Orientation Tracking

Chao Qu[*]

University of Pennsylvania
quchao@seas.upenn.edu

I. Introduction

Orientation Tracking is another important yet interesting topic in robotics. In this project, we are given raw sensor data, and are asked to track the rotation of a camera and stitch its images using our tracking results. We will implement a Unscented Kalman filter with quaternion for the tracking orientation.

II. Pre-processing

The raw IMU data we get is 10-bit ADC values, which technically ranges from 0 to 1023. Since it's difficult to use those values directly in our filter, we have to convert them so that they have real-world physical units and meanings. Our goal here is to convert the readings from the micro-controller to acceleration with a unit of m/s^2 and angular velocity with a unit of rad/s. The equations used for converting raw values to real values are shown in Eq.1 and Eq.2.

$$\vec{a} = (\vec{a}_{raw} - \vec{a}_{bias}) \cdot s_a \tag{1}$$

$$\vec{\omega} = (\vec{\omega}_{raw} - \vec{\omega}_{bias}) \cdot s_\omega \tag{2}$$

where

$$\vec{a}_{bias} = \begin{bmatrix} 511.5 & 511.5 & 511.5 \end{bmatrix} \tag{3}$$

and $\vec{\omega}_{bias}$ is determined empirically by averaging the first 100 readings of one of the IMU data

$$\vec{\omega}_{bias} = \begin{bmatrix} 373.63 & 375.20 & 369.66 \end{bmatrix} \tag{4}$$

s_a and s_ω are the scale factors and their values are 0.0106 and 0.0171.

[*]A thank you or further information

We then look at the rotations in details.

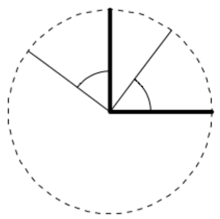

Figure 1: *A simple rotation along z-axis, with angle ψ*

Assume at some time t_0, the IMU is perfectly aligned with the world frame. In the next time step $t_0 + \Delta t$, there's a very small rotation along the z-axis, and the gyro thus measures an angular velocity ω_z, and thus an angle $\psi = \omega_z \cdot \Delta t$. This is shown in Fig.1. The rotation matrix of the rotation is denoted by $R_z(\psi)$ and it physically rotates the world SRT into the body SRT. Thus we call the rotation matrix *World-to-Body* and written in ${}^B R_W$. If there's rotation in all 3 axes, then we use a sequence of euler angles to indicate angular movement along each axis and this can be written as

$${}^B R_W = R_x(\phi) R_y(\theta) R_z(\psi) \tag{5}$$

where each of them is

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \tag{6}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \tag{7}$$

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8}$$

and we also notice that if the world coordinates and body coordinates of the gravity vector can be related by this rotation matrix

$$\begin{Bmatrix} 0 \\ 0 \\ g \end{Bmatrix} =^W R_B \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix} \tag{9}$$

where a_x, a_y and a_z are measurements by the accelerometer, where I use the convention that when the accelerometer is sitting rest, the reading would be $+1g$. The rotation matrix from vicon is ${}^W R_B$.

III. Methods

For the main unscented kalman filter algorithm, I basically followed the steps outlined in Kraft's paper [1]. For calculating quaternion mean, I used the method proposed by Cheon [2] instead of gradient descent. Another difference is that Kraft uses only $2n$ Sigma points with uniform weights, while Cheon uses $2n+1$ Sigma points with weights determined by parameters such as α, β and κ. Here I will describe the algorithm in detail. The following content is either from Kraft or from Cheon. It focuses more on the steps and implementations than the underlying theorems and principles, which serves the purpose for my personal review later.

I. Process Model

Process model in kalman filter takes the from

$$x_{k+1} = A(x_k, w_k) \tag{10}$$

where x is the state vector and w is the process noise with 0 mean and covariance Q. The first 3 components of w_k affect the orientation and the last three components of w_k affects angular velocity.

$$w_k = \begin{pmatrix} \vec{w}_q \\ \vec{w}_\omega \end{pmatrix} \tag{11}$$

The final process model is

$$x_{k+1} = A(x_k, w_k) = \begin{pmatrix} q_k q_w q_\Delta \\ \vec{\omega}_k + \vec{w}_\omega \end{pmatrix} \tag{12}$$

where q_k is the quaternion of the previous time step and q_w is the quaternion representing process noise

$$q_w = \left[\cos\left(\alpha_w/2\right) \quad \vec{e}_w \sin\left(\alpha_w/2\right)\right] \tag{13}$$

$$\alpha_w = \|w_q\|, \quad \vec{e}_w = \frac{w_q}{\|w_q\|} \tag{14}$$

and q_Δ is the quaternion representing the rotation during the interval of sensor update

$$q_\Delta = \left[\cos\left(\alpha_\Delta/2\right) \quad \vec{e}_\Delta \sin\left(\alpha_\Delta/2\right)\right] \tag{15}$$

$$\alpha_\Delta = \|\vec{\omega}_k\| \cdot \Delta t, \quad \vec{e}_\Delta = \frac{\vec{\omega}_k}{\|\vec{\omega}_k|} \tag{16}$$

