ECE276A Project 1: Orientation Tracking

Wilson Liao
Department of Electrical and Computer Engineering
University of California, San Diego
w4liao@ucsd.edu

I. Introduction

There are 2 parts in this project, which are Orientation Tracking and Panorama.

For orientation tracking, it refers to the process of measuring and monitoring the orientation or angular position of an object in 3-D space. This can be done using various sensors, such as accelerometers, gyroscopes, or magnetometers, and is commonly used in applications such as virtual reality, robotics, and navigation systems.

Our approach to this problem is to first calibrate the IMU readings and then plugin to the motion model to predict the next orientation. Besides, we also combine motion model and observation model to form a optimization problem and apply projected gradient descent to get the final estimation of our quaternion trajectory.

For panorama, it is a wide and expansive view of a physical space or landscape, typically captured using a camera or created by stitching together multiple photographs. It provides a wider and more immersive perspective than a regular photograph, allowing the viewer to see the scene as if they were actually present at the location. Panoramas are often used for artistic, commercial, or documentary purposes, and can be viewed as static images or interactive 360-degree panoramas.

Our approach to this problem is to first calibrate the IMU readings and then plugin to the motion model to predict the next orientation. Besides, we also combine motion model and observation model to form a optimization problem and apply projected gradient descent to get the final estimation of our quaternion trajectory.

II. Problem Formulation

A. Orientation Tracking

Consider a body rotates in an environment, and the IMU sensor on the robot measures the angular velocity w_t and linear acceleration a_t at time t. Given the quaternion kinematics motion model:

$$q_{t+1} = f(q_t, \tau_t w_t) := q_t \circ exp([0, \tau_t w_t/2]) \tag{1}$$

where τ_t is the differences between consecutive time stamps and $\exp(\cdot)$ is the exponential function for quaternions.

With this, we can predict the quaternion at the next step q_{t+1}.

Besides, since the robot is undergoing pure rotation, the acceleration of the body should be approximately $[0, 0, -g]$ in the world frame reference, where g is the gravity acceleration. With this, we can get the observation model:

$$a_t = h(q_t) := q_t^{-1} \circ [0, 0, 0, -g] \circ q_t \tag{2}$$

Combining (1) and (2), we can formulate an optimization problem to estimate the orientation trajectory $q_{1:T} := q1, q2, ..., q_T$. The following is the cost function:

$$\begin{aligned} c(q_{1:T}) := \frac{1}{2} \sum_{t=0}^{T-1} ||2log(q_{t+1}^{-1} \circ f(q_t, \tau_t w_t))||_2^2 + \\ \frac{1}{2} \sum_{t=1}^{T} ||a_t - h(q_t)||_2^2 \end{aligned} \tag{3}$$

where the first norm indicates the error between the estimated orientation and the motion model prediction, while the second norm indicates the error between the acceleration and the observation model prediction. For this optimization problem, we have the following constraints:

$$\begin{aligned} &\min_{q_1:T} c(q_{1:T}) \\ &s.t. ||q_t||_2 = 1, \forall t \in 1, 2, ...T \end{aligned} \tag{4}$$

The goal of this problem is to estimate the orientation trajectory $q_{1:T} := q1, q2, ..., q_T$.

B. Panorama

Consider body orientation $q_{1:T}$ and camera images, our objective is to construct a panoramic image by stitching the RGB camera images. Suppose the image lies on a sphere and compute the world coordinates of each pixel. We can perform the following steps to generate panorama.

- Find the corresponding longitude λ and latitude ϕ of each pixel using the number of rows and columns and the horizontal $60°$ and vertical $45°$ fields of view.
- Convert the spherical coordinate $(\lambda, \phi, 1)$ to Cartesian coordinates assuming depth equaling to 1.
- Rotate the Cartesian coordinates to the world frame using rotation matrix R.
- Convert Cartesian coordinates to spherical coordinates.
- Scale θ and ϕ to H and $2H$, where H denotes the height of the image.
- Obtain the image pixel and put it into the big canvas. Repeat the steps from 1 to 5 given the time interval from 1 to T.

III. Technical Approach

A. Orientation Tracking

We first calibrate the readings of the IMU by subtracting the bias form in the beginning of the measurement, where the bias is calculated from the average of the first 100 readings. Besides, we transform the raw A/D values (readings from IMU) into physical units with the following equation:

$$\text{value} = (\text{raw} - \text{bias}) * 3300/(1023 * \text{sensitivity}) * \pi/180 \tag{5}$$

where we choose sensitivity 300 and 3.3 for linear acceleration and angular velocity respectively.

