Orientation Estimation using IMU Data

Nicholas Ha, Nathan Godwin, Winson Quan
University of California, San Diego

Abstract

Orientation estimation is a problem with applications in many fields. Inertial measurement units present a convenient and inexpensive means to collect data on movements in a relative frame; translating these movements to an orientation in a global frame requires the use of sensor fusion algorithms. In this paper we compare the use of unscented Kalman filtering, extended Kalman filtering, and MadgwickAHRS for generating orientation information with respect to a global frame. Our results show that with a few exceptions, all methods track orientation well. Simple integration of gyro measurements does surprisingly well, and variations of the UKF typically do the best. In addition, UKF-based algorithms were proposed for estimating bias present in the sensor and to estimate the relative pose (position and orientation) between 2 IMUs. These algorithms were not successful, but other formulations may be investigated in the future.

1. Introduction

1.1. Background

Orientation estimation is a problem which is important to a variety of fields such as robotics, computer interaction, and autonomous navigation. Orientation measurements are relative to a certain frame of reference; since the majority of applications of orientation estimation are relative to a global frame of reference, rather than a sensor frame of reference, this paper focuses on estimation of orientation with respect to the global frame. Inertial measurement units (IMUs) present a cheap and effective means to collect data on relative rotational and translational motions. A 9 degree-of-freedom IMU generates data from an accelerometer, gyroscope, and magnetometer in three degrees of freedom per sensor. An accelerometer measures the acceleration of the sensor, including acceleration due to gravity, a gyroscope measures angular velocity, and a magnetometer measures the magnetic field.

Since the relative rotation of the sensor is provided by the angular velocity measurement from the gyroscope, a simple relative-frame orientation estimation can be generated by integrating the rotation measurements. However, small errors in the gyroscope measurements will accumulate over the course of the integration leading to an inaccurate estimation. Incorporating accelerometer data into the calculation allows relative orientation estimation algorithms to correct for rotational errors by using gravity as a reference in stationary measurement frames; however, in non-stationary measurement frames, sensor movement can produce acceleration in directions besides the direction of gravity, leading to errors in the perceived global reference direction. Therefore, any filter that generates an orientation estimation must either assume stationary measurements or correct for non-gravitational movement of the sensor.

While relative frame orientation can be determined by accelerometer and gyroscope data alone, the incorporation of magnetometer data is necessary to estimate orientation with respect to a global frame since it provides information on the sensor orientation with respect to magnetic north.

1.2. 3D Kinematics

Representations of orientation do not exist in a vector space; instead, they exist in a 3-dimensional special orientation group, SO(3), which contains all rotations about the origin of a three-dimensional Euclidean space. Operations in SO(3) are not the same as conventional vector space operations. Combining rotations is performed through composition, rather than addition, so it follows that differentiation, which relies on addition in a vector space, cannot be computed in the same way in SO3. Multiple ways exist to represent rotations, however each has advantages and drawbacks.

1.2.1 Axis-Angle Representation

One such way is axis-angle representation, which consists of a unit vector $\hat{\boldsymbol{u}}$, which describes the axis of rotation, and an angle θ which describes the amount of the rotation about $\hat{\boldsymbol{u}}$. Compositions between rotations are difficult in axis-angle representation and orientation is difficult to interpret, but axis-angles are useful for comparison between orientation estimation methods.

1.2.2 Euler Angle Representation

Another system for representing rotation is Euler angles which describe orientations as a composition of three rotations about the axes of a coordinate system. Euler angles are easier to interpret than axis-angles and easier to compose, however, Euler angles can suffer from gimbal lock, in which two axes become parallel, resulting in the loss of one degree of freedom.

1.2.3 Quaternion Representation

Quaternions are another means of representing rotation that are easy to compose and do not suffer from the flaws of Euler angle representation, however, quaternions are a 4-dimensional extension of the complex number system, represented in the form $\boldsymbol{q}$ = w + x$\boldsymbol{i}$ + y$\boldsymbol{j}$ + z$\boldsymbol{k}$ and cannot be easily interpreted. Rotation of a vector $\boldsymbol{u}$ in 3-dimensional Euclidean space using a quaternion $\boldsymbol{q}$ is performed by conjugating $\boldsymbol{u}$ by $\boldsymbol{q}$ such that

$$\boldsymbol{u'} = \boldsymbol{q}\boldsymbol{u}\boldsymbol{q}^{-1} \tag{1}$$

where $\boldsymbol{u}$ is assumed to have a real part, w, of 0 and $\boldsymbol{q}^{-1}$ = a - b$\boldsymbol{i}$ - c$\boldsymbol{j}$ - d$\boldsymbol{k}$. From this, it follows that composition of rotations with quaternions is simply the quaternion product. Quaternion products, using the operator $\otimes$, can be used to define a quaternion for a composition of rotations. The quaternion product is non-commutative and, for quaternions $\boldsymbol{p}$ and $\boldsymbol{q}$, is defined:

$$\boldsymbol{y} = \boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4 \\ p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3 \\ p_1q_3 - p_2q_4 + p_3q_1 + p_4q_2 \\ p_1q_4 + p_2q_3 - p_3q_2 + p_4q_1 \end{bmatrix} \tag{2}$$

Such that a rotation by the quaternion product $\boldsymbol{y}$ is equivalent to a composition of a rotation by $\boldsymbol{q}$ followed by a rotation by $\boldsymbol{p}$. In this paper, calculations are performed in Euler angles and quaternions and plots are represented in axis-angles.

