Object Tracking and Panoramic Scene Generation

Surya Lakshmi Subba Rao Pilla
Electrical and Computer Engineering
Course: Sensing and Estimation in Robotics

I. Introduction

Object tracking is a crucial part of robotics as it enables robot to operate effectively and efficiently in real-world environments. Some relevant application of object detection are in navigation and obstacle avoidance. Object tracking also plays an important role in sensing the environment and mapping the surroundings to maneuver in difficult environments. The objective of this project is to estimate the orientation of the robot using IMU data and generate a panoramic image by stitching the images obtained by camera mounted on the robot by correctly mapping the orientaion based on the estimated roll, pitch, yaw angles.

II. Problem Formulation

Given the IMU data, VICON data and camera data. The IMU data consists of angular velocity ω_t and linear acceleration a_{t} in the body frame. The VICON data consists of the ground truth rotation matrics of the body's orientation. Camera data provides 320x240 RGB images and unix time stamps at which images were collected. The are two parts of this project that are given below:

1) **Estimation of orientation of the body:** Estimate the orientation $q_{\mathrm{t}} \in H_*$ by optimizing the cost function consisting of motion and observation model using constraint gradient descent and then compare it with the available VICON data.
2) **Panorama image generation:** In this part the objective is to construct a panoramic image by stitching the images over time based on the body orientation $q_{\mathrm{1:T}}$ obtained in the first part.

III. Technical Approach

A. *Estimation of orientation of the body*

First step is to calibrate the IMU data using the scaling factor and bias. The scaling factor for accelerometer and gyroscope are derived from the data sheet. The bias is calculated by taking the average of first 500 points where the system was kep stable. Once calibration is done motion model is applied to the the unit quaternion q_{t} to predict the next step $q_{\mathrm{t+1}}$. the equation for the **motion model** is given below:

$$q_{\mathrm{t+1}} = f(q_{\mathrm{t}}, \tau_{\mathrm{t}}\omega_{\mathrm{t}}) := q_{\mathrm{t}} \circ exp([0, \tau_{\mathrm{t}}\omega_{\mathrm{t}}/2]) \tag{1}$$

The motion model assumes starting with $q_0 = [1, 0, 0, 0]$ and implements the motion model. Once the motion model is achieved the quaternions is converted to euler angles (raw,pitch and yaw) and similarly the rotation matrices obtained from ground truth are converted to euler angles. The obtained roll, pitch and yaw are compared with the VICON ground truth and plot are provided. The motion model gives us the idea of next state i.e. the orientation, but the information of the acceleration comes from the observation model. The observation model takes input as the obtained states and estimates the acceleration. As the body is in pure rotation the body should accelerate at [0,0,-g] in the world frame. Therefore the acceleration obtained through accelerometer should match with the this acceleration when rotated in IMU frame of reference. This is done through the below observation model. The observation model is rotating the acceleration due to gravity from world frame to IMU frame using the quaternions obtained from the motion model.

$$a_{\mathrm{t}} = h(q_{\mathrm{t}}) := q_{\mathrm{t}}^{-1} \circ [0, 0, 0, -g] \circ q_{\mathrm{t}} \tag{2}$$

Using the observation and motion model the next step is to estimate orientation trajectory $q_{\mathrm{1:T}}$. The cost function for the optimization problem is given below:

$$c(q_{\mathrm{1:T}}) := \frac{1}{2}\sum_{t=0}^{T-1} \left\|2log(q_{\mathrm{t+1}}^{-1} \circ f(q_{\mathrm{t}}, \tau_{\mathrm{t}}\omega_{\mathrm{t}}))\right\|_2^2 + \frac{1}{2}\sum_{t=1}^{T} \|a_t - h(q_t)\|_2^2 \tag{3}$$

$$min_{q_{1:T}} c(q_{1:T}) \left\|q_t\right\|_2 = 1; \forall t \in \ 1, 2, ...T \tag{4}$$

The above constraint is applied by doing gradient descent as shown below, where $J(q_{1:T})^k = \nabla c(q_{1:T})^k$ is the jacobian of the cost function at kth iteration:

$$q_{1:T}^{k+1} = \frac{q_{1:T}^k - \alpha J(q_{1:T})^k}{\left\|q_{1:T}^k - \alpha \ J(q_{1:T})^k\right\|} \tag{5}$$

The implementation was done with 10 iterations and $\alpha = 0.01$. These parameters are tunable to achieve better results.

B. *Panorama image generation*

The first step of panoramic image generation is to convert pixel coordinates to the spherical. Assuming that the image lies in the unit sphere the radius r will be zero, longitude and latitude are derived given vertical Field of View(FOV) as 45° and horizontal FOV as 60°. Therefore longitude $\lambda \in [-30, 30]$ and latitude $\phi \in [-22.5, 22.5]$. Considering the convention such that top left corner has $(\lambda, \phi, 1)$ as [30,-22.5,1], the pixels are mapped to latitude and longitude by a phase shift of $\frac{\pi}{2}$. Next, these spherical coordinates are converted to cartesian coordinates by below formula(subscript c tells that it is in camera frame):

$$x_c = \cos\lambda \sin\phi \tag{6}$$

$$y_c = \sin\lambda \sin\phi \tag{7}$$

$$z_c = \cos\phi \tag{8}$$

The cartesian coordinates obtained are in camera frame, are now converted to world frame by doing rotation and translation. The rotation matrix is obtained from the part 1. The equation below is used to convert coordinates from camera to world frame, where ${}_wR_c \in \mathbb{R}^{3x3}$ is the rotion matrix from camera to world frame, p is the position vector which is [0,0,0.1] as the camera is 10cms above the IMU (subscript w indicates that it is in world frame):

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = {}_wR_c \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + p \tag{9}$$

The world cartesian coordinates are converted to spherical coordinates with radius 1 (r_w = 1) using the below formula:

$$\lambda_w = \arctan(\frac{y_w}{x_w}) \tag{10}$$

$$\phi_w = \arccos(z_w) \tag{11}$$

The next step is to inscribe the sphere in a cylinder so that a point on sphere has height ϕ on the cylinder and longitude λ along the circumference of the cylinder. After this the cylindrical surface is unwrapped into a rectangular image with width 2π radians and height π radians. This is done by taking a blank image kernel with panCols, panRows as columns and rows and then projecting images on to it. This is done by first projecting the spherical coordinates onto cylinder, unwrapping cylinder and scaling it to fit onto the blank kernel using below equations:

$$pan_x = \frac{\lambda_w}{\pi} * panRows \tag{12}$$

$$pan_y = \frac{\phi_w}{2\pi} * panCols \tag{13}$$

IV. Results

The plots for the 9 training dataset comparing the Roll, Pitch and Yaw angles are shown in figures 1-27. The optimized angles are closer to the ground truth as the gradient cost function is converging to local minima. The roll, pitch and yaw are further used to build the observation model that estimates the acceleration of the body. The panorama images for training data are given in the figures 28-35, these are obtained by taking the rotation matrix obtained through optimization and compared with that obtained by VICON ground truth data. There is certainly some deviation this can be solved by optimizing for more number of iterations and tuning the α in gradient descent step. The results of test data are given in figures 36-43, figures 39,43 are the panoramic images obtained by the estimated roll, pitch, yaw obtained by optimization.