With only the motion model, we start with an initial quaternion $q_0 = [1, 0, 0, 0]$ and get the remaining q_t from the motion model. We then convert all the quaternions into Euler angles and then plot this with the Vicon rotation data to verify whether our IMU data was calibrated properly.

With both motion and observation models, we apply projected gradient descent to optimize the quaternion trajectory by the following equation:

$$q_{1:T}^{(t+1)} = q_{1:T}^{(t)} - 2\alpha^{(t)}\nabla c(q_{1:T}) \tag{6}$$

where $\alpha^{(k)}$ is the step size at timestep k.

B. Panorama

We first initialize a canvas with the size $[H, H*2, 3]$, where H denotes the height of the image. Then, we transform the image pixel into the Cartesian Coordinate by transforming the image pixel coordinates into FOV field, where we choose FOV Height = $\pi/4$, FOV Width = $\pi/3$. We then use the following equations to obtain the corresponding Cartesian coordinates.

$$X = \rho * sin(\phi) * cos(\theta) \tag{7}$$
$$Y = \rho * sin(\phi) * sin(\theta) \tag{8}$$
$$Z = \rho * cos(\phi) \tag{9}$$

We directly use the Vicon ground truth rotation matrix R to multiply with our new Cartesian coordinates. We then transform the new Cartesian coordinates to Spherical coordinates by using the following equations:

$$\phi = arccos(Z/1) \tag{10}$$
$$\theta = arctan(Y, X) \tag{11}$$

After that, we transform the angles to pixel coordinates with the following equations:

$$X = \theta * ratio \tag{12}$$
$$Y = \phi * ratio \tag{13}$$

where ratio equals to $\frac{H}{\pi}$.

Lastly, we use the closest-in-the-past timestamp of the Vicon ground truth to each camera image time stamp to align them.

IV. Results

A. Orientation Tracking

The following are the result of orientations from trainset IMU readings:

