian coordinates (x,y,z) of each pixel back to spherical coordinates $(\lambda, \phi, 1)$ using the following relation

$$\phi = \arcsin\left(-\frac{z}{r}\right) \quad \lambda = \arctan\left(\frac{y}{x}\right) \quad (12)$$

- Next we can imagine the sphere inscribed inside a unit-radius cylinder. We can use the same spherical coordinates for the cylinder obtained above after rotation to world frame. If we assume our image to be wrapped around the cylinder, then we can unwrap it and map each pixel into a rectangular canvas of size (720x1080) or (1080x1960).
- For each pixel (u, v) , we already know the transformed spherical coordinates in the world frame. We can use those to map to the panorama image coordinates using the following relation. Here $(H \times W) = (720 \times 1080)$.

$$U = \frac{\phi + 0.5\pi}{\pi} * H, \quad \text{and} \quad V = \frac{\pi - \lambda}{2\pi} * W \quad (13)$$

- Finally, for each pixel $I(u, v)$ in the original image, we know the corresponding pixel location in the panorama image $P(U, V)$. We copy the intensity values from $I(u, v)$ to $P(U, V)$.
- Because there is overlapping between consecutive images, multiple pixels in original image may map to same location in panorama image. In that case, one approach can be to take the average of overlapping pixels. For this assignment, I have simply overridden the new value.

4 Results: Orientation Tracking and Panorama Stitching

- For each training dataset, I have mainly plotted the raw IMU data (acceleration and angular velocity), the comparison between the ground-truth euler angles from VICON and estimated euler angles from the IMU sensor (before and after PGD optimization).
- For all train and test datasets, we can observe a clear decrease in the cost function. The cost function decreases rapidly initially and later converges as the optimization progresses. In most cases, the cost function converges to a small positive value.
- For datasets 1,2,4,5,6,7,8,9, we can clearly observe improvement in orientation trajectory before and after performing PGD optimization. In some datasets, such as 3,8, the improvement in trajectory is not clearly visible. This may be because the IMU data obtained in these trajectories was more accurate and hence similar to the vicon data trajectory.
- The optimized roll and pitch angles are very closer to the ground-truth roll and pitch angles. The yaw angles in general have more difference from their corresponding ground-truth yaw angles.
- In panorama stitching, for each train dataset, I have shown two stitched images. One using VICON orientation data for rotating images and another using optimized IMU orientation trajectory $q_{0:T}$.
- In panorama stitching, for test dataset, I have shown only one stitched image using the optimized orientation trajectory since the VICON data is unavailable.
- In panorama stitching, the stitched image from VICON is slightly better as compared to stitched image generated by IMU orientations.

4.1 Dataset-1

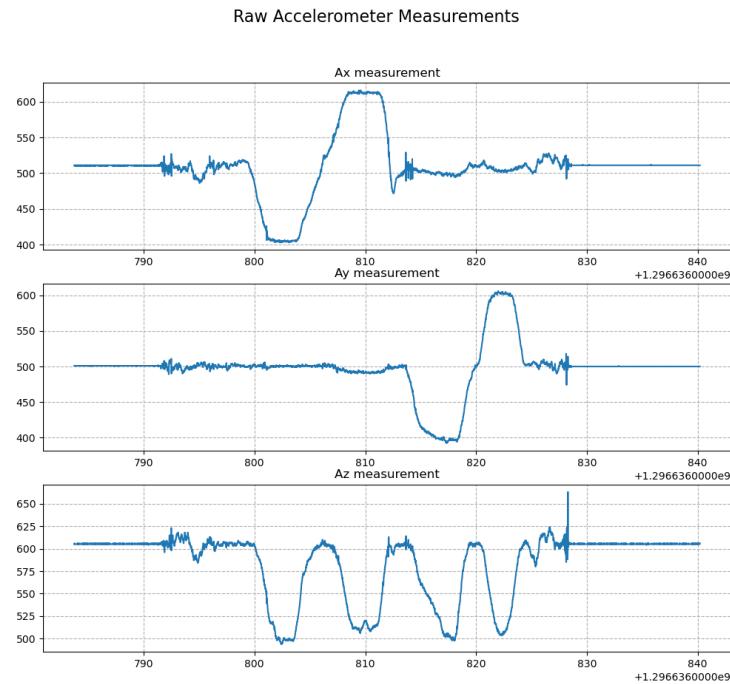


Figure 3: Raw Acceleration data

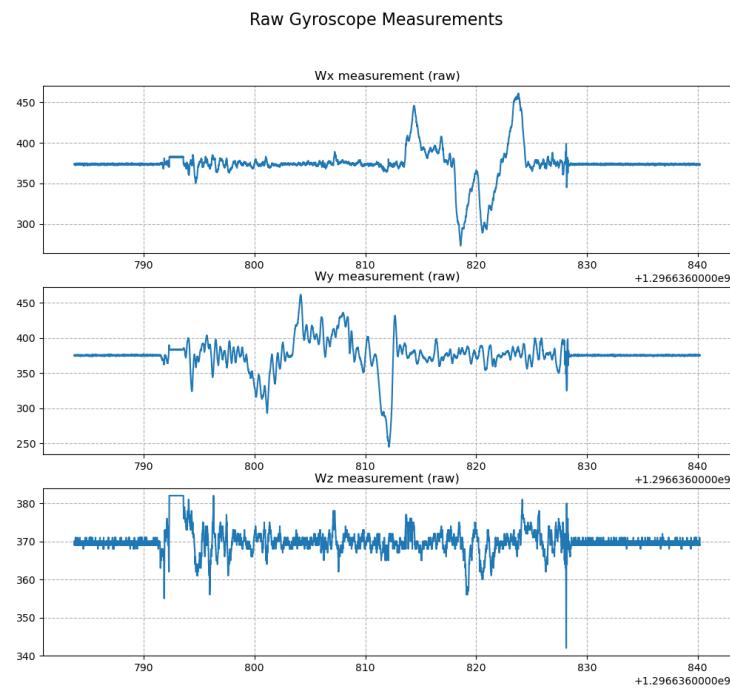


Figure 4: Raw Gyroscope data

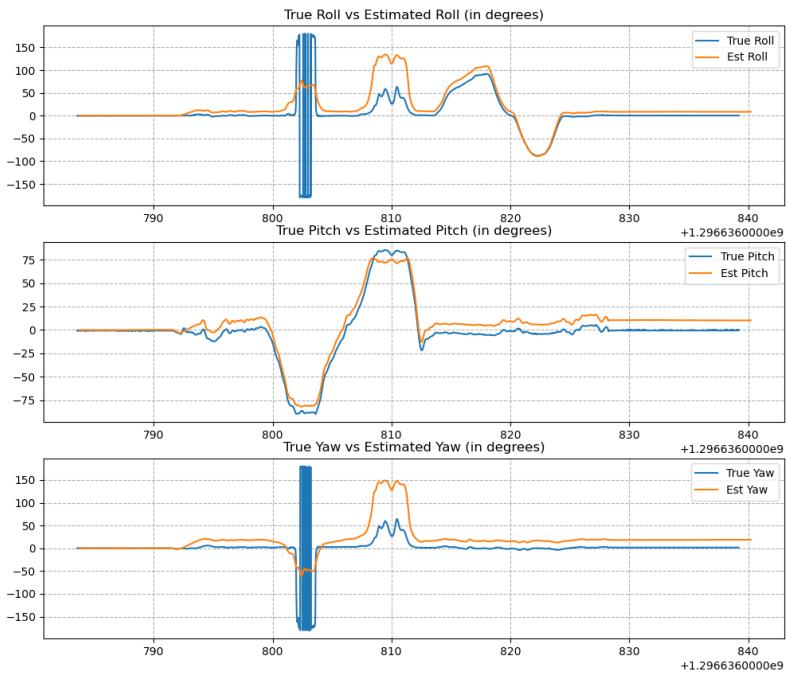


Figure 5: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

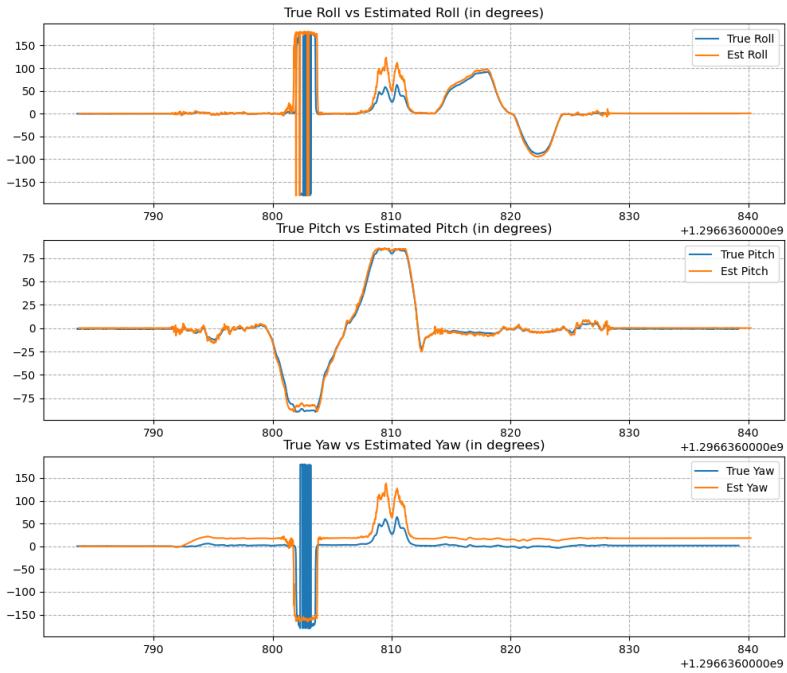


Figure 6: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)



Figure 7: Panorama Image (720 x 1280) (using VICON orientation)

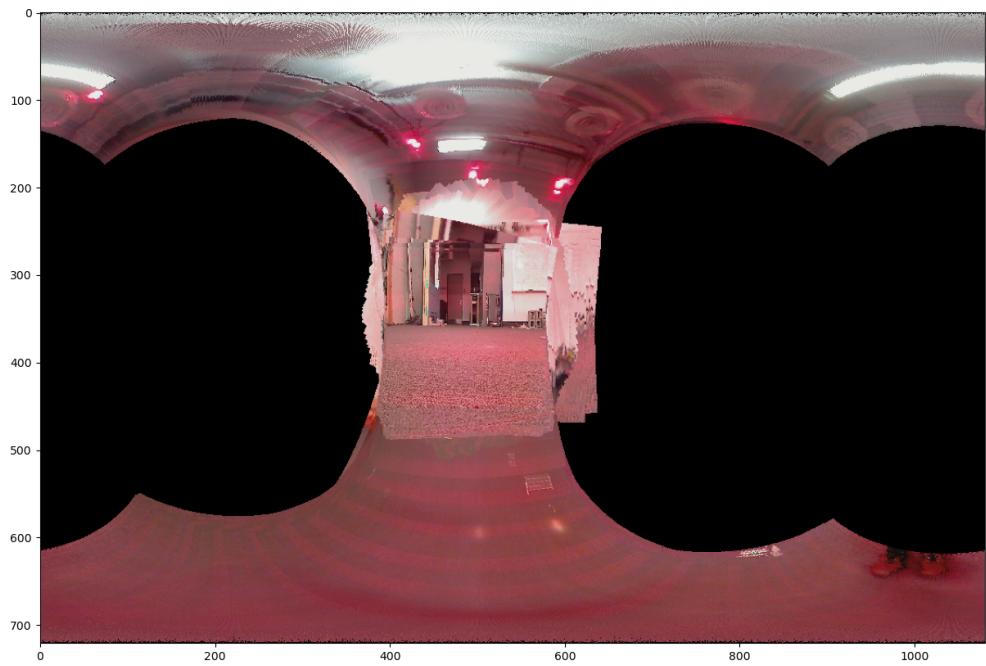


Figure 8: Panorama Image (720 x 1280) (using optimized IMU orientation)