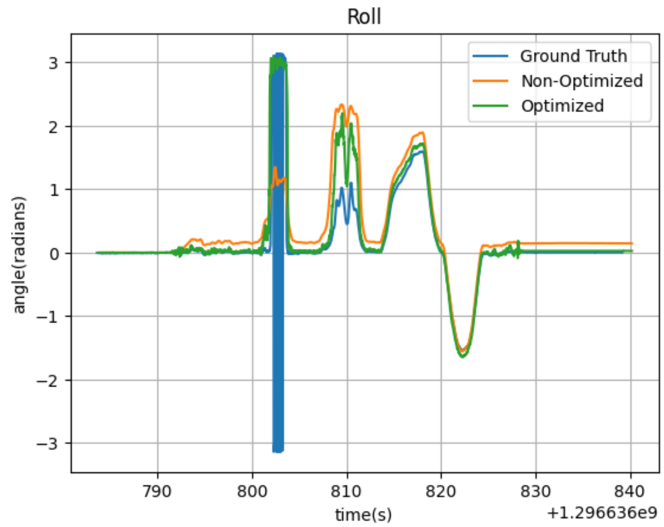

Figure 1: Train data 1

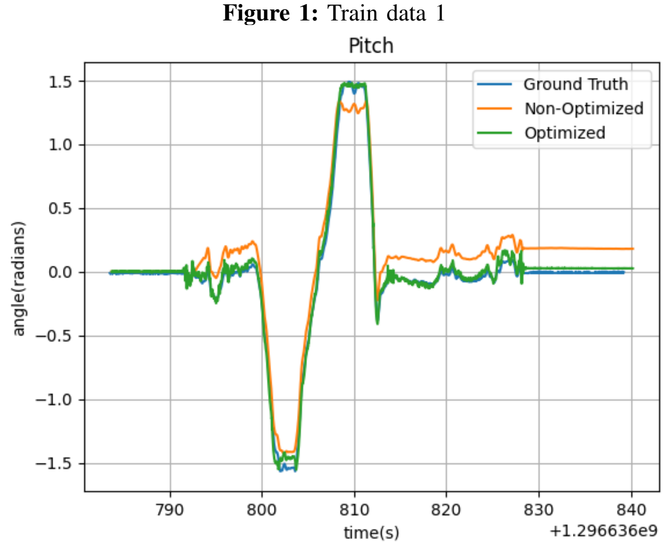

Figure 2: Train data 1

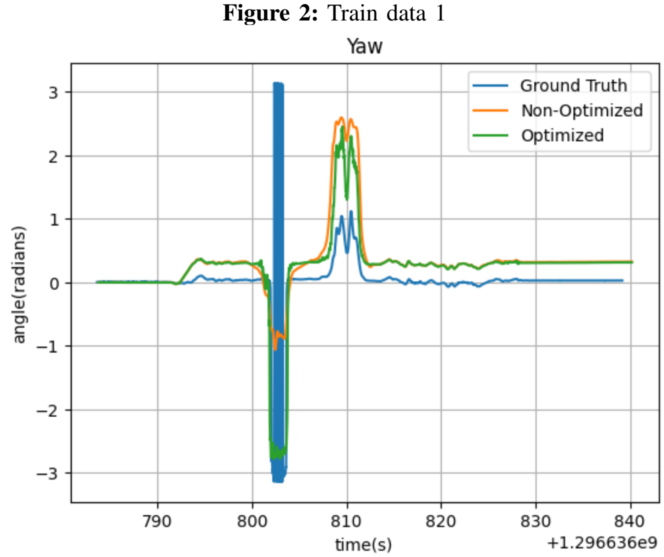

Figure 3: Train data 1

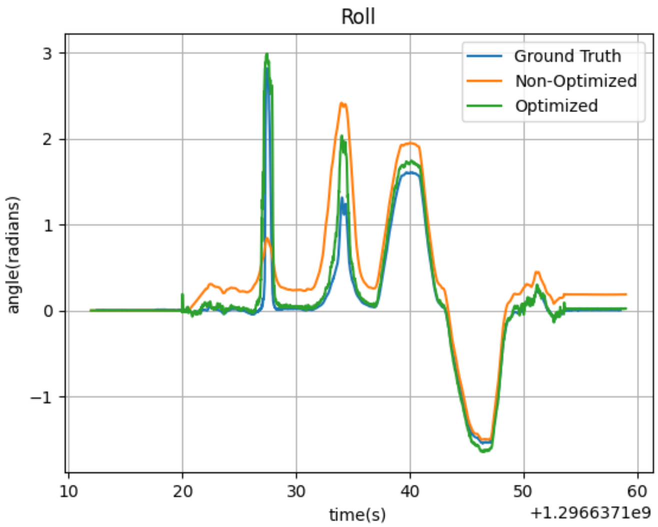

Figure 4: Train data 2

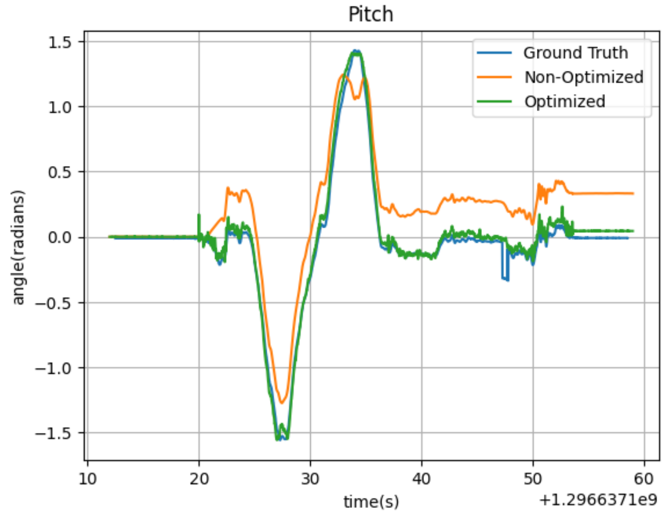

Figure 5: Train data 2

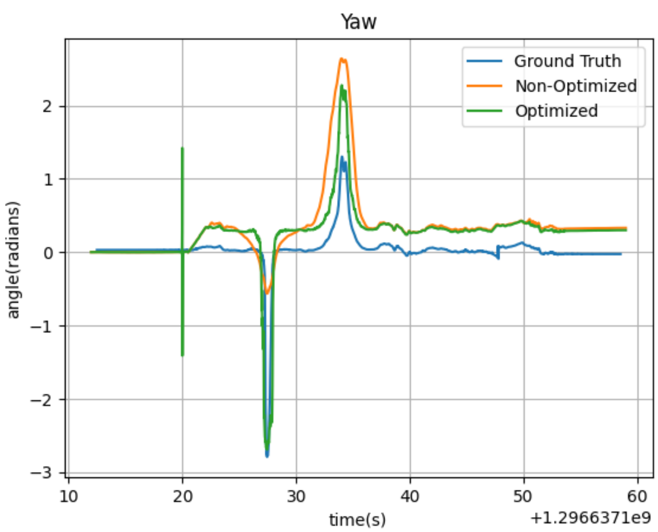

Figure 6: Train data 2

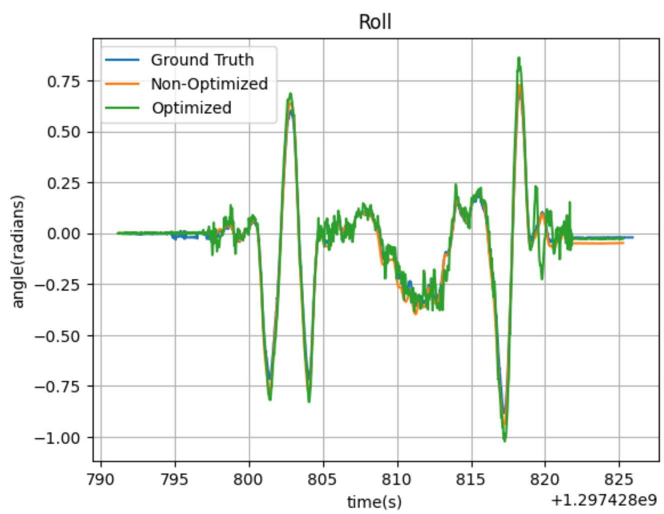

Figure 7: Train data 3

