Orientation Estimation

Varun Gupta and Nikolay A. Atanasov

Abstract— **Identifying the orientation of a camera has been an important problem for robots to construct an image of the environment in its exact form. This problem has been solved using Kalman Filter with IMU measurements. In this paper, we will discuss a variation of the kalman filter called the unscented kalman filter to estimate the orientation of the camera and compare the results with the vicon measurements. We will show how the Kalman filter helps improve the orientation estimation compared to the predictions from simple process model state transitions. We will also study the performance of the Kalman filter for different parameter settings and compare it with tuned parameter settings.**

I. INTRODUCTION

This project was accomplished to perform orientation estimation from IMU using a quarternion based Unscented Kalman Filter as a part of the course project in ESE 650: Learning in Robotics at University of Pennsylvania.

Orientation estimation has a wide range of applications in autonomous vehicle navigation, virtual reality and many other robotics applications. Here we study its importance in estimating the orientation of a camera in order to reconstruct a scene of the environment. We use a 9 degree of freedom MEMS strapdown inertial measurement unit which consists of an accelerometer, a gyroscope and a magnetometer. The accelerometers measure the accelerations along x, y, and z-axis in terms of gravitational acceleration units. The gyroscope provides the rotational velocity vector about the three orthogonal axes. This rotational velocity would be sufficient to estimate the orientation if the measurements were accurate. But in practice, it is observed that the error in the measurements grow with time due to quantization, integration and sensor errors. This necessitates the need for a correction algorithm that relies on useful inertial measurements. The third component of the IMU, that is the magnetometer, gives the change in magnetic field which does not influence the orientation for small distances, hence we do not use the magnetometer information in our model.

Kalman filter is a probabilistic tracking method that estimates the state of a discrete process using measurements obtained from reliable sensors. It assumes that the state is derived from a Gaussian distribution, the process model and the observation models are linear in the state and affected by Gaussian noise, and that the process and measurement noises are independent of each other. This filter does not work well with nonlinear process or measurement models and necessitates the need for an Extended Kalman Filter which uses first order Taylor series to approximate the nonlinear process and measurement models via linear functions and therefore enforce a Gaussian distribution on the posterior estimate of the state. Since EKF approximates the non-linear model to a linear model, it may not converge if the initial distributions are not right. Another version of the Kalman filter, called the Unscented Kalman filter avoids this by approximating the probability distribution by sampling a deterministic set of points which could represent the distribution. The state transitions based on the non-linear process and control models are propagated for these sample points in order to estimate the approximate distribution.

II. PROCEDURE

A. Data Preprocessing

For the purpose of orientation estimation, we use the accelerometer and gyroscope data from an Inertial Measurement Unit. The specifications of the sensor can be found in [3]. The information obtained from the sensor is in the form of 10 bit ADC values and is obtained at a frequency of 100Hz. The three acceleration components are measured as normal forces and due to the design of the IMU, the x and y components need to be reversed in direction to get the acceleration of the IMU in the inertial frame. To get the measurements in the desired units, the raw values from the sensor need to be subtracted by the bias and scaled by a factor that depends on the sensitivity of the sensor.

For the acceleration, the equation governing the conversion of units from ADC values to 'g' units is:

$$\alpha = \frac{(raw - bias) \times \dfrac{V_{ref}}{1023}}{sensitivity} \tag{1}$$

For the angular velocity, the equation governing the conversion of units from ADC values to 'rads/sec' is:

$$\omega = \frac{(raw - bias) \times \dfrac{V_{ref}}{1023}}{sensitivity} \times \frac{\pi}{180} \tag{2}$$

The bias is obtained by taking the mode of the data in the initial static case, that is, roughly the first 200 values. The V_{ref} is 3300 mV. The sensitivity of the accelerometer is 330 mV/g and the sensitivity of the gyroscope is 3.33 mV/deg/sec.

B. Algorithm

Here we present the algorithm followed to implement the quaternion based UKF for orientation estimation of the camera using acceleration and gyro measurements from an IMU. A quaternion representation is chosen because quaternions offer a singularity free description (as opposed to Euler angles) and rotations are computed more effectively compared to rotation matrices. The states are represented by a 4 dimensional quaternion instead of a 7 dimensional representation combining the orientation (4 dimensional quaternion) and the angular velocities (3 dimensional vector) as followed by Kraft. In this approach, the gyro measurements are treated as process control inputs which help in the prediction of the orientation following the process model equations; and the accelerations are treated as observations in the measurement model which are used to find the required update to the distribution of the state in order to reduce the error in the observations. The two steps of the filter, namely the prediction step and the update step, are detailed in Algorithm 1 and Algorithm 2 respectively.