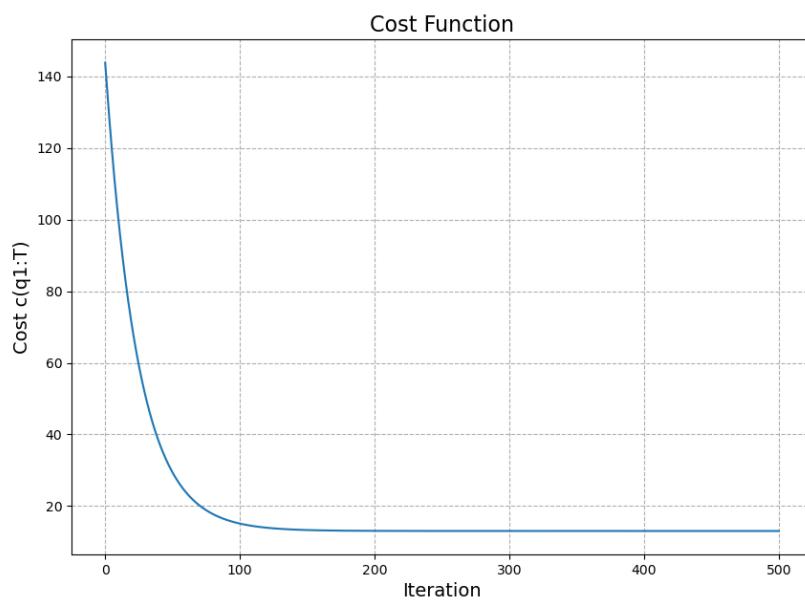


Figure 9: PGD Cost function variation

4.2 Dataset-2

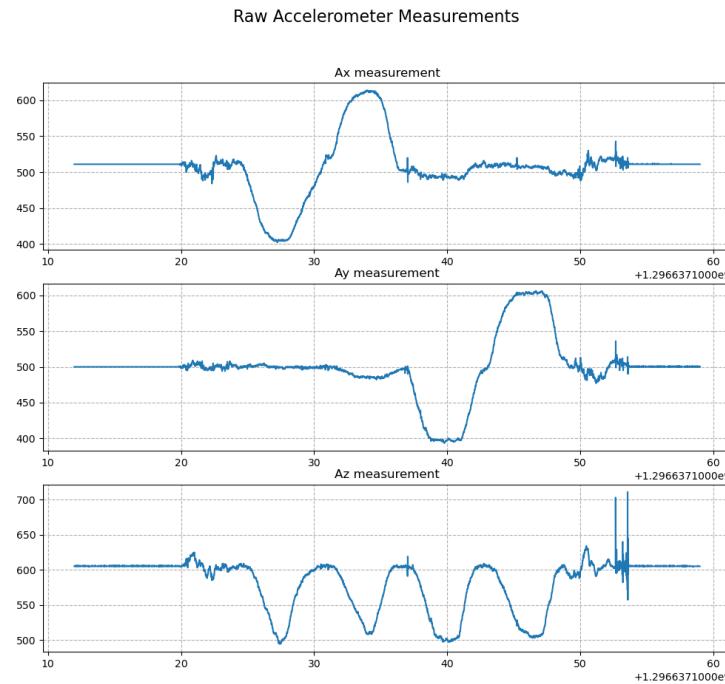


Figure 10: Raw Acceleration data

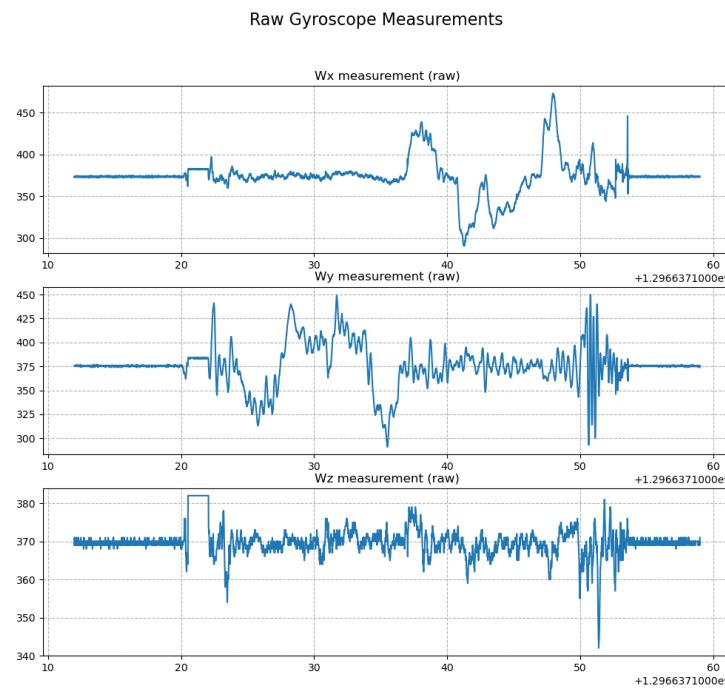


Figure 11: Raw Gyroscope data

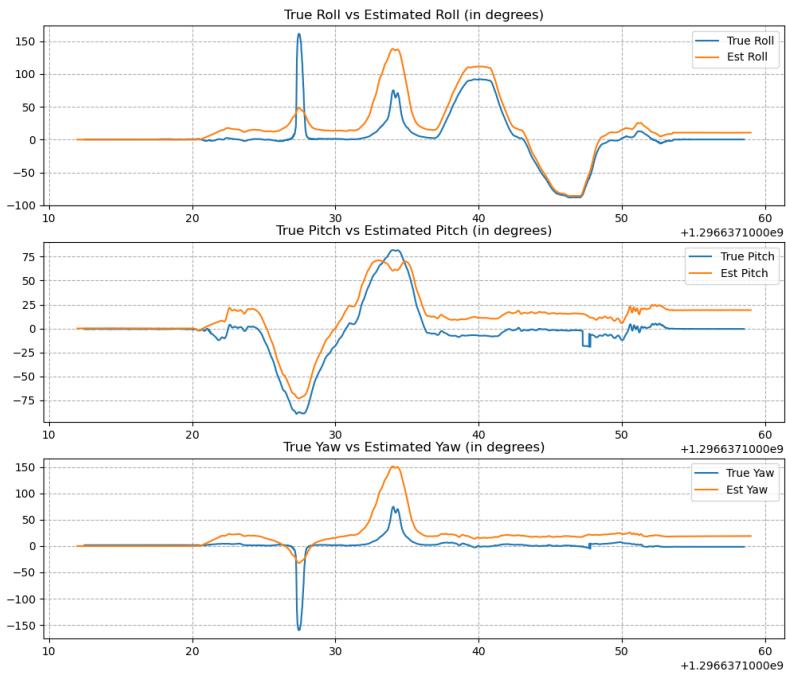


Figure 12: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

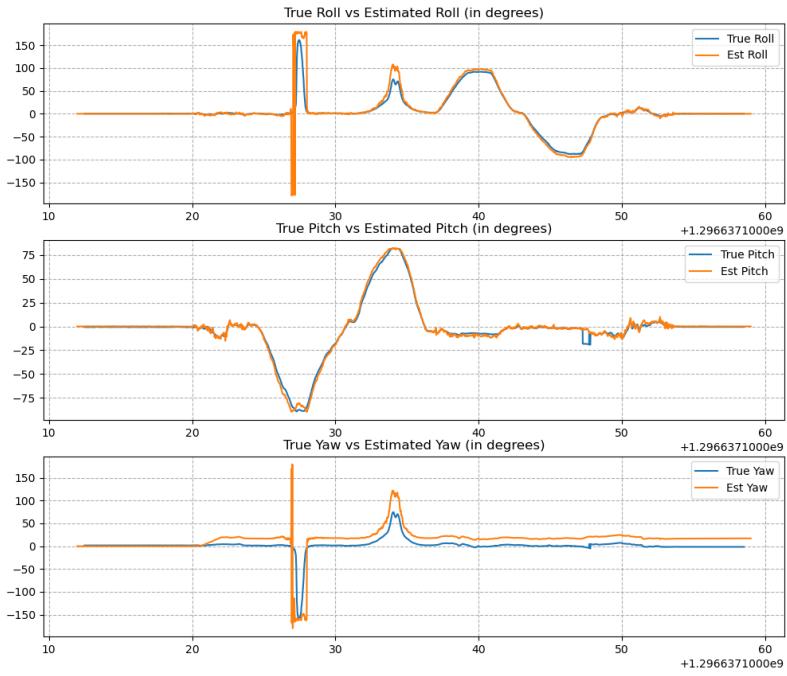


Figure 13: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

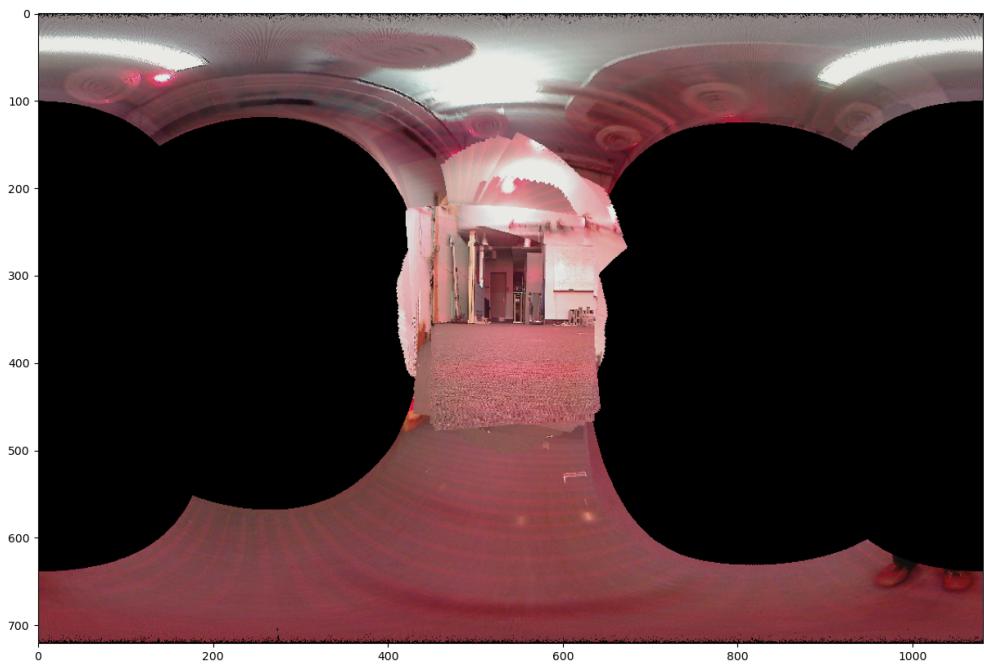


Figure 14: Panorama Image (720 x 1280) (using VICON orientation)

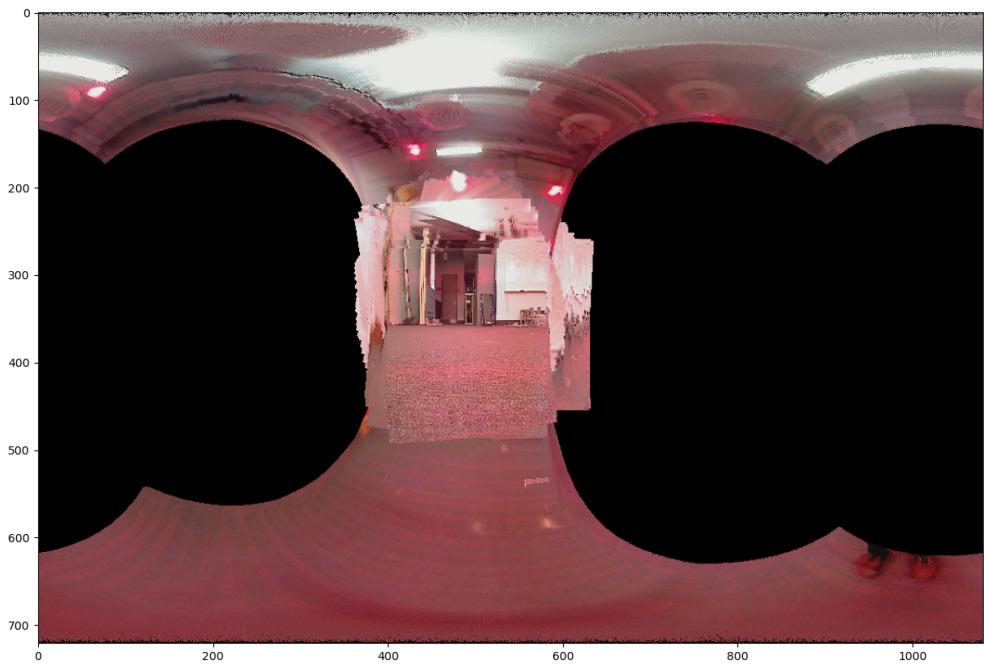


Figure 15: Panorama Image (720 x 1280) (using optimized IMU orientation)