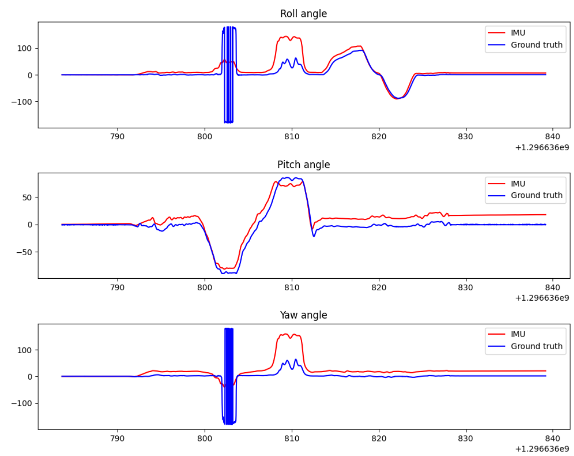

Fig. 1. IMU data 1

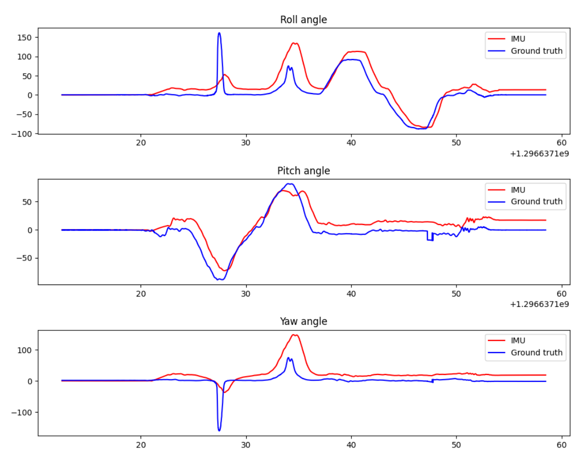

Fig. 2. IMU data 2

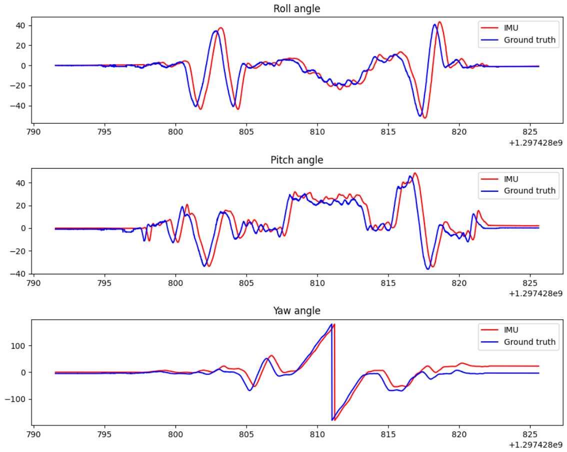

Fig. 3. IMU data 3

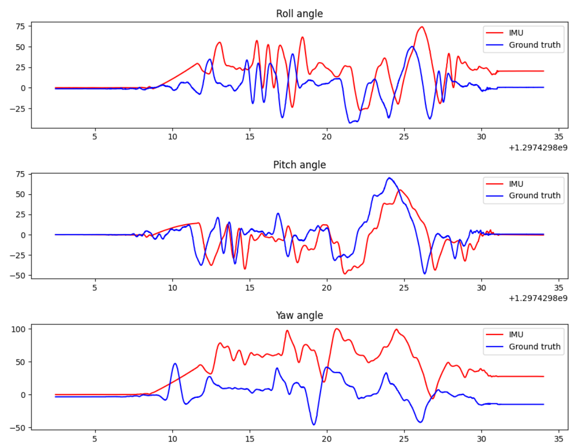

Fig. 4. IMU data 4

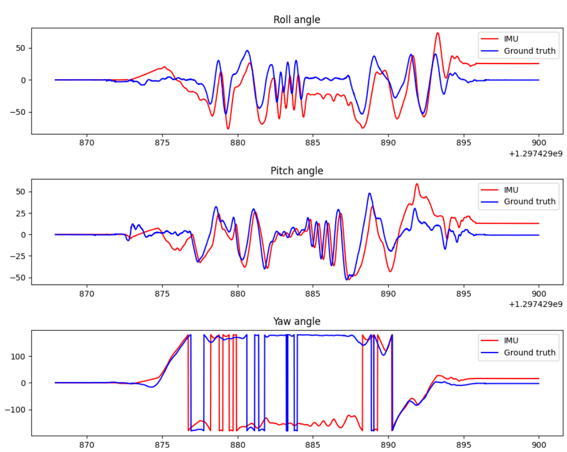

Fig. 5. IMU data 5

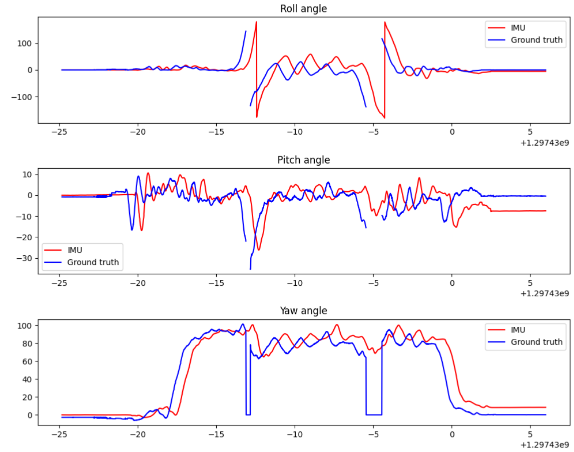

Fig. 6. IMU data 6

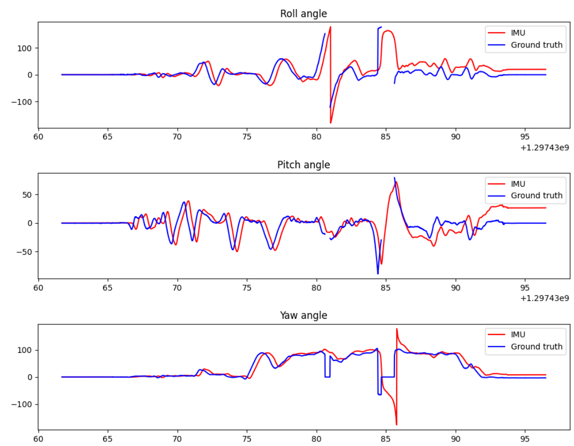

Fig. 7. IMU data 7

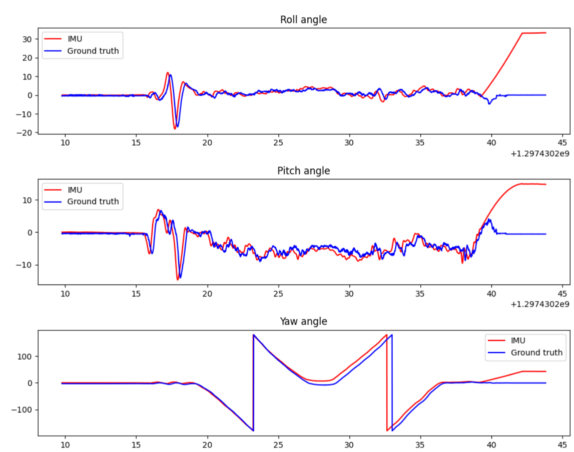

Fig. 8. IMU data 8

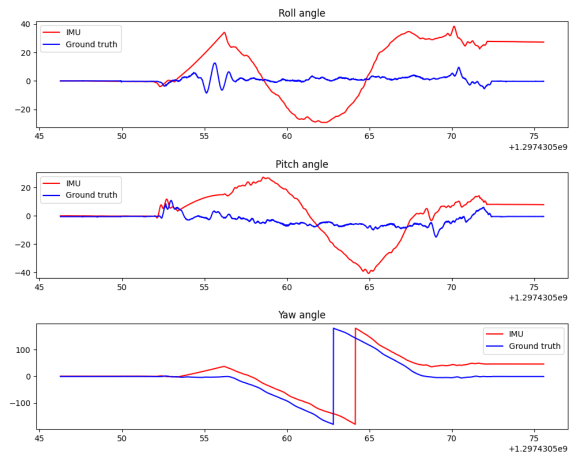

Fig. 9. IMU data 9

The following are the result of orientations from testset IMU readings:

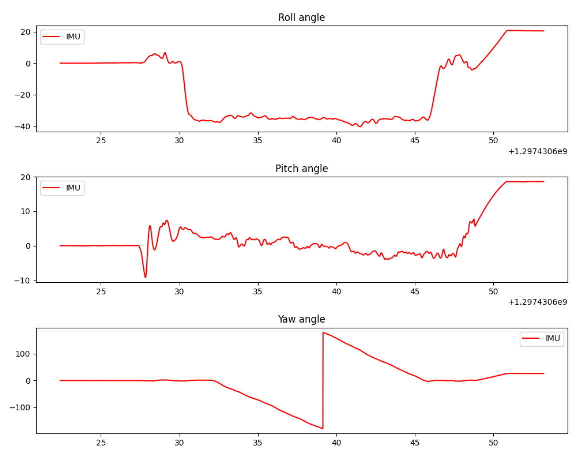

Fig. 10. IMU data 10

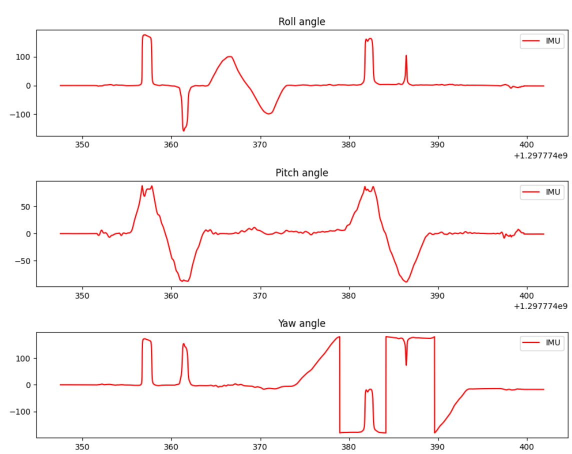

Fig. 11. IMU data 11