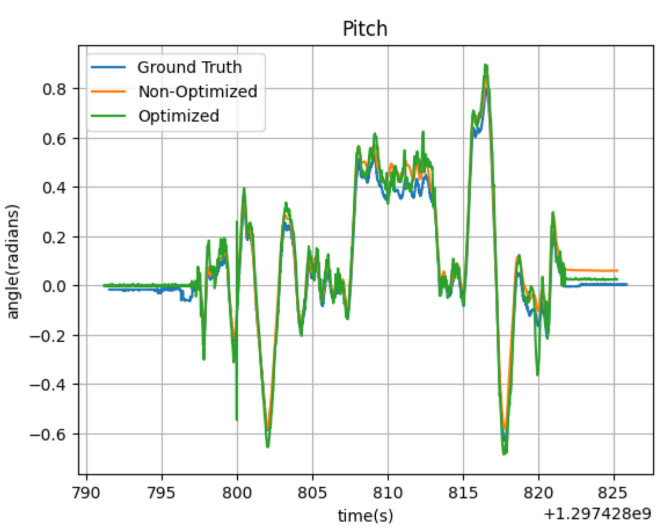

Figure 8: Train data 3

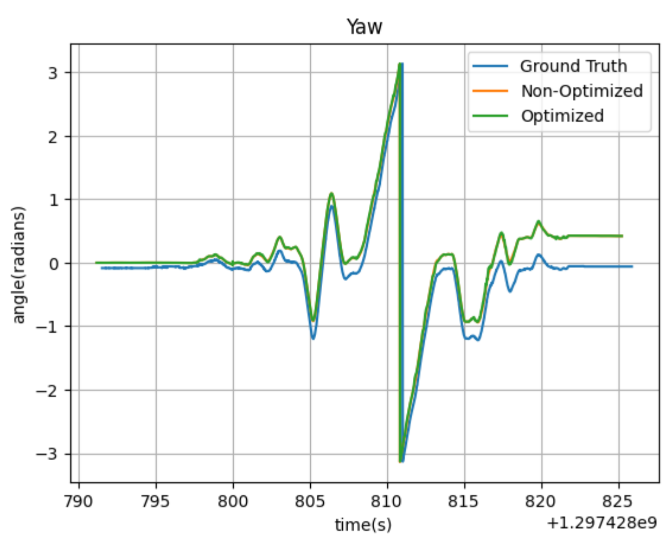

Figure 9: Train data 3

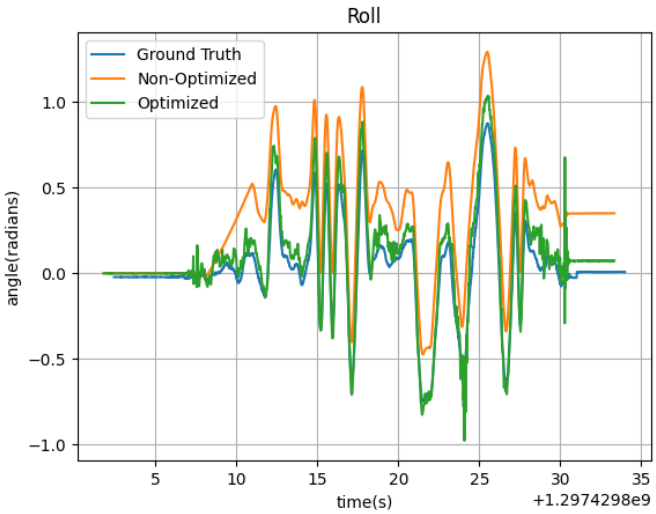

Figure 10: Train data 4

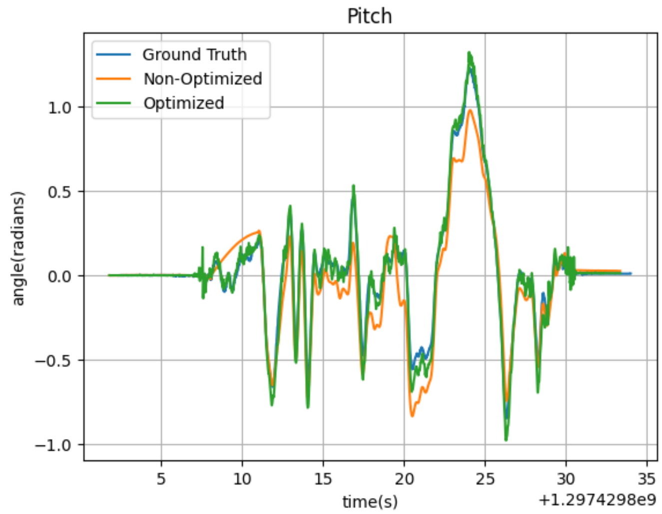

Figure 11: Train data 4

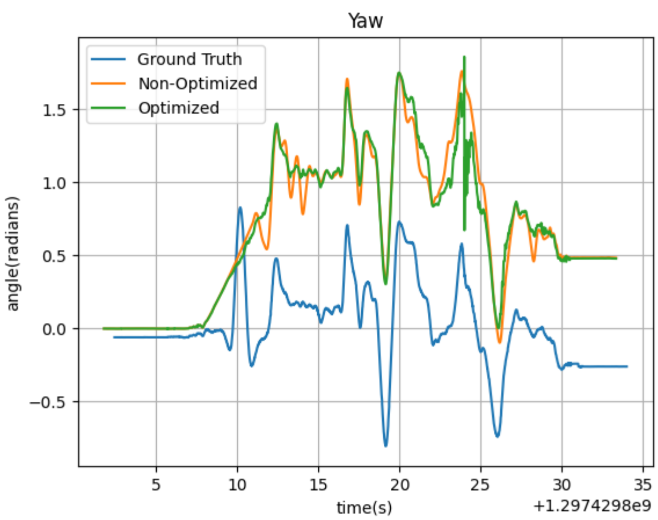

Figure 12: Train data 4

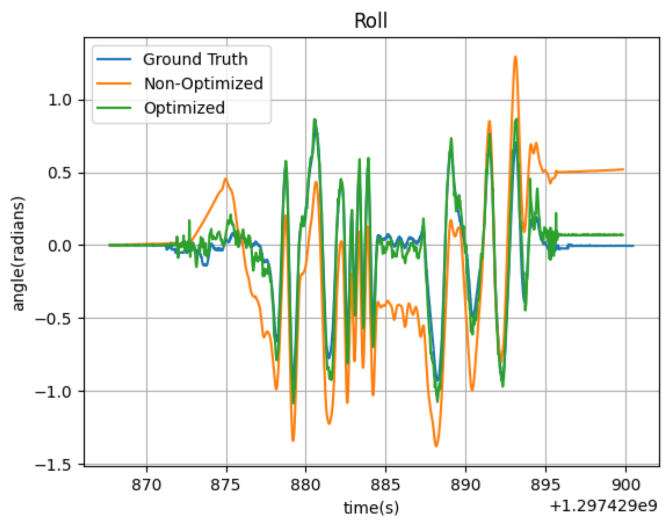

Figure 13: Train data 5

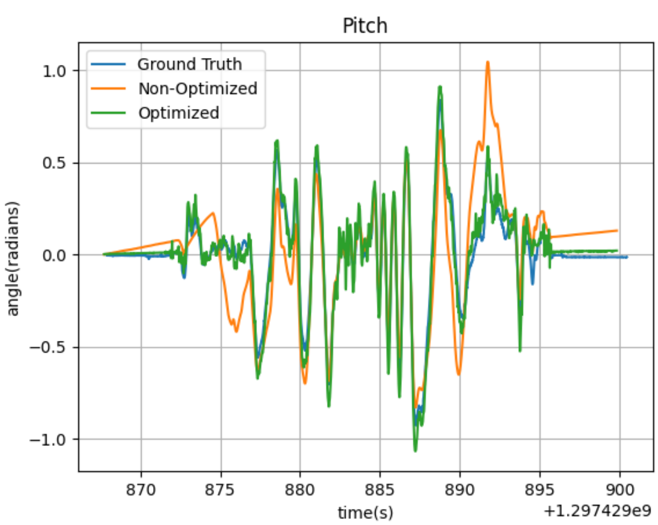

Figure 14: Train data 5