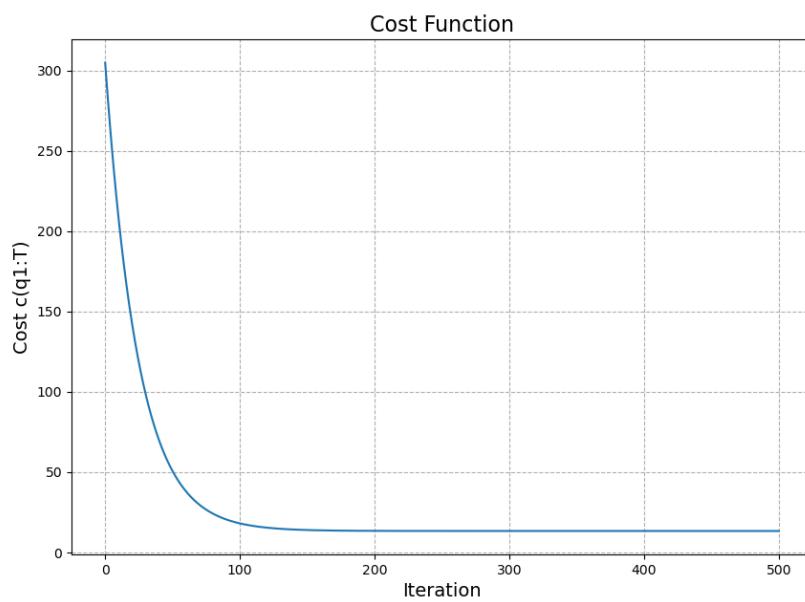


Figure 16: PGD Cost function variation

4.3 Dataset-3

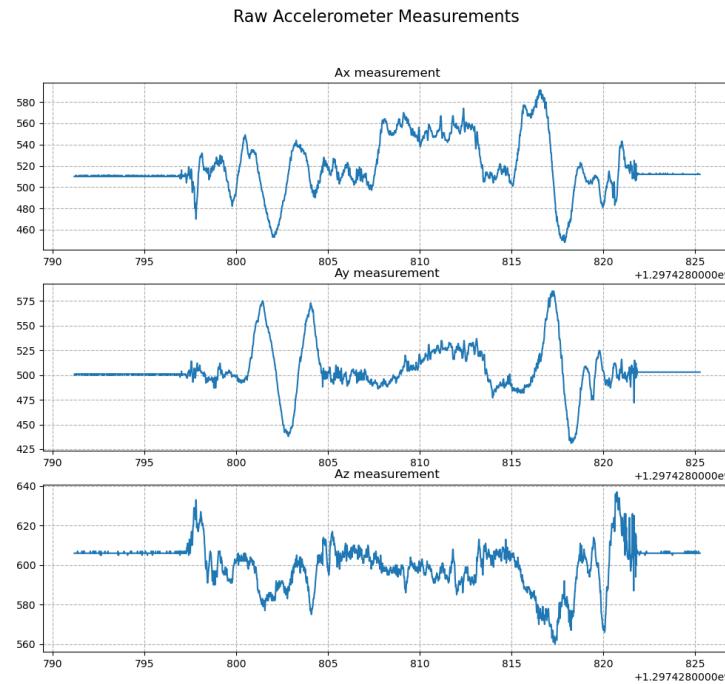


Figure 17: Raw Acceleration data

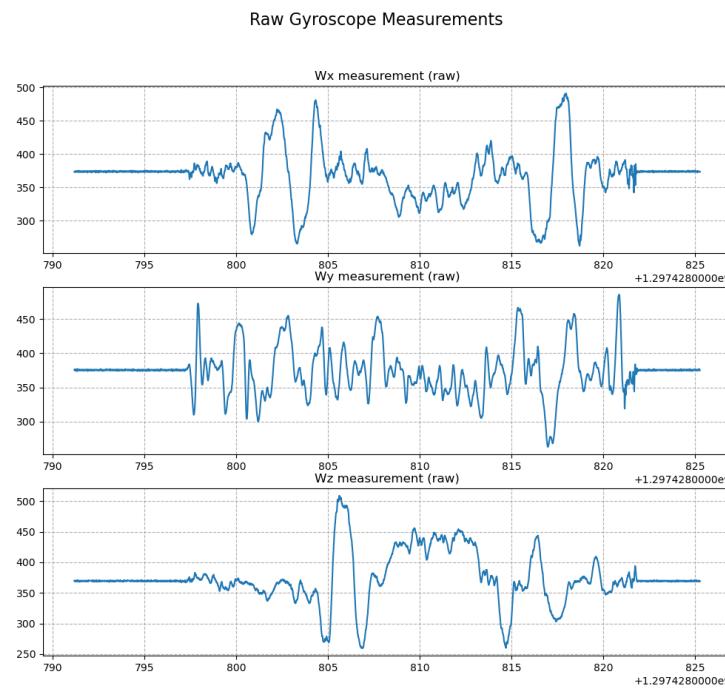


Figure 18: Raw Gyroscope data

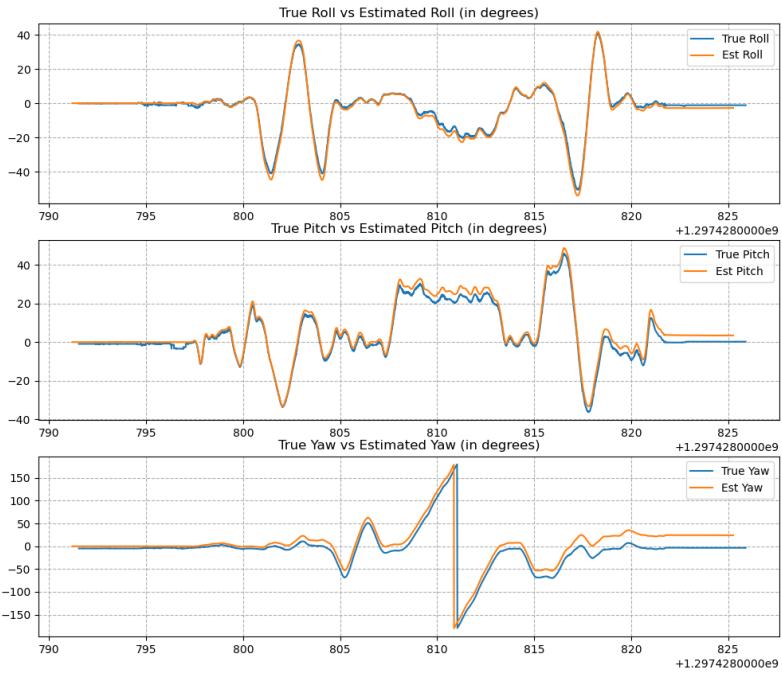


Figure 19: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

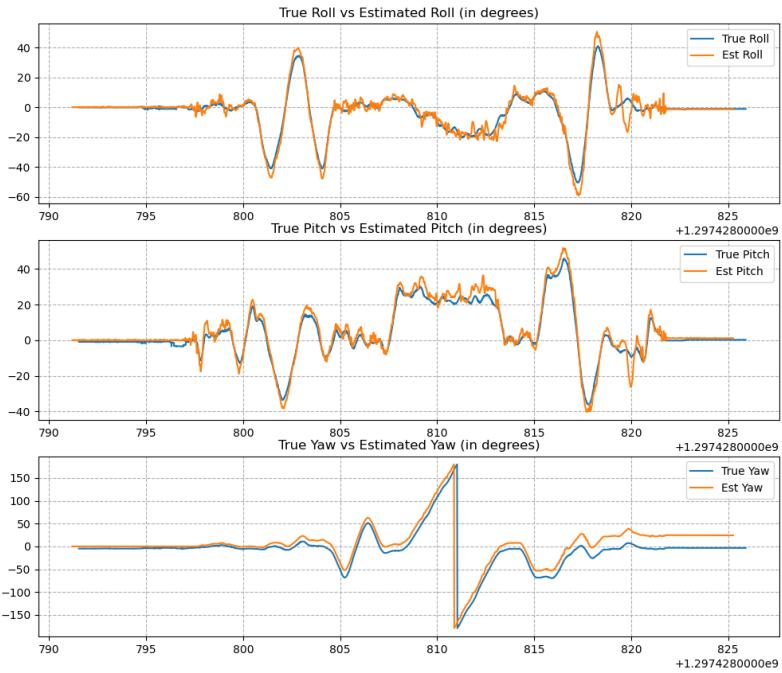


Figure 20: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

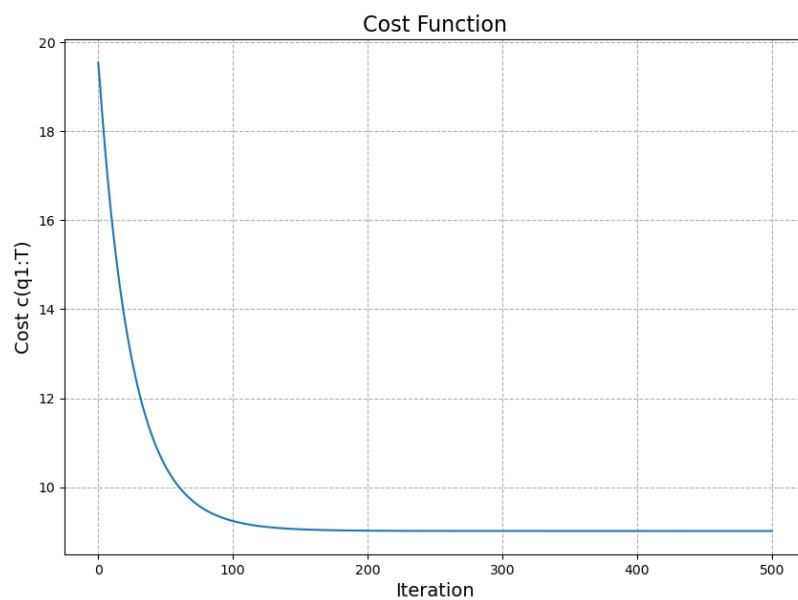


Figure 21: PGD Cost function variation

4.4 Dataset-4

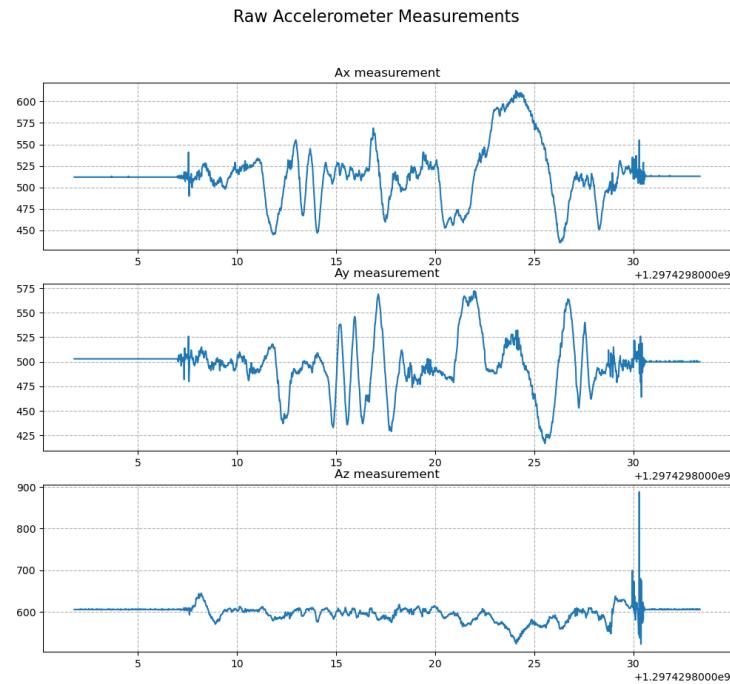


Figure 22: Raw Acceleration data



Figure 23: Raw Gyroscope data

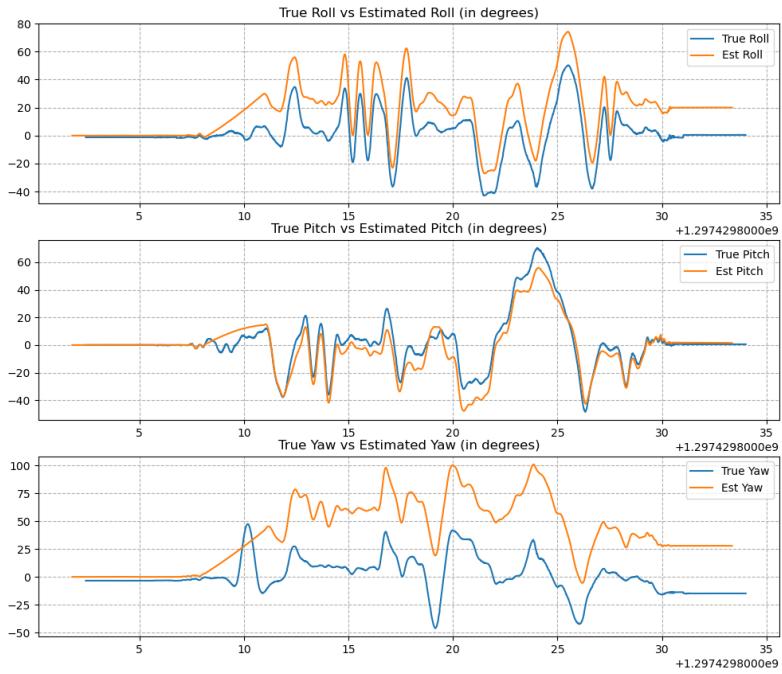