1) Notation: The notations and conventions followed in this paper are the same as the UKF paper by Edgar Kraft. However certain dimensionality have been altered to formulate a simpler Kalman filter. x denotes the state of the system, that is, the orientation of the IMU. P denotes the error covariance matrix, Q denotes the process noise covariance and R denotes the measurement noise covariance. Here, Q and R are fixed and need to be tuned to get stable results. n indicates the degree of freedom of the state variable. n is 3 despite there being 4 state variables because the quaternion has inter-dependency in their components. ω denotes the angular velocity and α denotes the acceleration measurements obtained by the gyroscope and accelerometer respectively. The subscript k indicates the time instant. These are the major notations followed by the Kraft paper.

The algorithm uses a lot more notations to represent significant parameters computed during the filtering. X_i's denote the sigma points capturing the distribution. Y_i's denote the transformed sigma points after applying the process model. The quaternion average of the x_{k-1} weighed half and the 2n Y_i terms weighed $\frac{1}{4n}$ each gives the predicted state estimate $\hat{x}_k$. Z_i's denote the transformed sigma points after applying the measurement model and its average gives z_k. ν_k is called the innovation term which is basically the error between the acceleration observation and its estimate. $W_i^{'}$ denotes the error vectors in the state. P_{xz} is the cross correlation matrix, P_{zz} is the measurement covariance matrix. P_{vv} is the covariance of the innovation. K is the Kalman gain for the update term. More information about these terms can be found in Kraft's paper.

The algorithm involves a lot of quaternion operations such as multiplication, inverse, conjugate as well as conversion to and from rotation vectors. The quaternion is converted to a rotation vector by taking the logarithm of the quaternion; and the rotation vector is converted to a quaternion by exponentiation of the rotation vector. The weighted averaging of quaternions is performed using an iterative method with a certain acceptable error threshold.

Algorithm 1 Algorithm for prediction step of UKF

Input: ω_k, $\hat{P}_{k-1}$, $\hat{x}_{k-1}$
Output: $\hat{x}_k$, $\hat{P}_k$
Initialization : $Q \leftarrow k_q \times I_{3,3}$
$S = cholesky(P_{k-1} + Q)$
$W_i = columns(\pm\sqrt{2n}S)$
$q_\Delta = [cos(\mid \omega \mid \Delta t/2), sin(\mid \omega \mid \Delta t/2) \times \vec{\omega}]$
q_{k-1} = RotToQuat(x_{k-1})
for $each\ w_i\ in\ W_i$ **do**
  q_i = RotToQuat(w_i)
  X_i = QuatMultiply($q_{k-1} \times q_i$)
  Y_i = QuatMultiply($X_i \times q_\Delta$)
end for
$q_k = QuatAverage([x_{k-1}, columns(Y_i)], [\frac{1}{2}, \frac{1}{4n}])$
for $each\ y_i\ in\ Y_i$ **do**
  $W_k^{'}$ = QuatToRot($y_i \times$ QuatInv(q_k))
  $\hat{P}_k = \hat{P}_k + W_k^{'} \bigotimes W_k^{'T}$
end for
$\hat{x}_k$ = QuatToRot(q_k)
return $\hat{P}_k$, $\hat{x}_k$

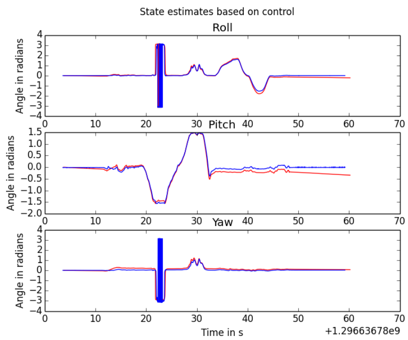
Algorithm 2 Algorithm for update step of UKF

Input: α_k, $\hat{P}_k$, $\hat{x}_k$, Y_i, $W_i^{'}$
Output: $\hat{x}_k$, $\hat{P}_k$
Initialization : $R \leftarrow k_r \times I_{3,3}$, $g \leftarrow [0, 0, 0, 1]$
q_k = RotToQuat(x_k)
for $each\ y_i\ in\ Y_i$ **do**
  q_i = RotToQuat(y_i)
  Z_i = QuatMultiply(QuatInv(q_i) $\times g \times q_i$)
end for
$\hat{z}_k = Average(Z_i)$
for $each\ z_i\ in\ Z_i$ **do**
  $P_{xz} = P_{xz} + W_i^{'} \bigotimes (Z_i - \hat{z}_k)^T$
  $P_{zz} = P_{zz} + (Z_i - \hat{z}_k) \bigotimes (Z_i - \hat{z}_k)^T$
end for
$P_{vv} = P_{zz} + R$
$K = P_{xz} P_{vv}^{-1}$
$\hat{P}_k = \hat{P}_k - K P_{vv} K^T$
$\nu_k = \alpha_k - z_k$
$\hat{x}_k$ = QuatMultiply(RotToQuat($K\nu_k/2$) $\times q_k$)
return $\hat{P}_k$, $\hat{x}_k$