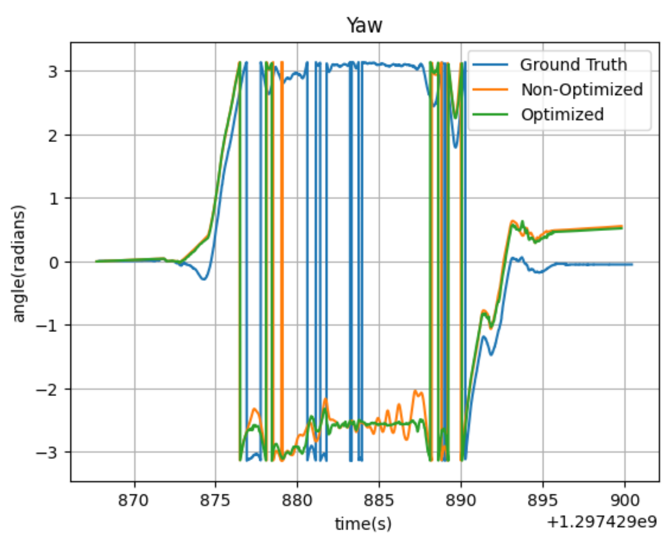

Figure 15: Train data 5

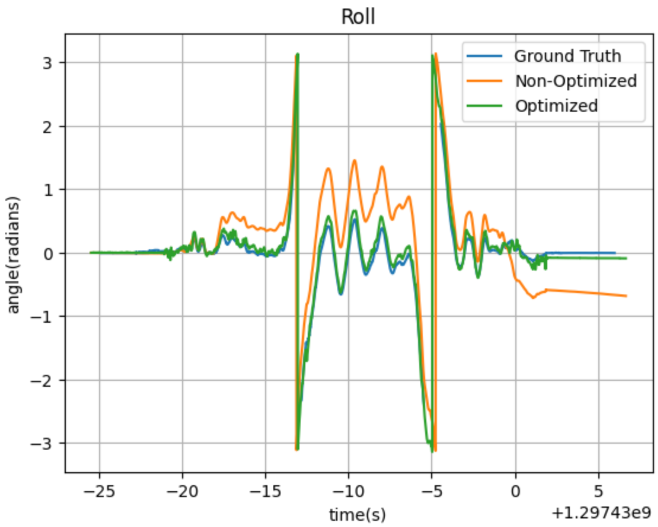

Figure 16: Train data 6

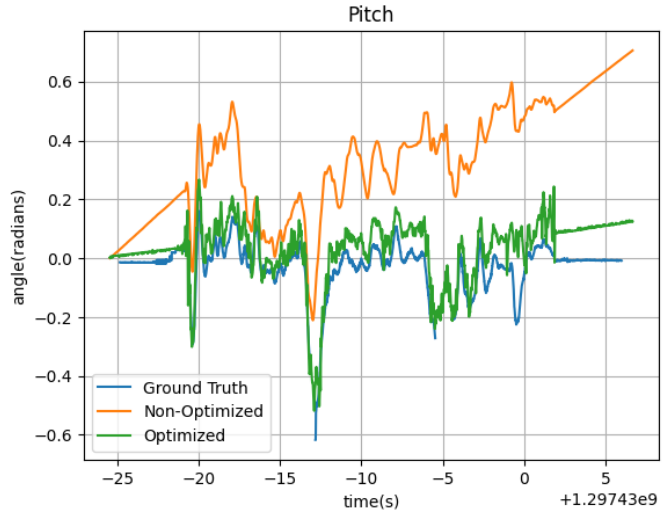

Figure 17: Train data 6

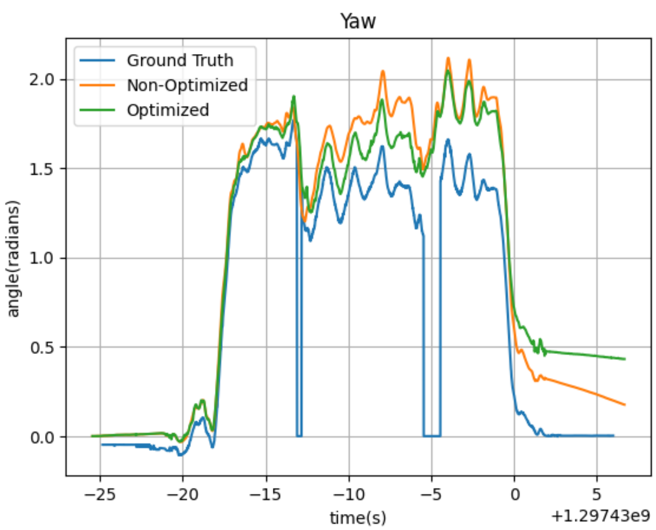

Figure 18: Train data 6

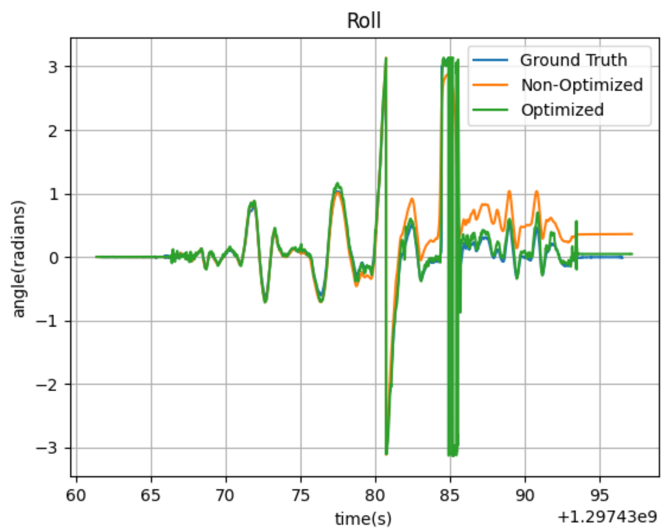

Figure 19: Train data 7

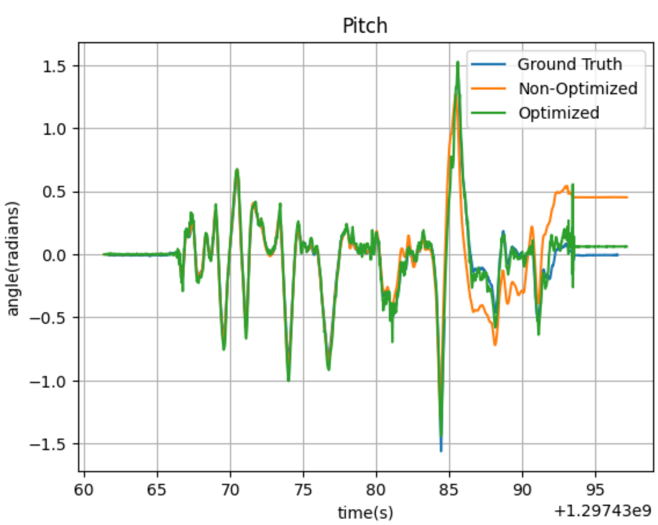

Figure 20: Train data 7

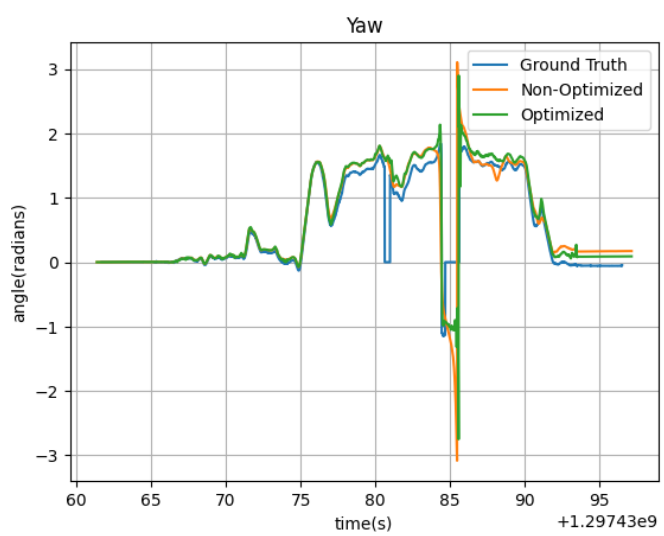