Figure 24: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

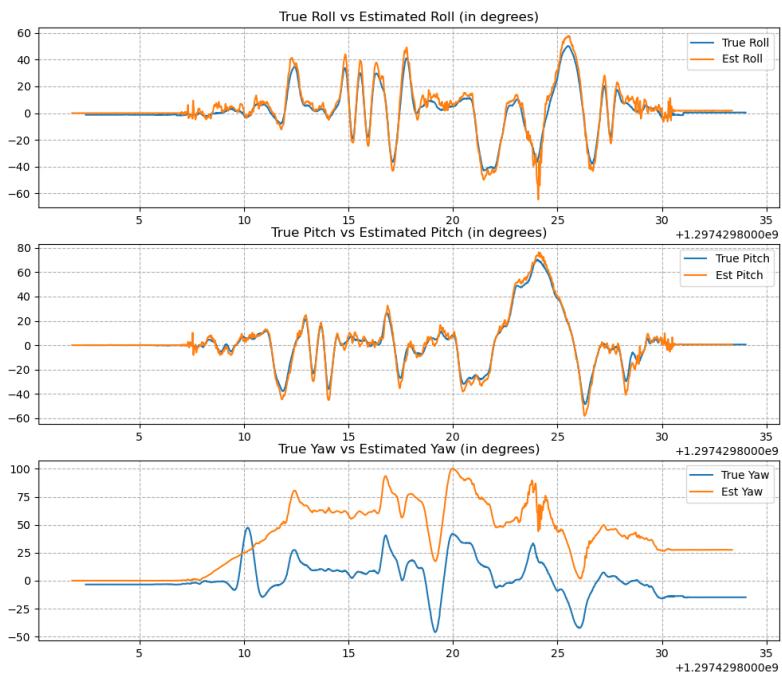


Figure 25: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

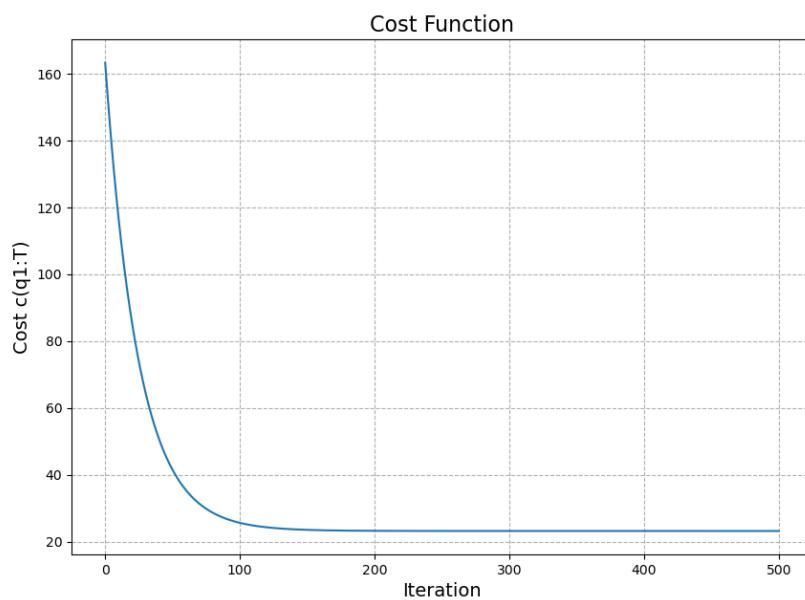


Figure 26: PGD Cost function variation

4.5 Dataset-5

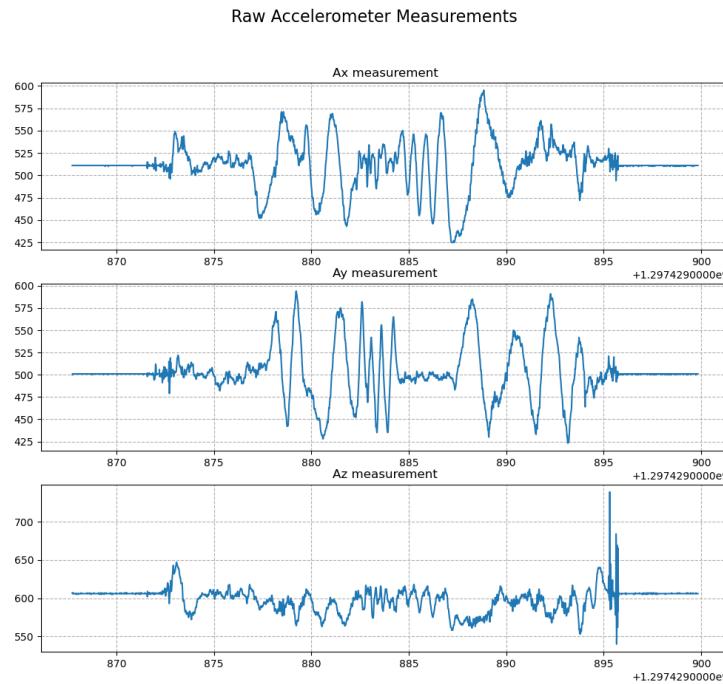


Figure 27: Raw Acceleration data



Figure 28: Raw Gyroscope data

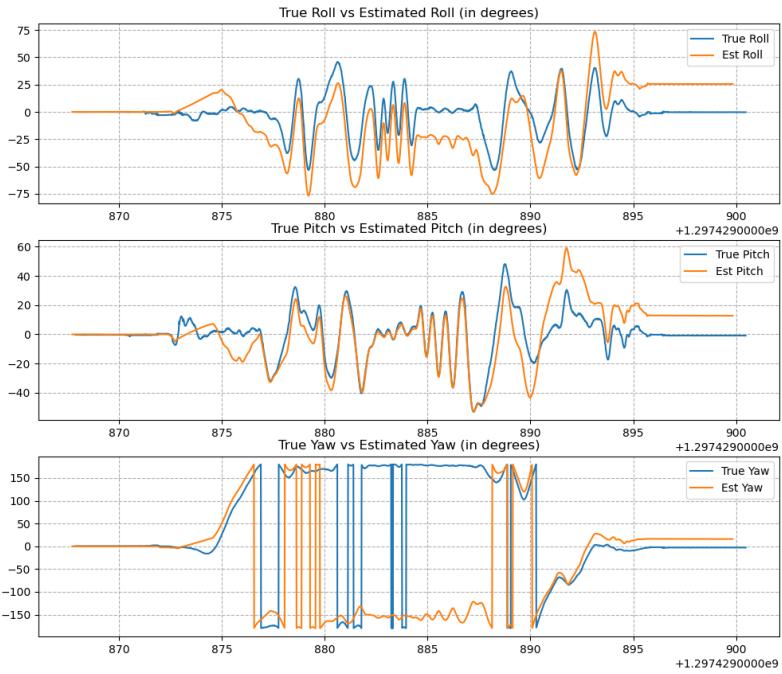


Figure 29: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

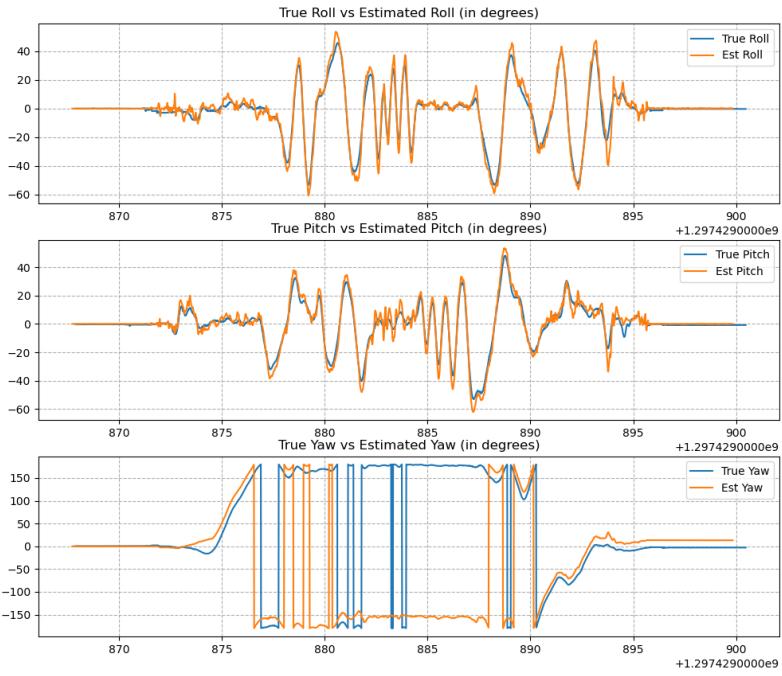


Figure 30: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

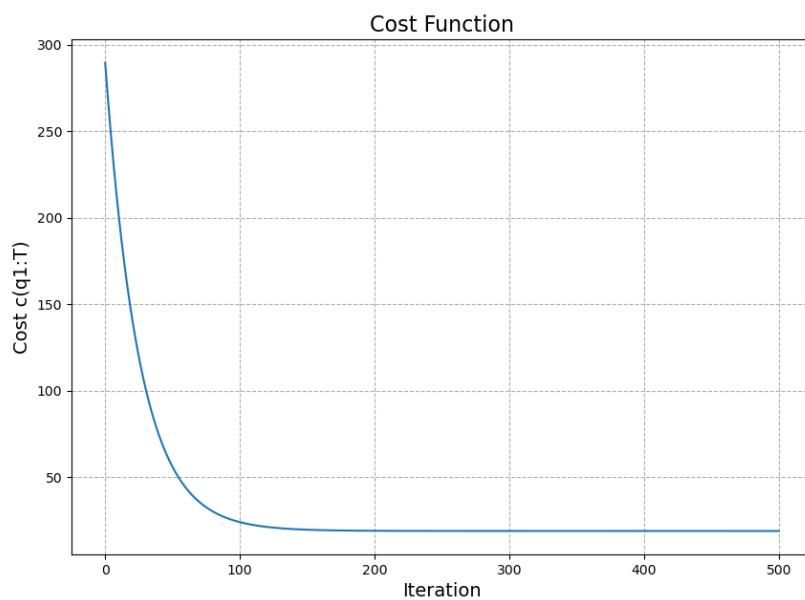


Figure 31: PGD Cost function variation

4.6 Dataset-6

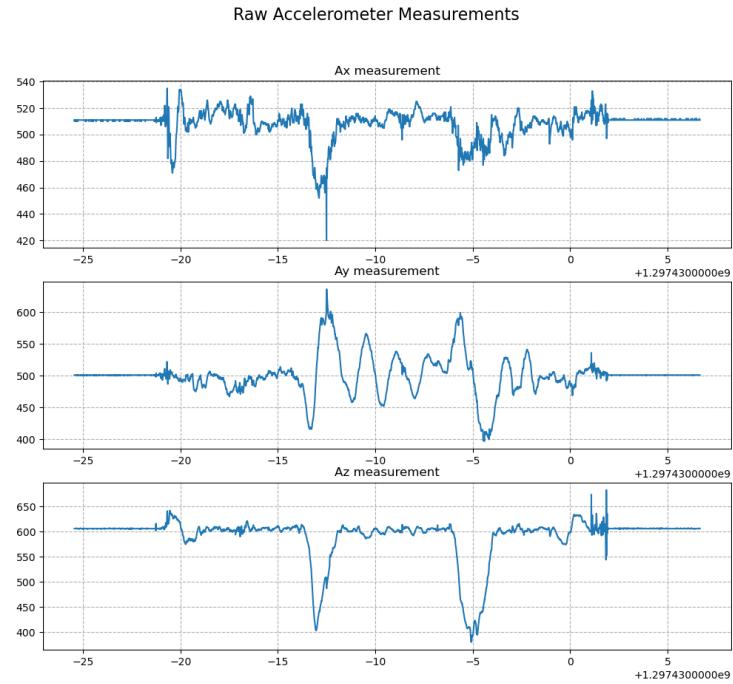