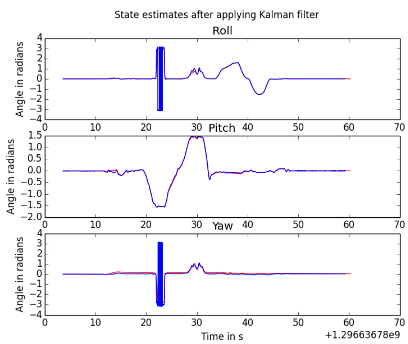
III. PERFORMANCE

The implementation of the Unscented Kalman filter uses a quaternion utility library developed by me. The performance of the Kalman filter has been compared with the ground truth obtained from VICON measurements for a set of 9 datasets. Certain parameters that needed to be tuned such as the noise covariance matrix and the measurement covariance matrix are tuned by experimentation but following a certain idea. The idea is that increasing the noise covariance values means that we are relying more on the acceleration values measurement for the estimation and decreasing means the opposite. Similarly increasing the measurement covariance values means that we are relying more on the gyroscope data for the estimation.

Fig1 shows two plots with subplots for roll, pitch and yaw orientations. The first plot in Fig.1 shows the accumulating error in case of orientation estimation based on the control

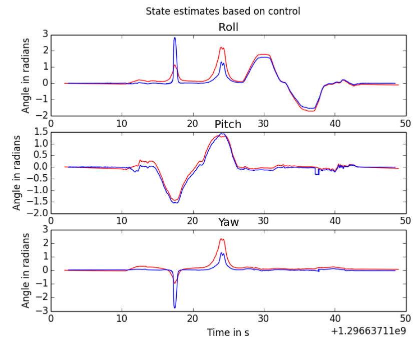


(a) State Estimate based on control

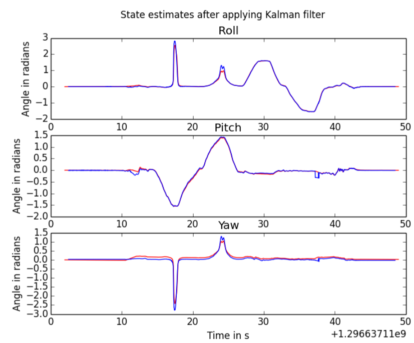


(b) UKF State Estimate

Fig. 1. IMU DATA 1



(a) State Estimate based on control



(b) UKF State Estimate

Fig. 2. IMU DATA 2

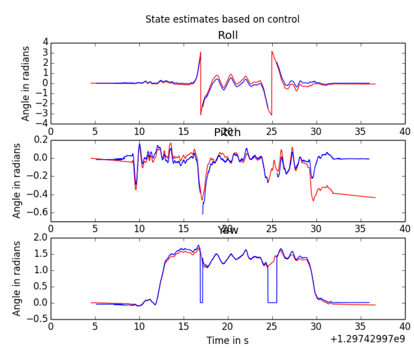
process only. The second plot shows the orientations estimated by the Unscented Kalman filter. The red curve depicts the predicted/estimated state and the blue curve depicts the true states by converting the quaternions obtained from VICON to Euler angles. The color code for the plots remains the same throughout the paper.

The first plot in Fig.2 shows the poor estimation in the absence of filter in case of abrupt changes in the angles and the second plot shows that the Kalman filter handles these cases. Fig.3 and Fig.4 show the shortcomings of the Kalman filter in estimating the yaw angle. In both these cases, the estimation of yaw is better without the filter whereas the estimation of pitch and roll improve tremendously with the addition of the filter.

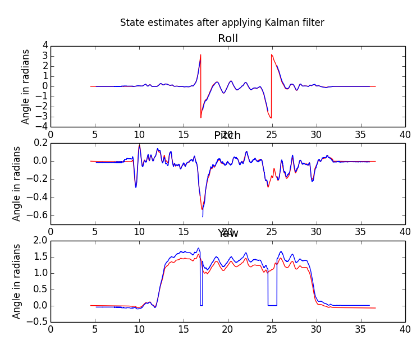
Fig.5 shows a case where a certain drift is introduced in the yaw estimates from the process model. Fig.6 shows the case where representation in quaternion helps in eliminating the problem of singularity. The rest of the figures show some of the test results for which the ground truth is not available but its performance can be evaluated by stitching together the images based on the estimated states.