Figure 21: Train data 7

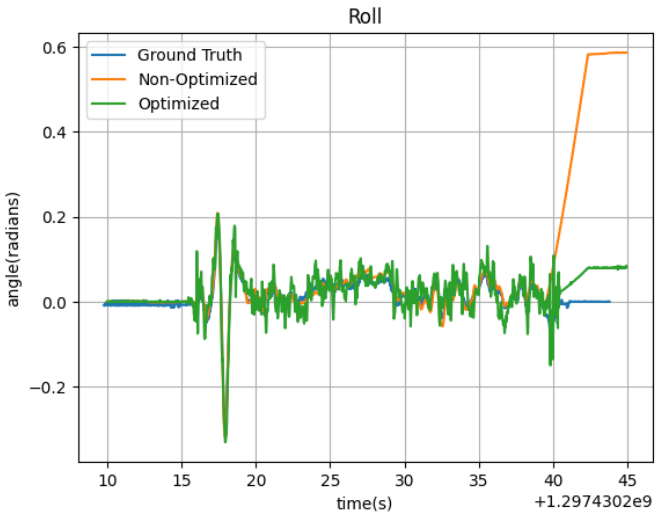

Figure 22: Train data 8

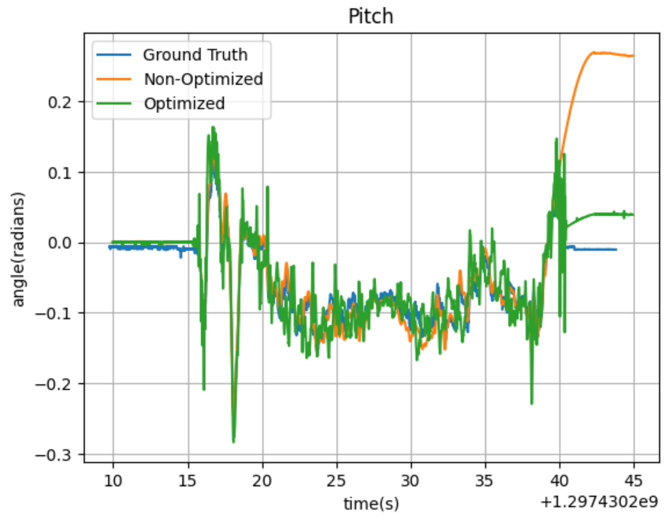

Figure 23: Train data 8

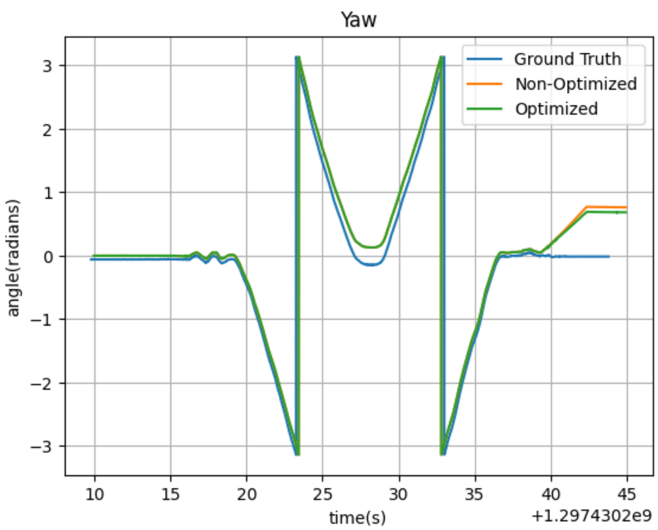

Figure 24: Train data 8

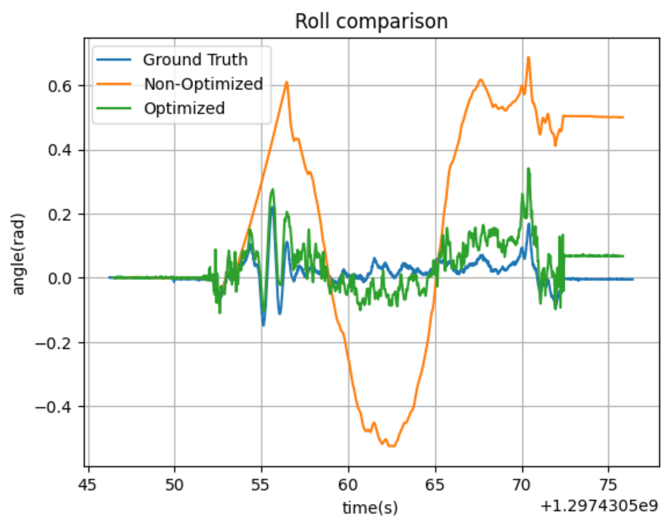

Figure 25: Train data 9

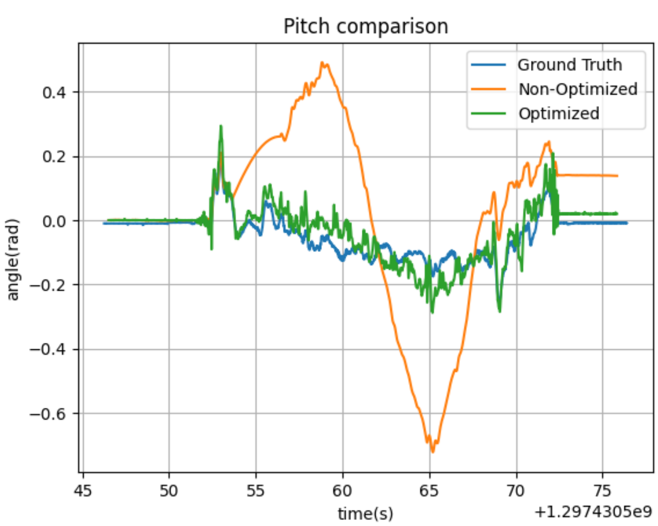

Figure 26: Train data 9

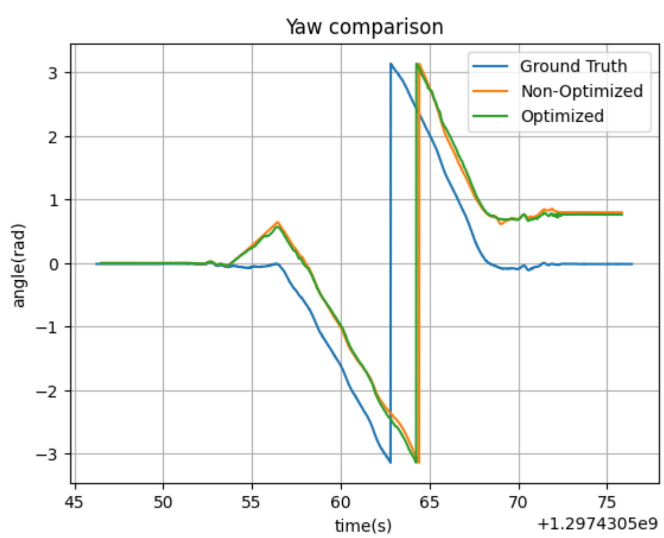

Figure 27: Train data 9

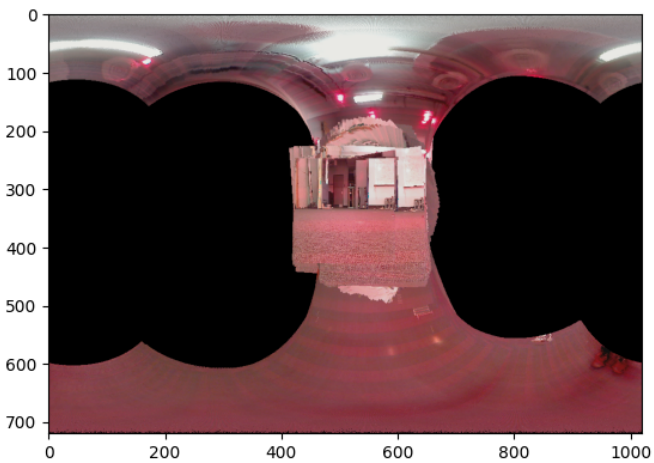

