Orientation Tracking

Chao Qu[*]

University of Pennsylvania
quchao@seas.upenn.edu

I. Introduction

Orientation Tgracking

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.

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

$$x_{k+1} = A(x_k, w_k) \tag{5}$$

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.

$$w_k = \begin{pmatrix} \vec{w}_q \\ \vec{w}_\omega \end{pmatrix} \tag{6}$$

The final process model is

$$x_{k+1} = A(x_k, w_k) = \begin{pmatrix} q_k q_w q_\Delta \\ \vec{\omega}_k + \vec{w}_\omega \end{pmatrix} \tag{7}$$

where q_k is the quaternion of the previous time step and q_w is the quaternion representing pro-

[*] A thank you or further information

cess noise

$$q_w = \begin{bmatrix}\cos(\alpha_w/2) & \vec{e}_w \sin(\alpha_w/2)\end{bmatrix} \quad (8)$$

$$\alpha_w = \|w_q\|, \quad \vec{e}_w = \frac{w_q}{\|w_q\|} \quad (9)$$

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

$$q_\Delta = \begin{bmatrix}\cos(\alpha_\Delta/2) & \vec{e}_\Delta \sin(\alpha_\Delta/2)\end{bmatrix} \quad (10)$$

$$\alpha_\Delta = \|\vec{\omega}_k\| \cdot \Delta t, \quad \vec{e}_\Delta = \frac{\vec{\omega}_k}{\|\vec{\omega}_k|} \quad (11)$$

II. Measurement Model

Measurement model in kalman filter takes the from

$$z_k = H(x_k, v_k) \quad (12)$$

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

$$H_1: \quad \vec{z}_{rot} = \vec{\omega}_k + \vec{v}_{rot} \quad (13)$$

$$H_2: \quad \vec{z}_{acc} = \vec{g'} + \vec{v}_{acc} \quad (14)$$

where

$$g' = q_k g q_k^{-1} \quad (15)$$

$$g = (0, \; [g_x, g_y, g_z]) \quad (16)$$

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

$$x_0 = \begin{pmatrix} q_0 \\ \omega_0 \end{pmatrix} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^\top \quad (17)$$

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

$$\alpha = 0.5, \quad \beta = 2, \quad \kappa = 0 \quad (18)$$

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

$$\mathcal{X}_i = \begin{pmatrix} q_k & q_k q_\mathcal{W} \\ \vec{\omega}_k & \vec{\omega}_k + \vec{\omega}_\mathcal{W} \end{pmatrix} \quad (19)$$

in which

$$\mathcal{W}_i = \text{columns}(\pm\gamma\sqrt{P_k + Q_k}) \quad (20)$$

$$\gamma = \sqrt{n + \lambda}, \quad n = 6 \quad (21)$$

$$\lambda = \alpha^2(n + \kappa) - n \quad (22)$$

$$\mathcal{W}_{i,q} = \text{first 3 rows}(\mathcal{W}_i) \quad (23)$$

$$\mathcal{W}_{i,\omega} = \text{last 3 rows}(\mathcal{W}_i) = \vec{\omega}_\mathcal{W} \quad (24)$$

then

$$q_\mathcal{W} = \begin{bmatrix}\cos(\alpha_\mathcal{W}/2) & \vec{e}_\mathcal{W} \sin(\alpha_\mathcal{W}/2)\end{bmatrix} \quad (25)$$

$$\alpha_\mathcal{W} = \|\mathcal{W}_{i,q}\|, \quad \vec{e}_\mathcal{W} = \frac{\mathcal{W}_{i,q}}{\|\mathcal{W}_{i,q}\|} \quad (26)$$

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

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0) \quad (27)$$

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

$$\hat{x}_k^- = \text{mean}(\{\mathcal{Y}_i\}) \quad (28)$$

$$P_k^- = \text{cov}(\{\mathcal{Y}_i\}) \quad (29)$$

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

$$\hat{q}_k^- = \frac{\sum_{i=0}^{2n} W_i \mathcal{Y}_i^q}{\| \sum_{i=0}^{2n} W_i \mathcal{Y}_i^q \|} \tag{30}$$

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

$$P_k^- = \sum_{i=0}^{2n} \mathcal{W}_i' \mathcal{W}_i'^\top \tag{31}$$

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

$$\mathcal{W}_i' = \begin{pmatrix} \vec{r}_{\mathcal{W}'} \\ \vec{\omega}_{\mathcal{W}'} \end{pmatrix} \tag{32}$$

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

$$\vec{\omega}_{\mathcal{W}'} = \vec{\omega}_i - \bar{\vec{\omega}} \tag{33}$$

$\vec{r}_{\mathcal{W}'}$ 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

$$r_{\mathcal{W}'} = q_i \bar{q}^{-1} \tag{34}$$

V. Correction

IV. Discussion

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).