II. Measurement Model

Measurement model in kalman filter takes the from

$$z_k = H(x_k, v_k) \tag{17}$$

Since we have gyro and accelerometer, we have two different type of measurement. The two measurement models are

$$H_1: \quad \vec{z}_{rot} = \vec{\omega}_k + \vec{v}_{rot} \tag{18}$$

$$H_2: \quad \vec{z}_{acc} = \vec{g}' + \vec{v}_{acc} \tag{19}$$

where

$$g' = q_k g q_k^{-1} \tag{20}$$

$$g = (0,\ [g_x, g_y, g_z]) \tag{21}$$

and $[g_x, g_y, g_z]$ is the acceleration readings from the IMU.

III. Initialization

At the initialization step, we first initialize the state vector to be

$$x_0 = \begin{pmatrix} q_0 \\ \omega_0 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top \tag{22}$$

because we roughly know that at the beginning the platform is stationary on the ground. Then we generates the weights $W^{(m)}$ and $W^{(c)}$ using the following parameters

$$\alpha = 0.5, \quad \beta = 2, \quad \kappa = 0 \tag{23}$$

Details for calculating the weights can be found in [3]. We then initialize the state covariance P, process covariance Q and measurement covariance R, all of them are 6×6 diagonal matrices.

IV. Prediction

At the prediction step, given the state vector x_k, state covariance P_k and process covariance Q_k, we calculate a set of $(2n+1)$ sigma points $\mathcal{X}_i$ using the following equations

$$\mathcal{X}_i = \begin{pmatrix} q_k & q_k q_{\mathcal{W}} \\ \vec{\omega}_k & \vec{\omega}_k + \vec{\omega}_{\mathcal{W}} \end{pmatrix} \tag{24}$$

in which

$$\mathcal{W}_i = \text{columns}(\pm\gamma\sqrt{P_k + Q_k}) \tag{25}$$
$$\gamma = \sqrt{n+\lambda}, \quad n = 6 \tag{26}$$
$$\lambda = \alpha^2(n+\kappa) - n \tag{27}$$
$$\mathcal{W}_{i,q} = \text{first 3 rows}(\mathcal{W}_i) \tag{28}$$
$$\mathcal{W}_{i,\omega} = \text{last 3 rows}(\mathcal{W}_i) = \vec{\omega}_{\mathcal{W}} \tag{29}$$

then

$$q_{\mathcal{W}} = \begin{bmatrix} \cos(\alpha_{\mathcal{W}}/2) & \vec{e}_{\mathcal{W}} \sin(\alpha_{\mathcal{W}}/2) \end{bmatrix} \tag{30}$$
$$\alpha_{\mathcal{W}} = \|\mathcal{W}_{i,q}\|, \quad \vec{e}_{\mathcal{W}} = \frac{\mathcal{W}_{i,q}}{\|\mathcal{W}_{i,q}\|} \tag{31}$$

We then transform the sigma points ahead of time through the process model

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0) \tag{32}$$

using the process model we described in previous section. The set $\{\mathcal{Y}_i\}$ samples the probability distribution of the *a priori* estimate. And the mean and covariance of this distribution are

$$\hat{x}_k^- = \text{mean}(\{\mathcal{Y}_i\}) \tag{33}$$
$$P_k^- = \text{cov}(\{\mathcal{Y}_i\}) \tag{34}$$

Calculating the mean of quaternions is a little bit tricky in this case. Instead of using a gradient descent method, we use a barycentric mean with renormalization which is proposed in [2]. The equation for calculating quaternion mean is

$$\hat{q}_k^- = \frac{\sum_{i=0}^{2n} W_i \mathcal{Y}_i^q}{\| \sum_{i=0}^{2n} W_i \mathcal{Y}_i^q \|} \tag{35}$$

This ensures that the result quaternion mean also lies on the unit quaternion sphere. We then need to calculate covariance P_k^-, this is done by

$$P_k^- = \sum_{i=0}^{2n} \mathcal{W}_i' \mathcal{W}_i'^\top \tag{36}$$

$\mathcal{W}_i'$ has a rotation vector component $\vec{r}_{\mathcal{W}'}$ and an angular velocity vector component $\vec{\omega}_{\mathcal{W}'}$

$$\mathcal{W}_i' = \begin{pmatrix} \vec{r}_{\mathcal{W}'} \\ \vec{\omega}_{\mathcal{W}'} \end{pmatrix} \tag{37}$$