Figure 28: Train data 1 Pan image based on estimated rotation

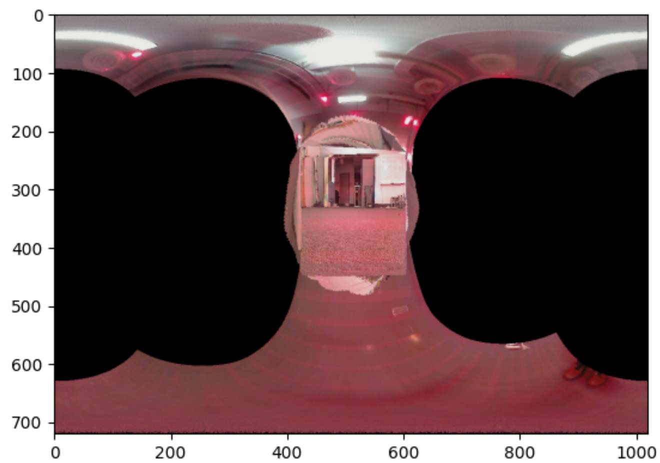

Figure 29: Train data 1 Pan Image based on VICON

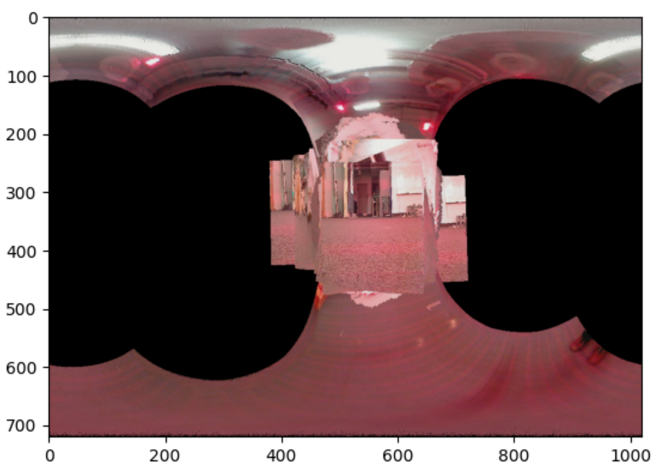

Figure 30: Train data 2 Pan image based on estimated rotation

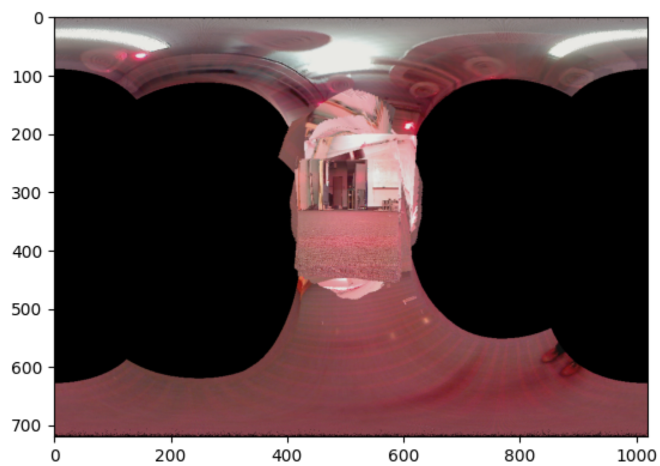

Figure 31: Train data 2 Pan Image based on VICON

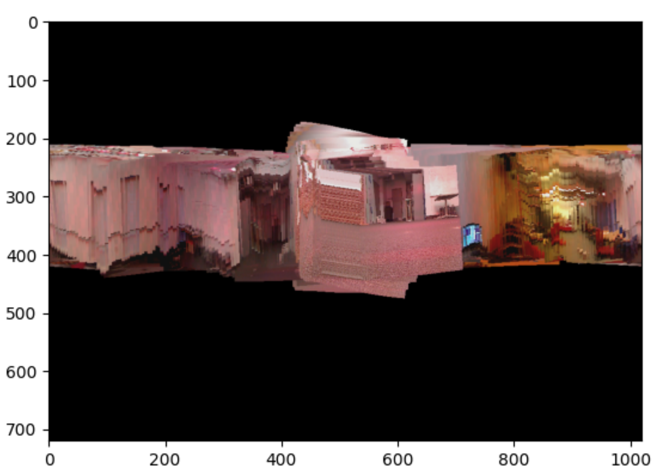

Figure 32: Train data 8 Pan image based on estimated rotation