I bump into some problem in using jax to get the gradient of the cost function with respect to the quaternion trajectory, like we mentioned in (6). The figures above are generated using only the motion model; thus some of them do not align with the Vicon data well, but it is impressive to see that most of them have a fine approximation to the ground truth.

B. Panorama

Since I fail to get the optimization result in orientation tracking problem, I do not have the rotation information for the camera testing data; I just get the panorama of the training set. The following are the results of panorama from trainset camera images:

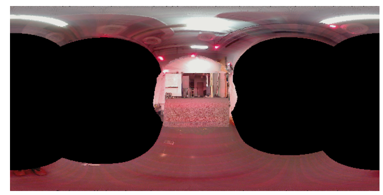

Fig. 12. CAM data 1

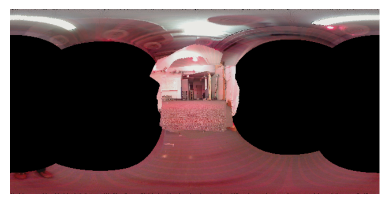

Fig. 13. CAM data 2

Fig. 14. CAM data 8

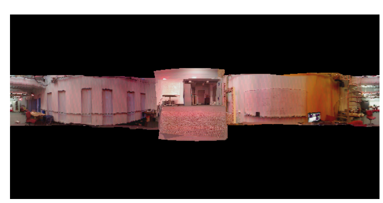

Fig. 15. CAM data 9