Figure 32: Raw Acceleration data

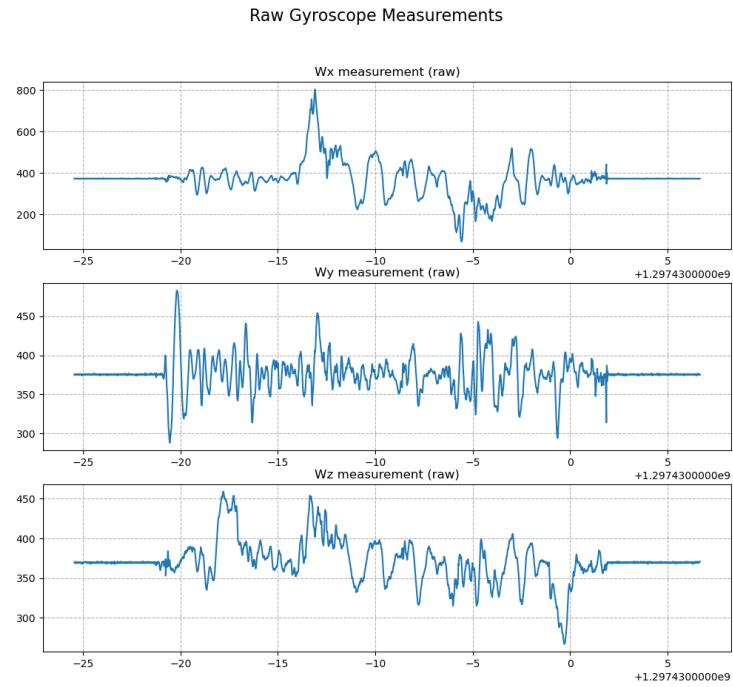


Figure 33: Raw Gyroscope data

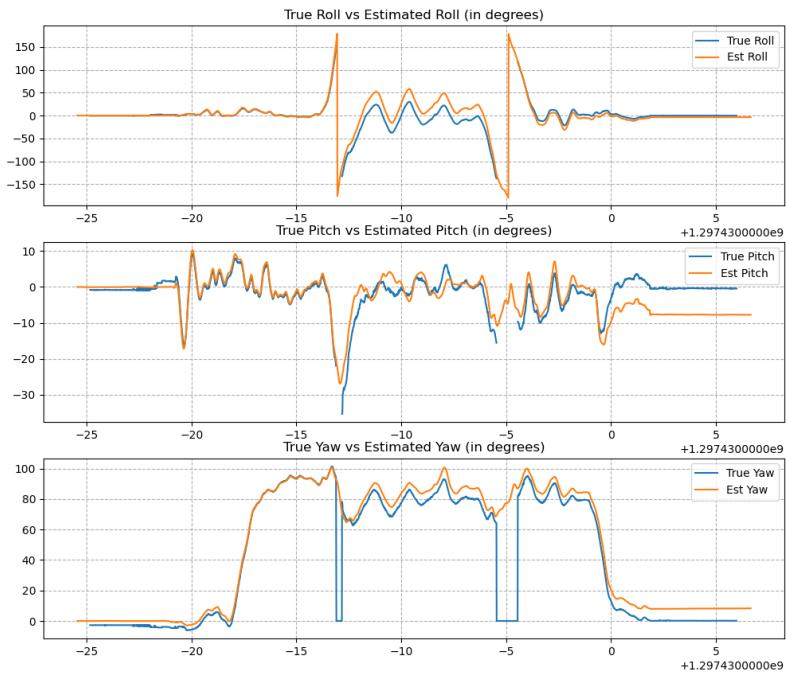


Figure 34: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

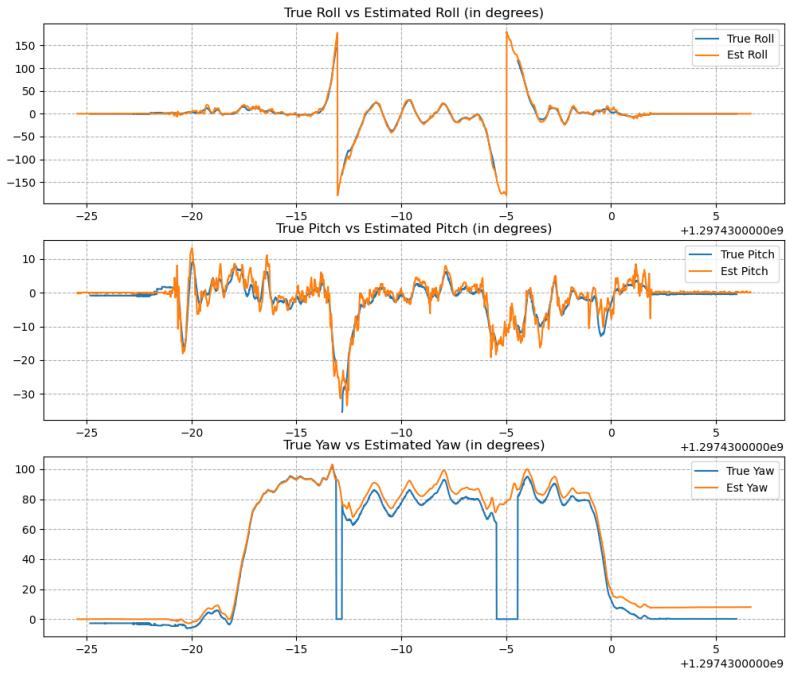


Figure 35: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

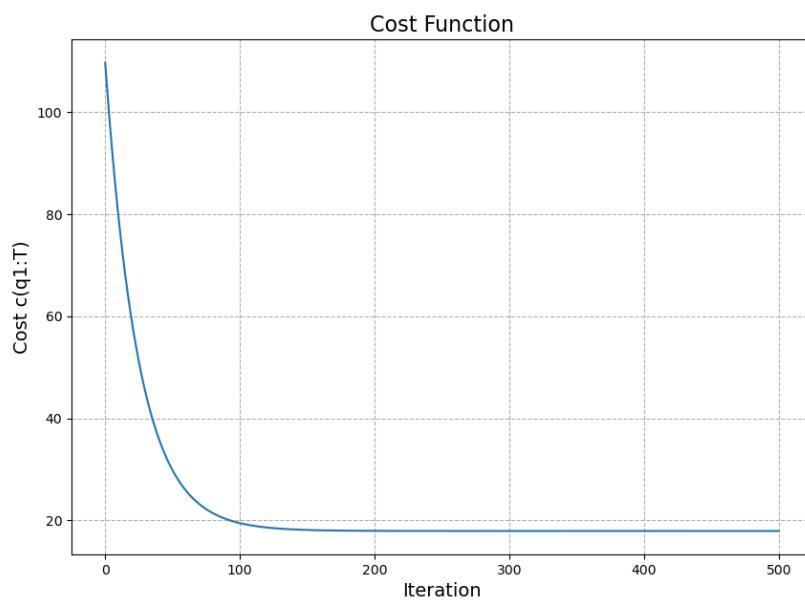


Figure 36: PGD Cost function variation

4.7 Dataset-7

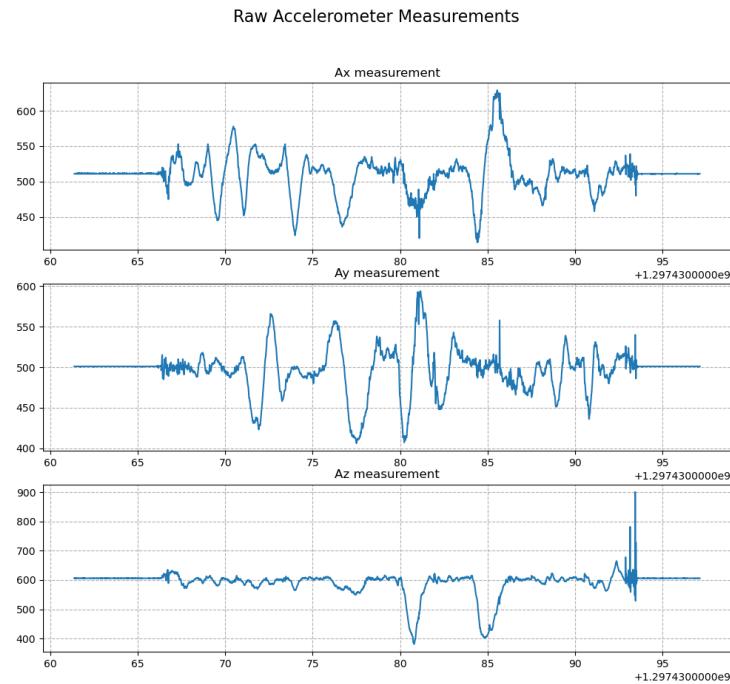


Figure 37: Raw Acceleration data



Figure 38: Raw Gyroscope data

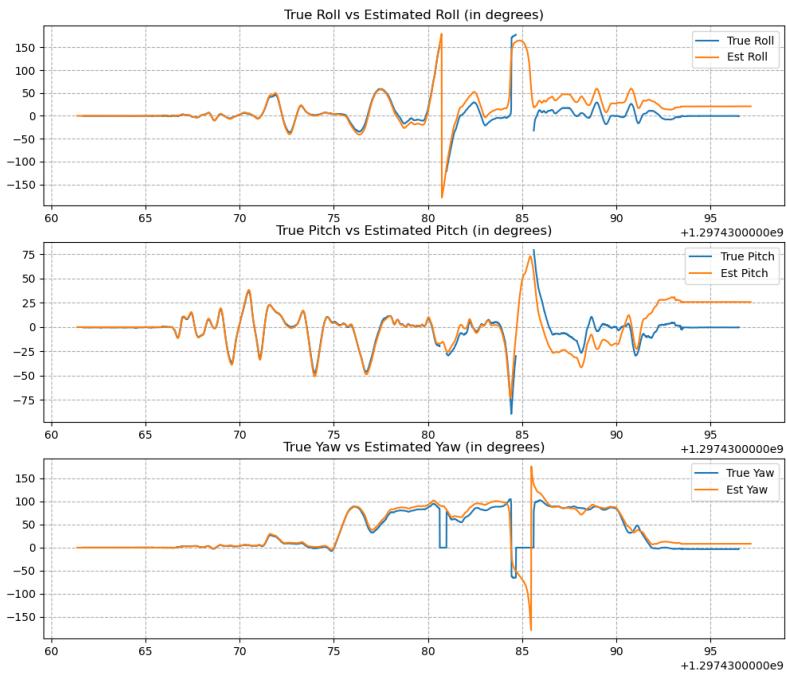


Figure 39: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

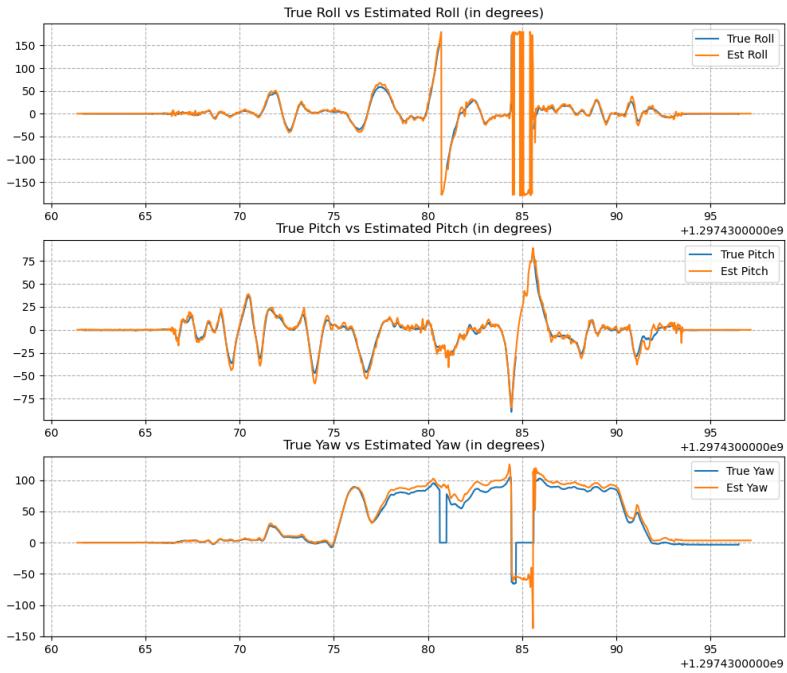


Figure 40: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)