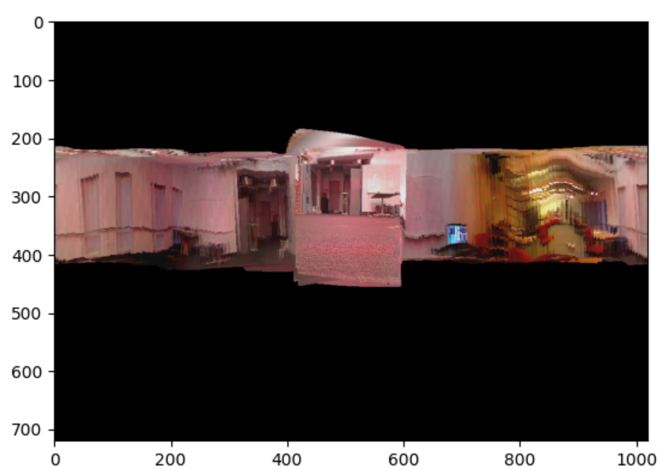

Figure 33: Train data 8 Pan Image based on VICON

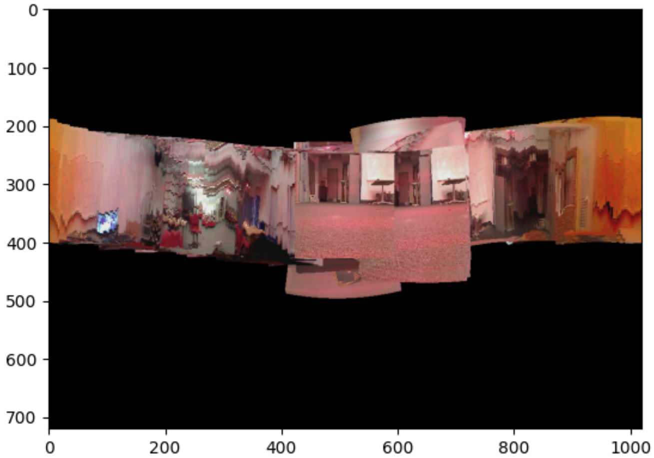

Figure 34: Train data 9 Pan image based on estimated rotation

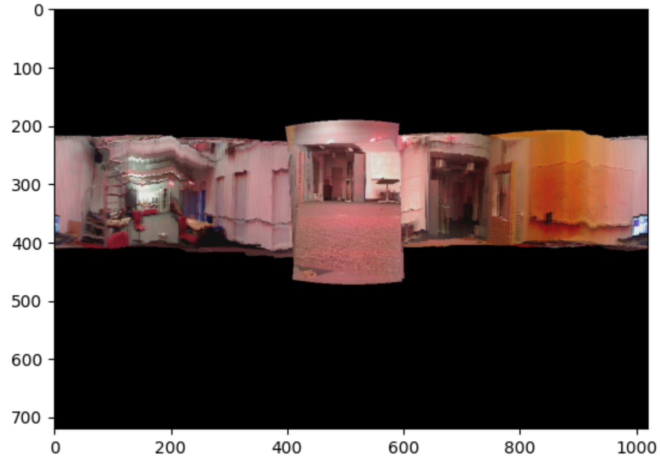

Figure 35: Train data 9 Pan Image based on VICON

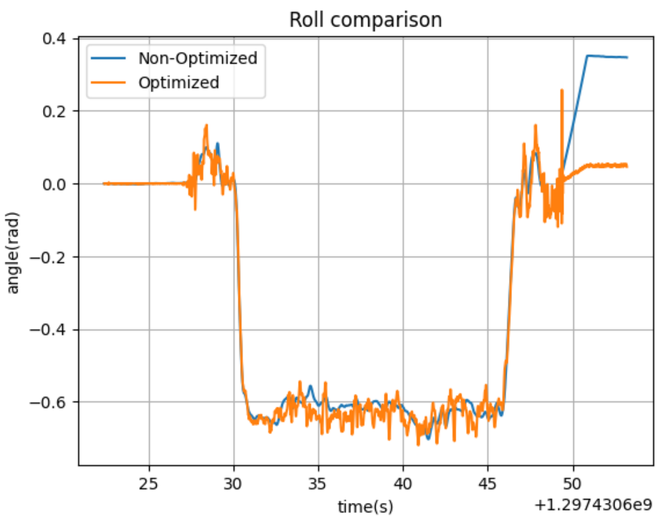

Figure 36: Test data 10 roll

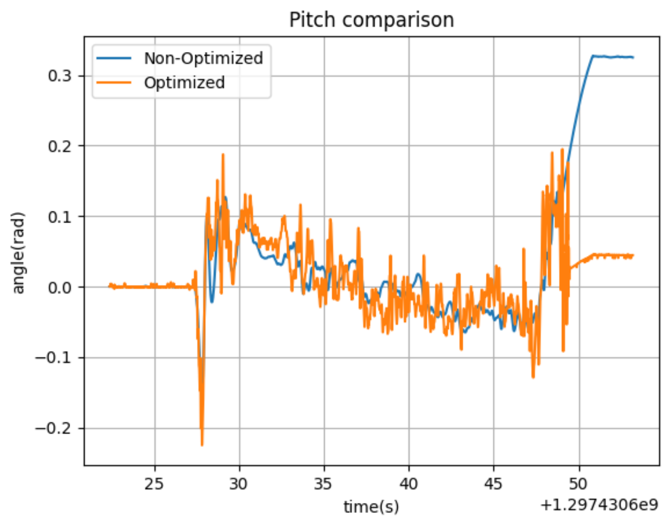

Figure 37: Test data 10 pitch