IV. CONCLUSIONS

In this project, a quaternion based Unscented Kalman filter was implemented to estimate the orientation (roll, pitch and yaw) of a camera mounted with an IMU. A custom quaternion library was built to perform quaternion operations. The results show that the UKF improved the estimation of the orientation to almost match the ground

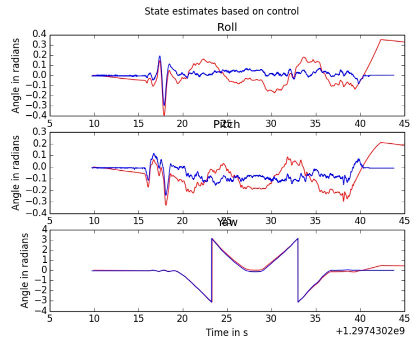


(a) State Estimate based on control

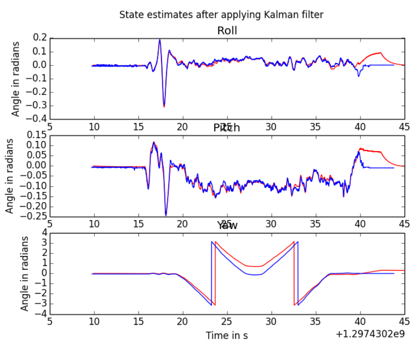


(b) UKF State Estimate

Fig. 3. IMU DATA 6

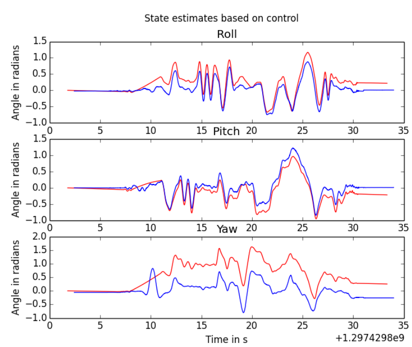


(a) State Estimate based on control

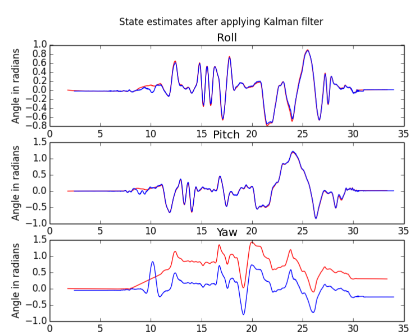


(b) UKF State Estimate

Fig. 4. IMU DATA 8

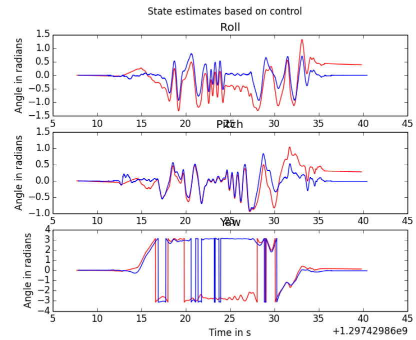


(a) State Estimate based on control

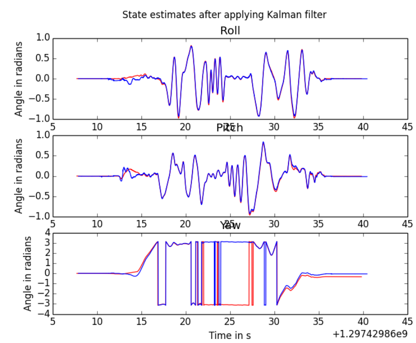


(b) UKF State Estimate

Fig. 5. IMU DATA 4

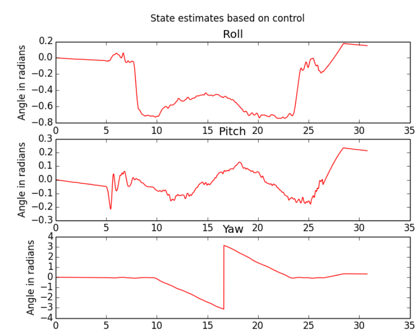


(a) State Estimate based on control

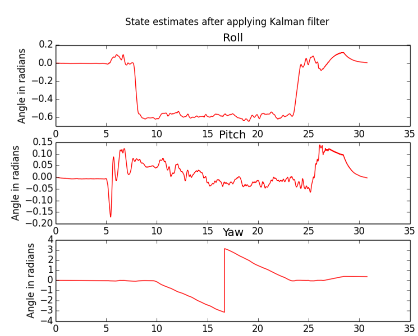


(b) UKF State Estimate

Fig. 6. IMU DATA 5

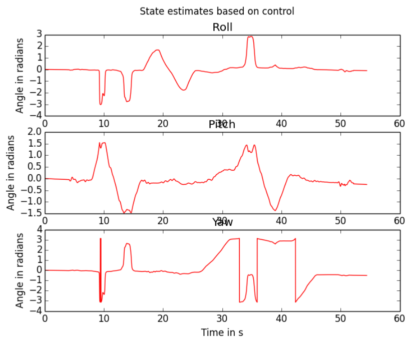


(a) State Estimate based on control

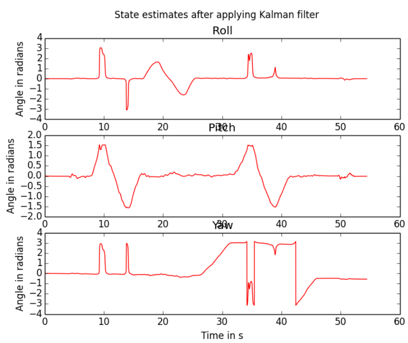


(b) UKF State Estimate

Fig. 7. IMU DATA 10

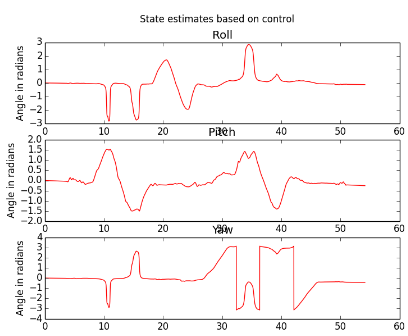


(a) State Estimate based on control

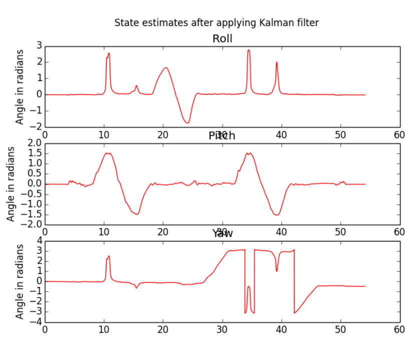


(b) UKF State Estimate

Fig. 8. IMU DATA 11