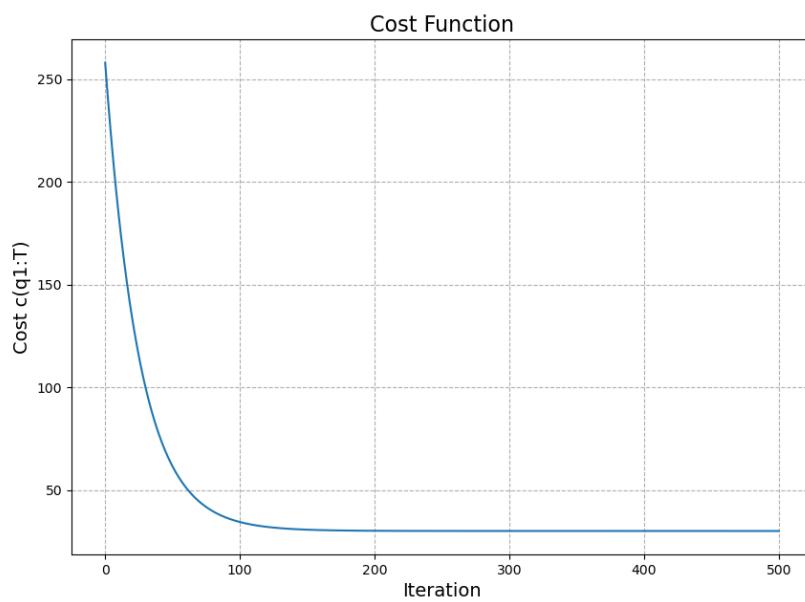


Figure 41: PGD Cost function variation

4.8 Dataset-8

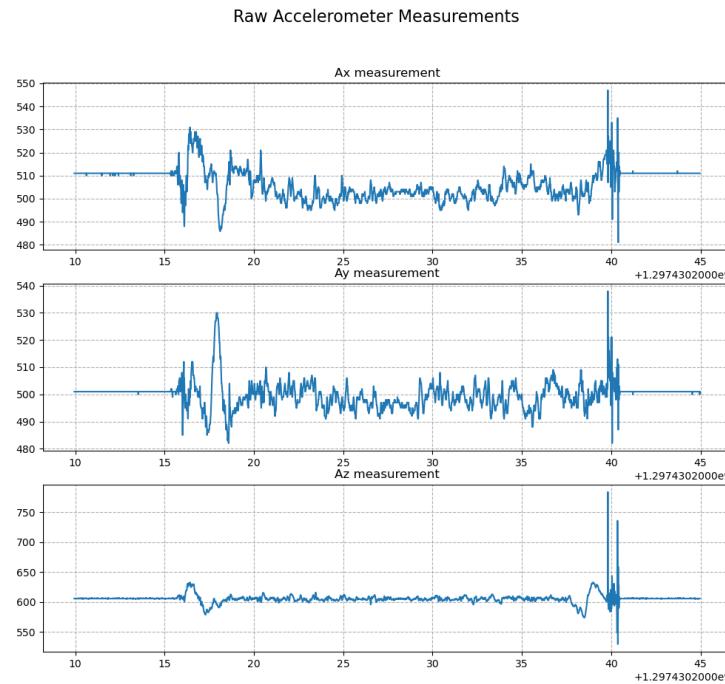


Figure 42: Raw Acceleration data

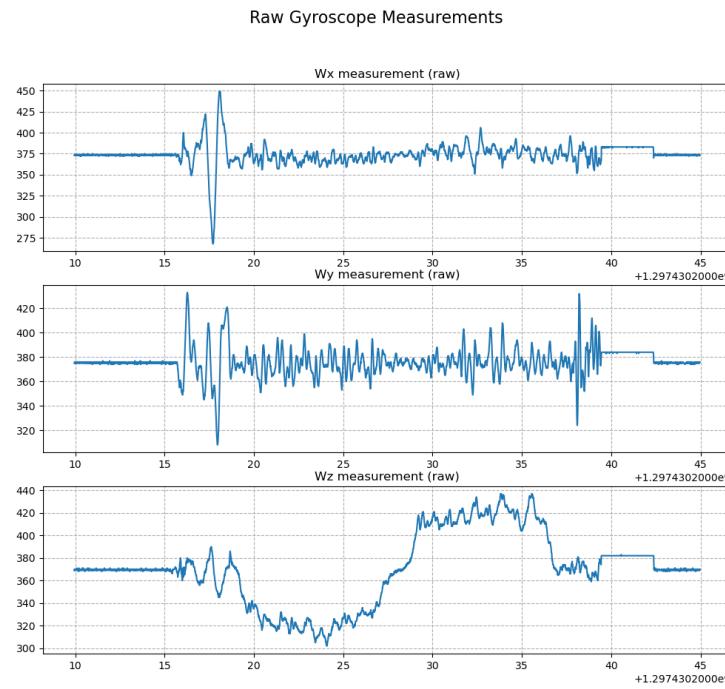


Figure 43: Raw Gyroscope data

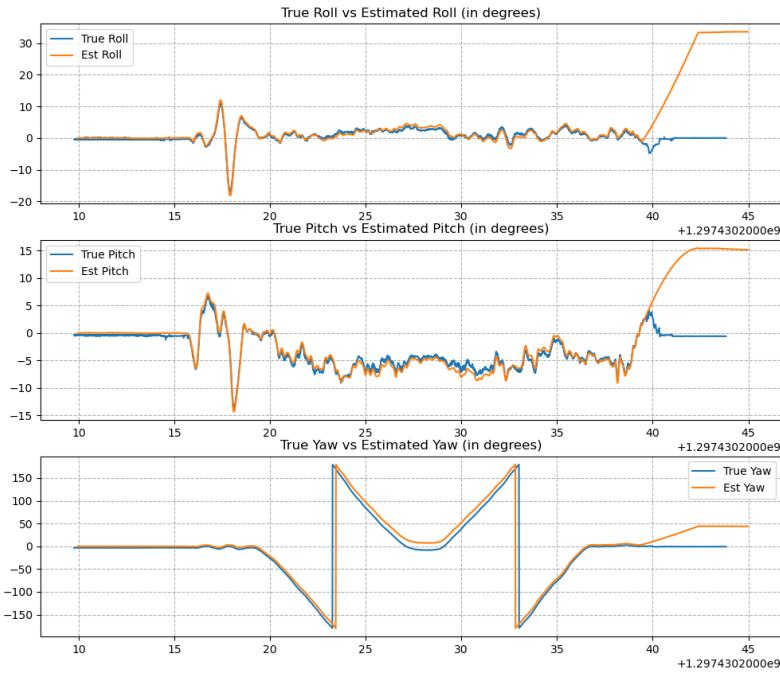


Figure 44: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

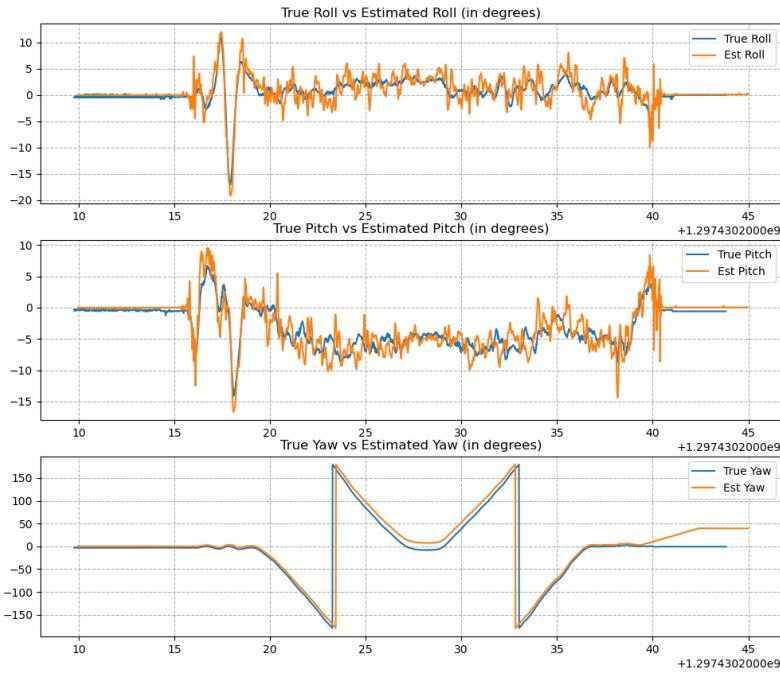


Figure 45: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)



Figure 46: Panorama Image (720 x 1280) (using VICON orientation)

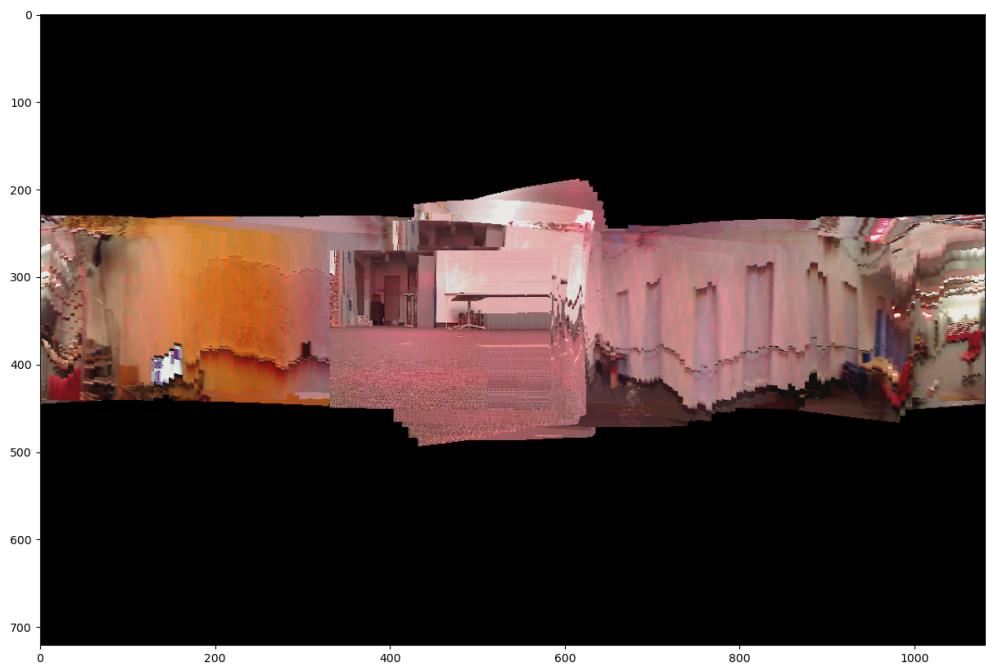


Figure 47: Panorama Image (720 x 1280) (using optimized IMU orientation)