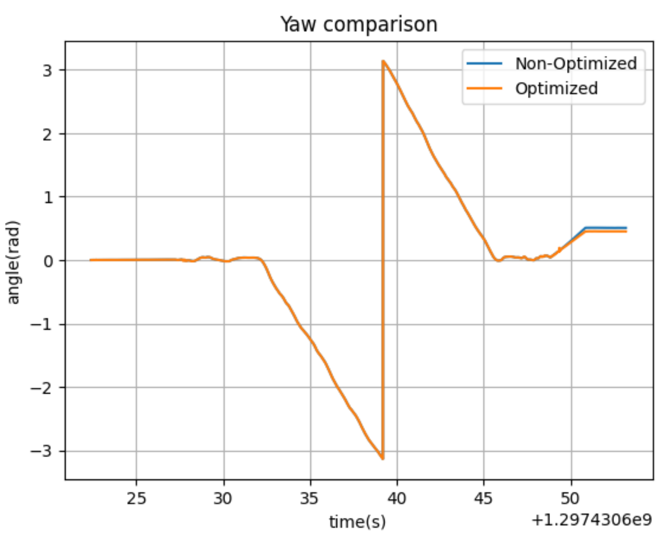

Figure 38: Test data 10 yaw

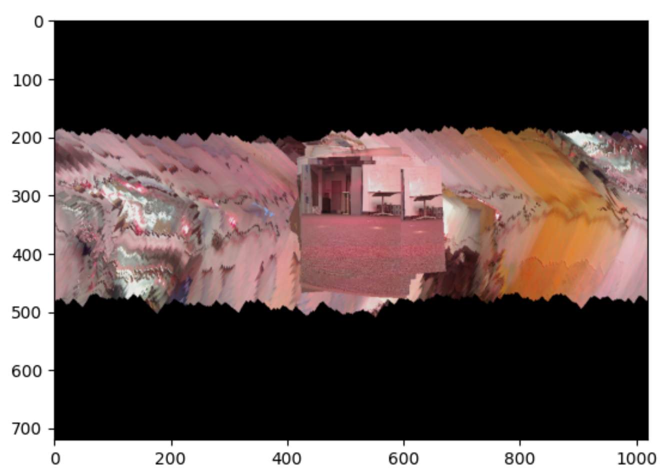

Figure 39: Test data 10 Pan Image based on estimation

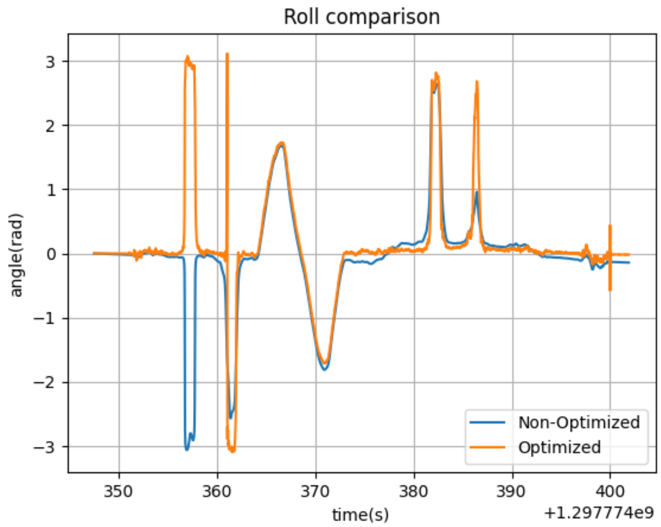

Figure 40: Test data 11 roll

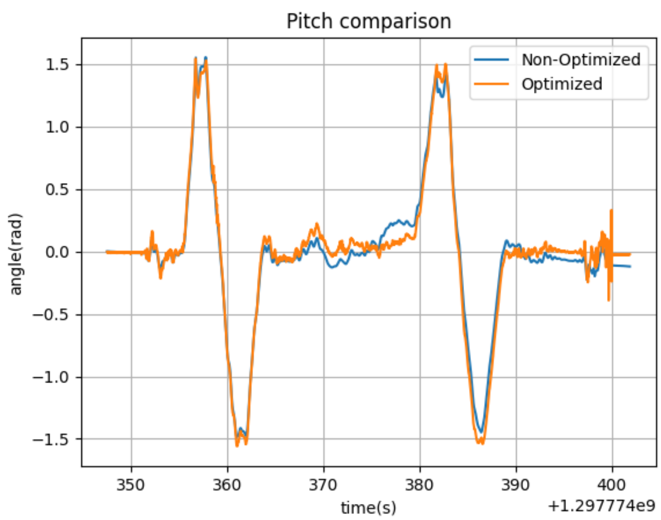

Figure 41: Test data 11 pitch

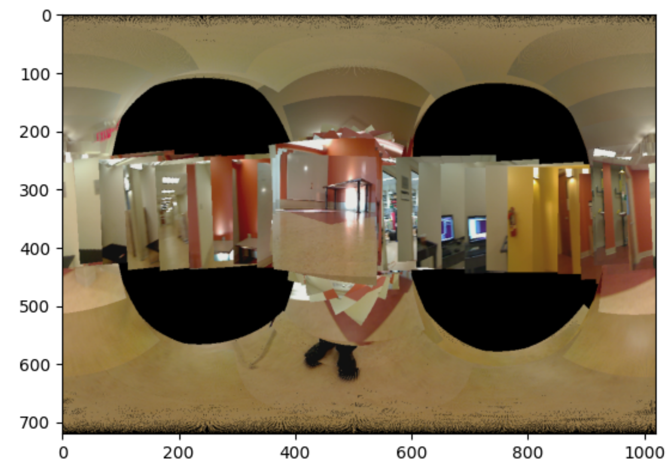

Figure 43: Test data 11 Pan Image based on estimation

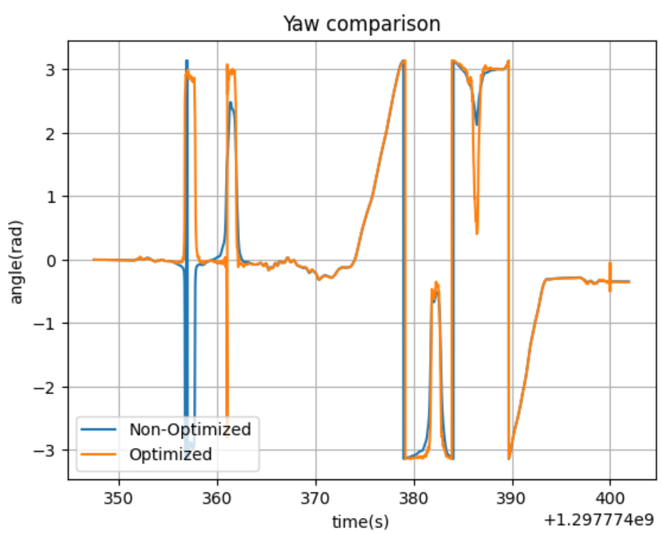

Figure 42: Test data 11 yaw