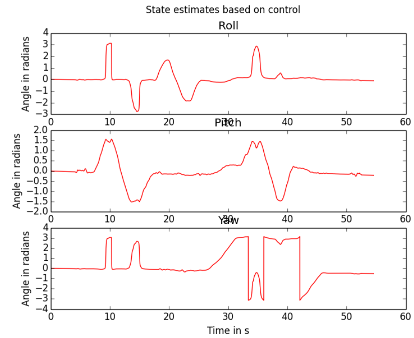


(a) State Estimate based on control

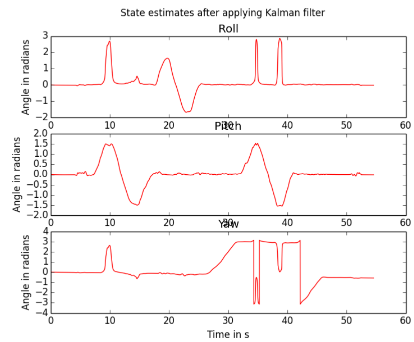


(b) UKF State Estimate

Fig. 9. IMU DATA 12



(a) State Estimate based on control



(b) UKF State Estimate

Fig. 10. IMU DATA 13

truth values for most cases. Several cases were discussed such as cases where there was a drift induced, or cases where only the control process gave better estimate which could be attributed to irregular accelerometer data. The parameters, namely the noise covariance matrix and the measurement covariance matrix were tuned by experimentation.

Further work is being done to stitch the images obtained by the camera following the orientation as estimated by the Kalman filter.

ACKNOWLEDGMENT

Thanks to University of Pennsylvania for providing me an opportunity to work on interesting projects by conducting a course on Learning in Robotics. Special thanks to Professor Nikolay A. Atanasov and the Teaching Assistants for providing the datasets and guiding me during the project. I would also like to thank my classmate Mustafa Hasekioglu for collaborating on estimating the bias of the sensors.

REFERENCES

[1] Edgar Kraft. A Quaternion Based Unscented Kalman Filter for Orientation Tracking.
[2] Nikolay A. Atanasov. Lecture Notes for ESE 650: Learning in Robotics, 2017.
[3] https://www.sparkfun.com/products/retired/9956