$\vec{\omega}_{\mathcal{W}'}$ is the difference of the angular velocity components of $\mathcal{Y}_i$ and $\hat{x}_k^-$

$$\vec{\omega}_{\mathcal{W}'} = \vec{\omega}_i - \bar{\vec{\omega}} \tag{38}$$

$\vec{r}_{\mathcal{W}'}$ is a representation of the rotation which turns the orientation part of $\hat{x}_k^-$ into $\mathcal{Y}_i$. The corresponding quaternion of this rotation is

$$r_{\mathcal{W}'} = q_i \bar{q}^{-1} \tag{39}$$

V. Correction

At correction step, we transform a priori state sigma points $\mathcal{Y}_i$ through measurement model H to get measurement sigma points $\mathcal{Z}_i$. Since $\mathcal{Z}_i$ is just acceleration and angular velocity, we could get their mean by simply averaging them. The measurement estimate covariance is

$$P_{zz} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Z}_i - z_k^-][\mathcal{Z}_i - z_k^-]^\top \tag{40}$$

and innovation covariance

$$P_{vv} = P_{zz} + R \tag{41}$$

The cross correlation matrix is

$$P_{xz} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{W}'_i][\mathcal{Z}_i - z_k^-]^\top \tag{42}$$

With these covariance matices, we can finally calculate the Kalman gain and do the Kalman update.

$$K_k = P_{xz} P_{vv}^{-1} \tag{43}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \nu_k \tag{44}$$

Thus, we've finished one iteration of our Unscented Kalman filter for orientation estimation.

VI. Stitching

As for the image stitching part, I worked together with Tarik Tosun. We use a cylindrical coordinate system to project images on to a cylinder and unwrap the cylinder to get a panorama image. The idea is that you first find a vector that points from the optical center to a pixel in the image plane, you then rotate that vector into the world frame, using the rotation matrix you found either from your estimation or from vicon. Then you project the rotated vector onto a unit cylinder, and convert (X, Y, Z) to cylindrical coordinate $(\cos\theta, \sin\theta, h)$. Finally, you one that cylinder with some scaling to get the final image. We assume a unit cylinder that has z pointing along the cylinder. The equations we use are

$${}^{B}P = \begin{Bmatrix} X_B \\ Y_B \\ F \end{Bmatrix} \tag{45}$$

$${}^{W}P = {}^{W}R_B * \begin{Bmatrix} X_B \\ Y_B \\ F \end{Bmatrix} = \begin{Bmatrix} X_W \\ Y_W \\ Z_W \end{Bmatrix} \tag{46}$$

$$\theta = \arctan(\frac{Y_W}{X_W}) \tag{47}$$

$$h = \frac{Z_B}{\sqrt{X_W^2 + Y_W^2}} \tag{48}$$

$$(\hat{x}, \hat{y}) = -(f\theta, fh) + (\hat{x}_c, \hat{y}_c) \tag{49}$$

IV. Results

Here are some results from the dataset provided for training.

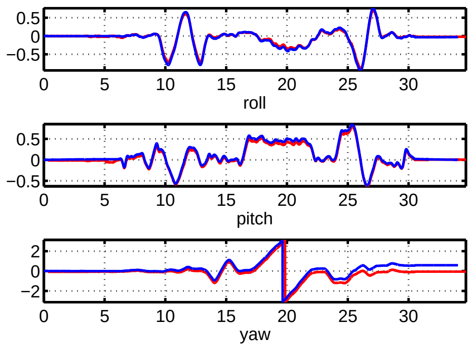

Figure 2: *Result for dataset 3, blue is estimation*

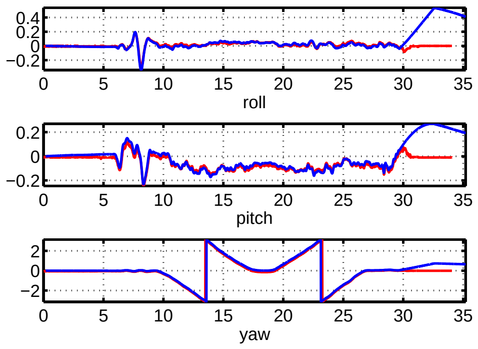

Figure 3: *Result for dataset8, blue is estimation*

and here's some stitching images.

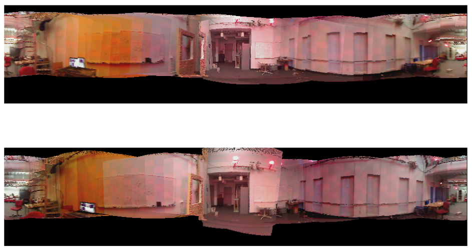

Figure 4: *Stitching for dataset 8 and 9*

V. Discussion

I also implemented a filter with different process and measurement model. The process model sees the gyro readings as input, and the measurement would just be acceleration. The state vector is also 7-dimensional, with 4 quaternion and 3 angular rate bias. This model seems to be perform better than the previous one.

References

[1] Edgar Kraft, *A quaternion-based unscented kalman filter for orientation tracking*. Information Fusion, 2003. Proceedings of the Sixth International Conference on Information Fusion, Vol. 1 (2003), pp. 47-54.

[2] Y.-J. Cheon and J.-H. Kim, *Unscented filtering in a unit quaternion space for spacecraft attitude estimation*. IEEE International Symposium on Industrial Electronics (ISIE 2007) (2007), pp. 66-71.

[3] S.J. Julier, and J.K. Uhlmann, *A new extension of the Kalman filter to nonlinear systems*. Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls, (1997).