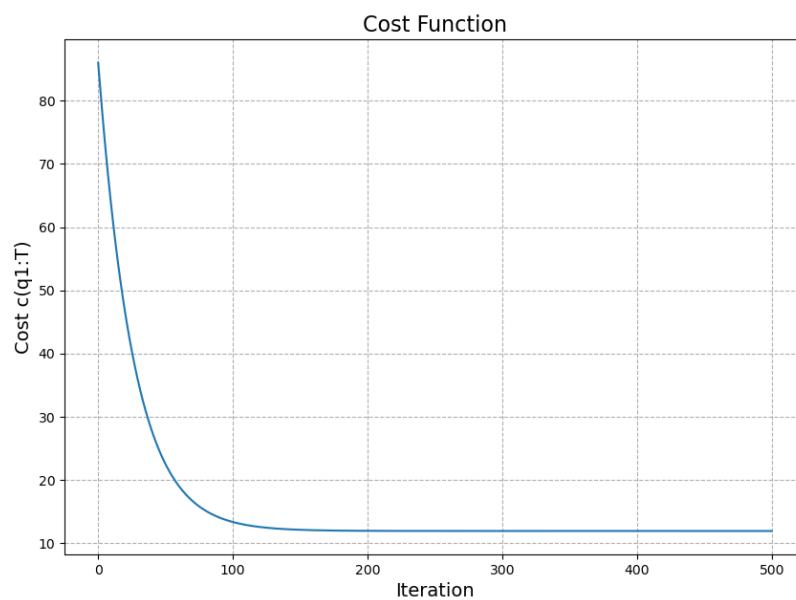


Figure 48: PGD Cost function variation

4.9 Dataset-9

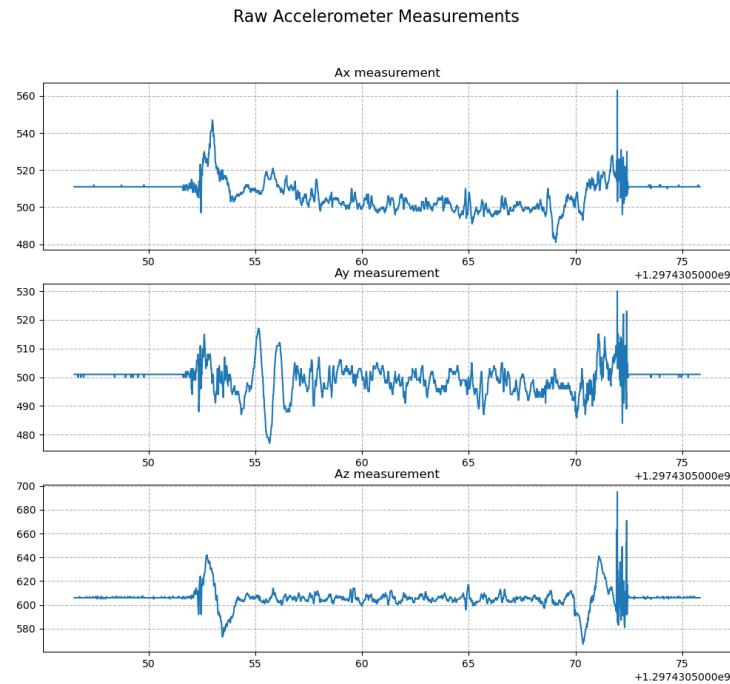


Figure 49: Raw Acceleration data

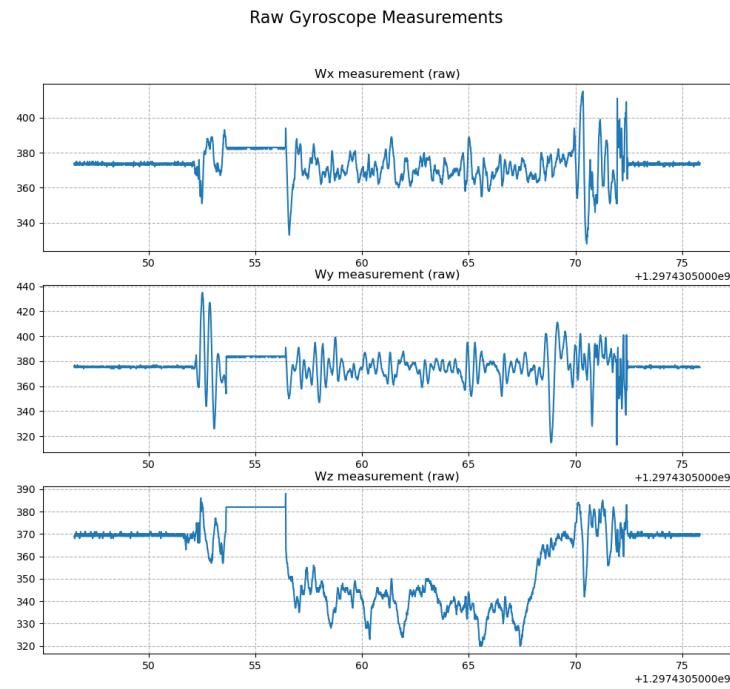


Figure 50: Raw Gyroscope data

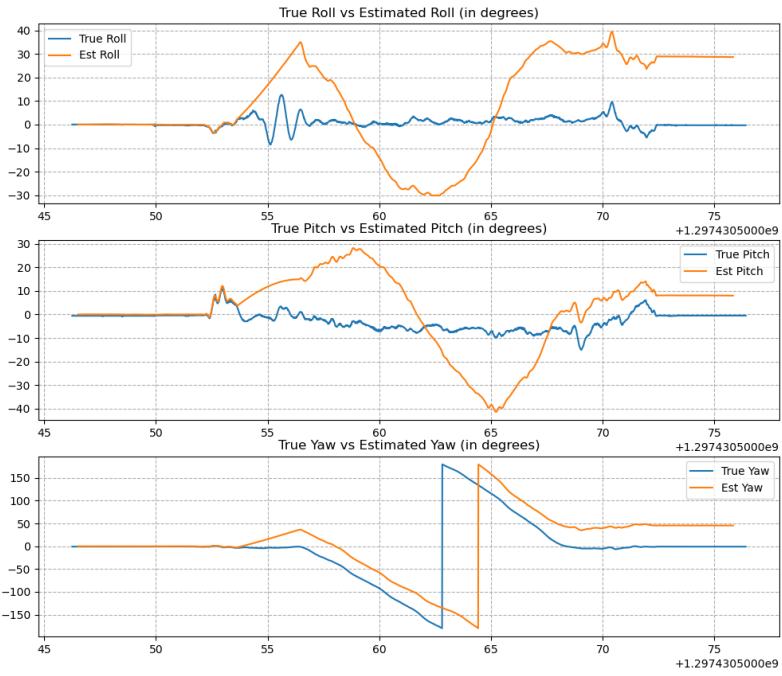


Figure 51: Estimated vs True Roll, Pitch and Yaw angles (Before PGD Optimization)

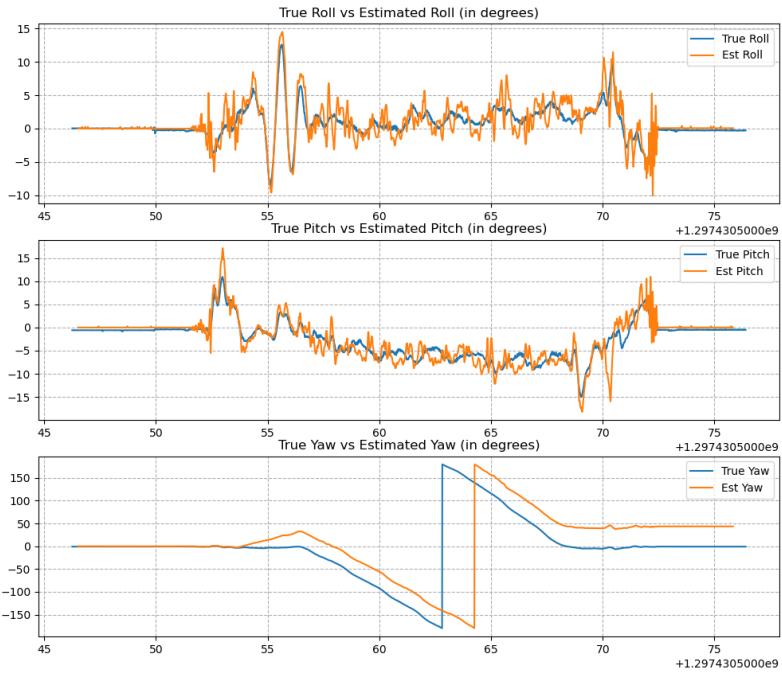


Figure 52: Estimated vs True Roll, Pitch and Yaw angles (After PGD Optimization)



Figure 53: Panorama Image (720 x 1280) (using VICON orientation)



Figure 54: Panorama Image (720 x 1280) (using optimized IMU orientation)

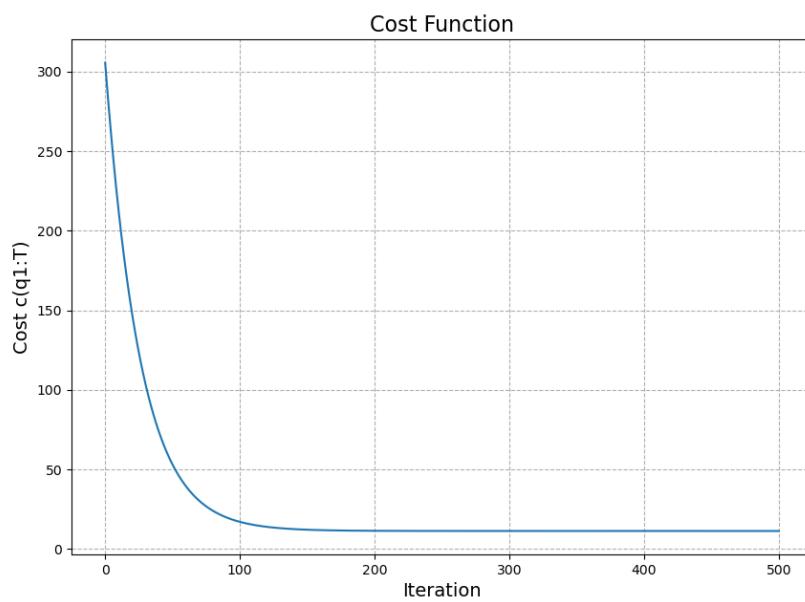


Figure 55: PGD Cost function variation

4.10 Dataset-10

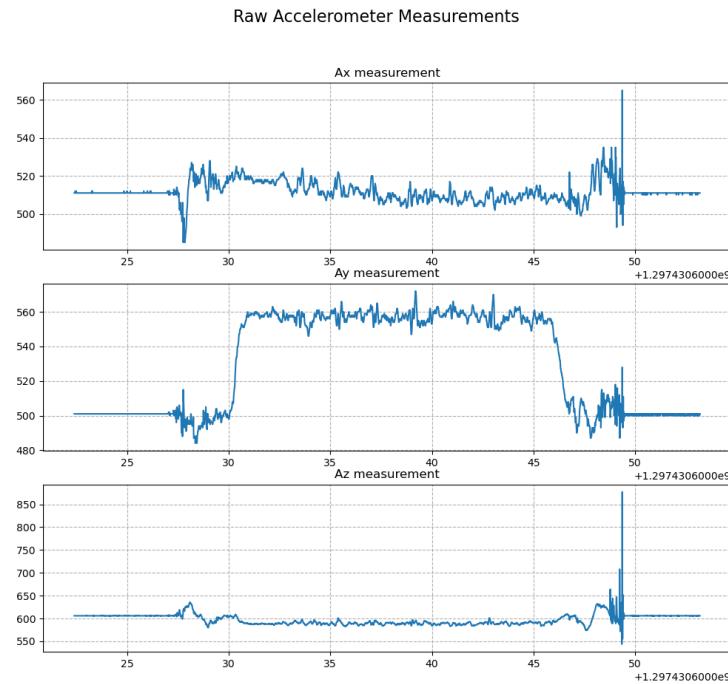


Figure 56: Raw Acceleration data



Figure 57: Raw Gyroscope data

Optimized Orientation Trajectory on test data

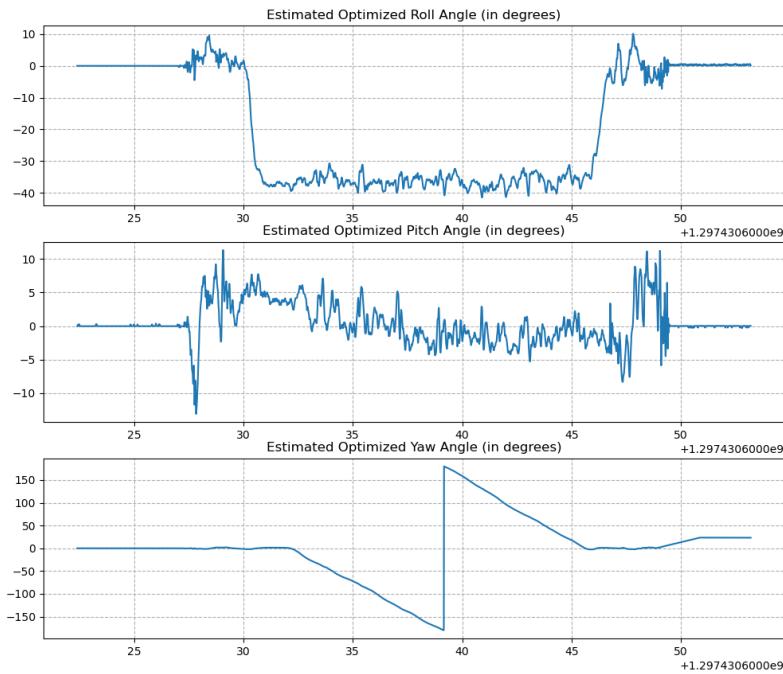


Figure 58: Estimated Roll, Pitch and Yaw angles (After PGD Optimization)

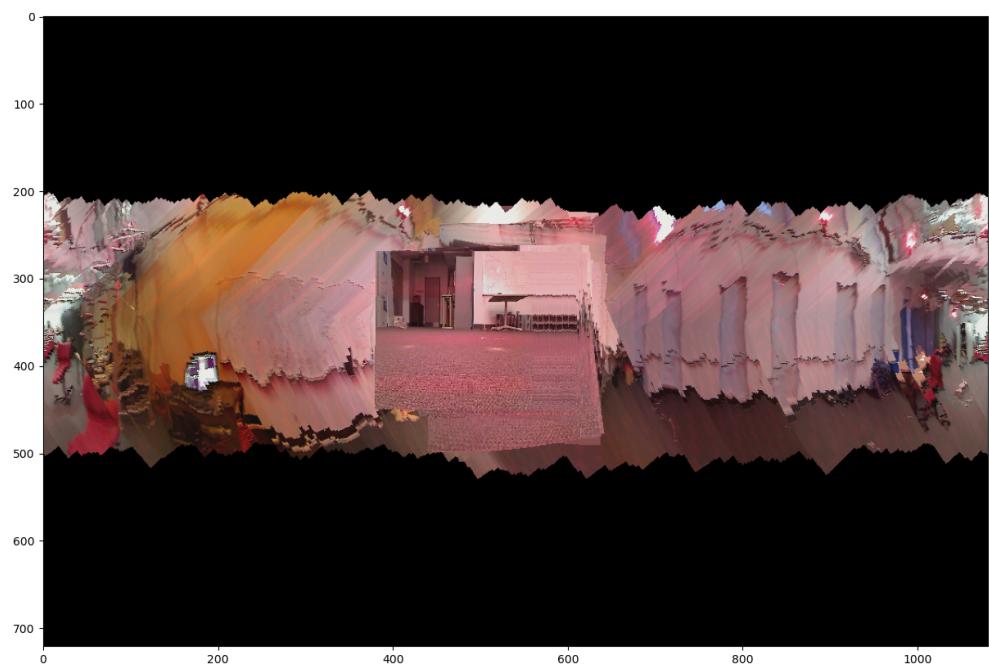


Figure 59: Panorama Image (720 x 1280) (using optimized IMU orientation)

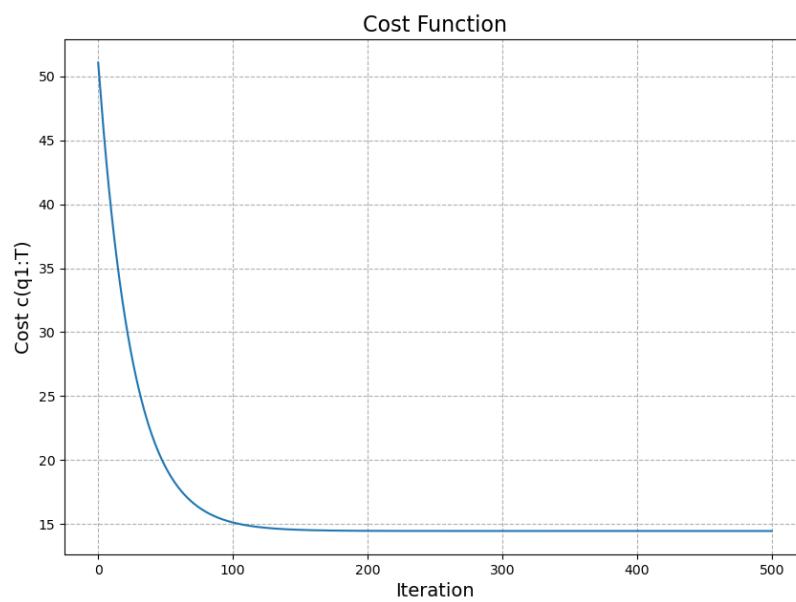


Figure 60: PGD Cost function variation

4.11 Dataset-11

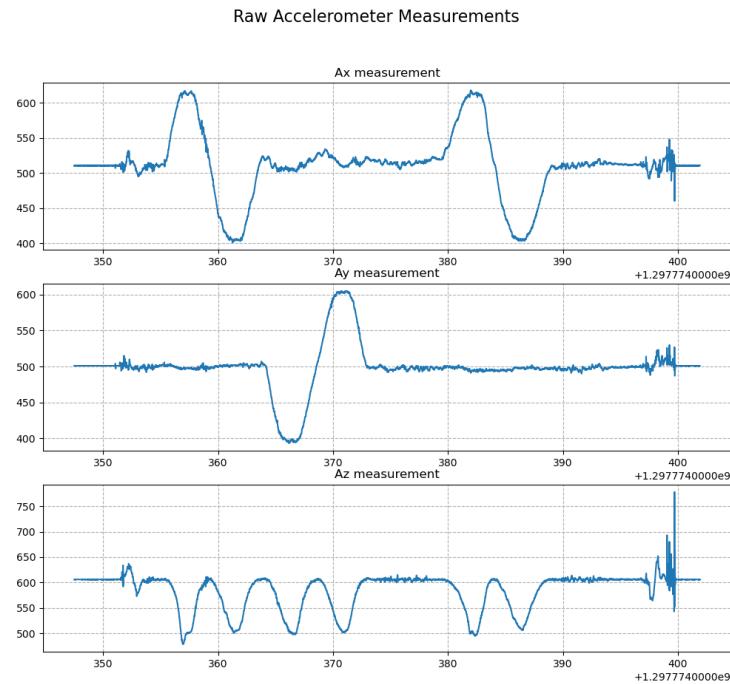


Figure 61: Raw Acceleration data

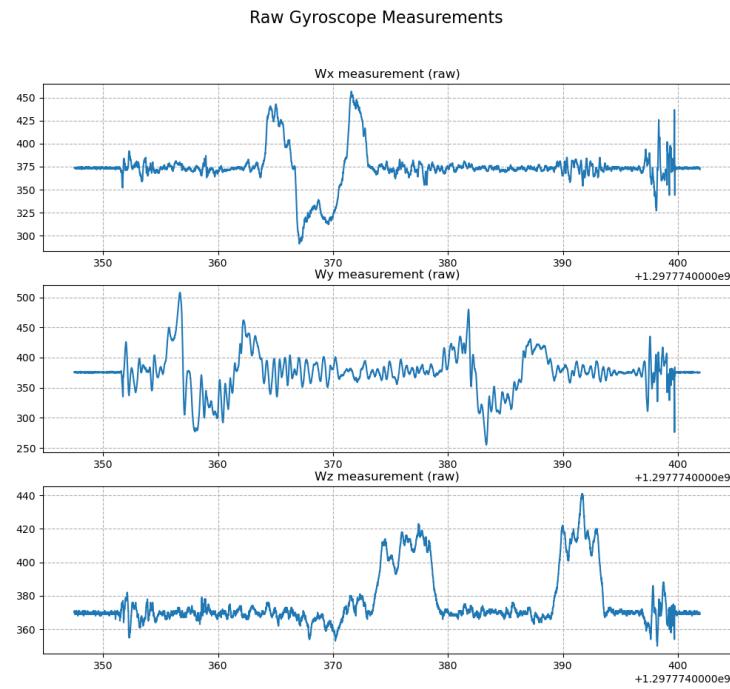


Figure 62: Raw Gyroscope data

Optimized Orientation Trajectory on test data

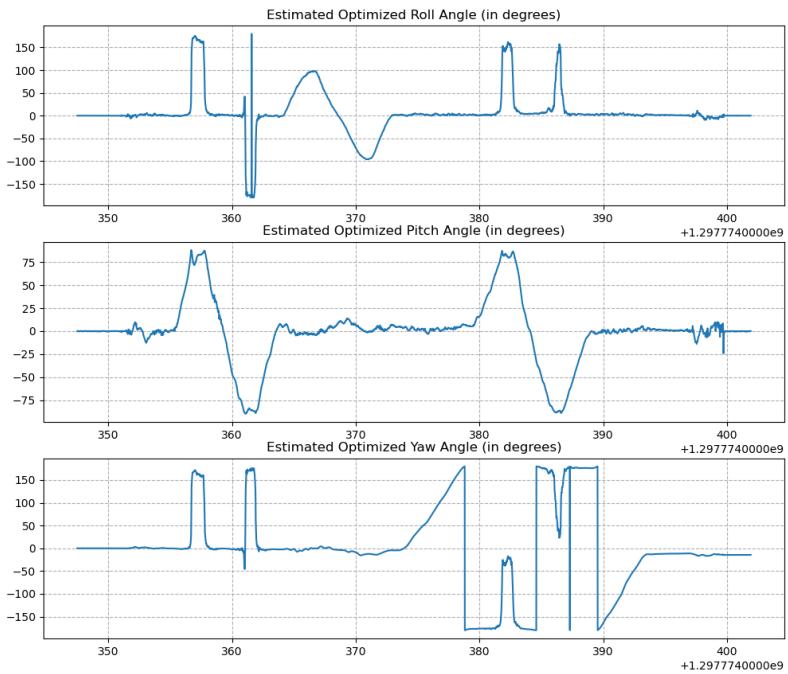


Figure 63: Estimated Roll, Pitch and Yaw angles (After PGD Optimization)

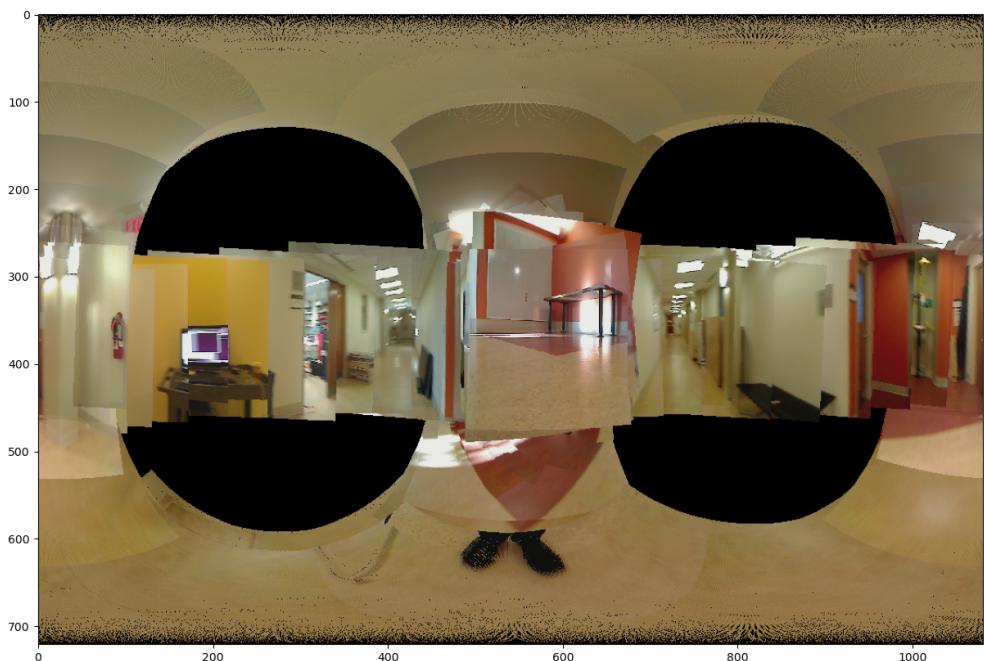


Figure 64: Panorama Image (720 x 1280) (using optimized IMU orientation)

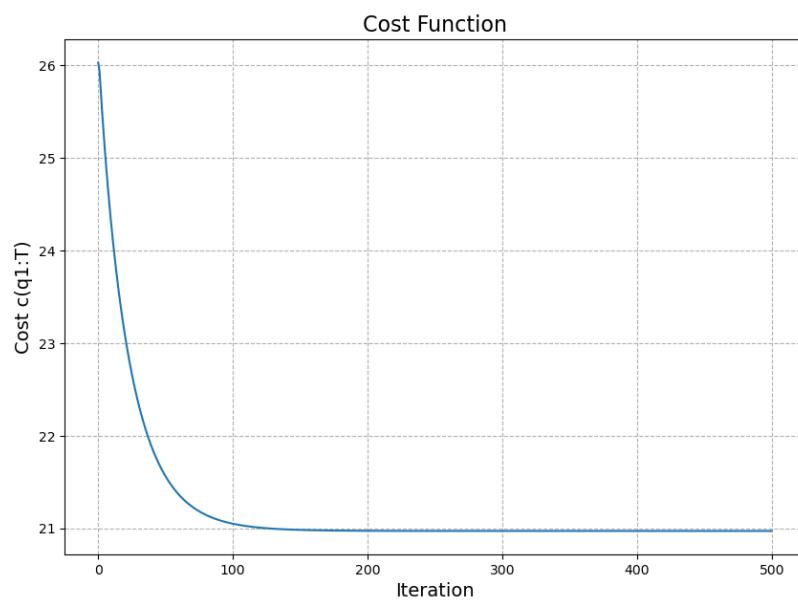


Figure 65: PGD Cost function variation