1.3. Orientation Estimation Methods

In this paper, we examine the use of integration of angular velocity, an extended Kalman filter, an unscented Kalman filter, and MadgwickAHRS for sensor fusion and orientation estimation.

1.3.1 Angular Velocity Integration

Because the measurement of angular velocity at time t provides a relative change from the previous orientation, integrating over the angular velocity measurements will produce an estimate of the orientation relative to its starting point. However, measurement errors from the gyroscope will accumulate over the course of the integration, resulting in an inaccurate orientation estimate.

1.3.2 Kalman Filter

The Kalman filter estimates the state of a linear system [2]. However, because rotations are non-linear, two extensions to Kalman filtering were considered: the extended Kalman filter and the unscented Kalman filter.

1.3.3 Extended Kalman Filter

In the Extended Kalman Filter (EKF), state and measurement models do not need to describe a linear system and instead may describe a non-linear system with differentiable state and measurement models.

1.3.4 Unscented Kalman Filter

The classical Kalman filter propagates distribution mean and covariance through the fact that linear transformations of a jointly Gaussian distribution is also jointly Gaussian. The unscented Kalman filter propagates covariances by apply the state transition and measurement models to a set of representable points, called sigma points [3], [1]. The new means and covariances can then be computed by sampling these points.

1.3.5 MadgwickAHRS

MadgwickAHRS [5] is a commonly used sensor fusion algorithm for orientation estimation. Madgwick generates an estimate of the orientation from integration of the angular velocity which is corrected by a gradient descent problem incorporating accelerometer and magnetometer data. In his paper, Madgwick asserts that MadgwickAHRS produces lower error than Kalman filtering methods for orientation estimation and is therefore used for comparison to the EKF and UKF methods implemented in this paper.

2. Methods

2.1. Objective

The objective of this project is to estimate the orientation of an inertial measurement unit using sensor fusion algorithms to combine accelerometer, gyroscope, and magnetometer data.

2.2. Data Set

The data set used for this paper, titled T-Stick, was generated by Szczsna et al. [7] as a part of a reference data set, repoIMU, for measuring the accuracy of orientation estimation algorithms. The data was generated by attaching an IMU to a T-shaped mount and performing specific movements with it. Motion capture was used to generate ground truth data for each movement.

2.3. Extended Kalman Filter

2.3.1 Process Model

Equation 3 describes the process model, where $\boldsymbol{x}$ is the state and $\boldsymbol{w}$ is the noise. For a EKF, the Jacobians with respect to the state and noise has to be computed as in equations 4 and 5.

The state transition model for the extended Kalman filter assumes that the orientation at the next time step, $\boldsymbol{q}_{k+1}$, is the current orientation, $\boldsymbol{q}_k$, composed with a small change in orientation, $\boldsymbol{q}_\triangle$. The small change in orientation is measured as angular velocity by the gyroscope before being converted to a quaternion.

This produces a time varying state transition model because the angular velocity measured by the gyroscope changes with respect to time. Since the angular velocity is not kept track of in the state, its conversion into a quaternion does not need to be differentiated. Only the Jacobian of its relatively simple quaternion product needs to be computed, as shown in equation 4.

For the noise Jacobian, since a rotation has 3 degrees of freedom while a quaternion has 4 dimensions, the noise has to be converted from an axis-angle representation to a quaternion. The resulting Jacobian with respect to noise, as in equation 5, is slightly messy but manageable.

We also take the liberty of simply adding the noise to the state instead of computing the quaternion product. This avoids having to compute the Jacobian of the composition of a quaternion product with a quaternion conversion.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{w}_k) \qquad (3) \\ &= \boldsymbol{q}_k \otimes \boldsymbol{q}_\triangle + \left[\cos\left(\frac{\|\boldsymbol{w}\|}{2}\right), \boldsymbol{w}^T \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right) \right]^T \end{aligned}$$

$$A = \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{x}} = \begin{bmatrix} q_{\triangle w} & -q_{\triangle x} & -q_{\triangle y} & -q_{\triangle z} \\ q_{\triangle x} & q_{\triangle w} & q_{\triangle z} & -q_{\triangle y} \\ q_{\triangle y} & -q_{\triangle z} & q_{\triangle w} & q_{\triangle x} \\ q_{\triangle z} & q_{\triangle y} & -q_{\triangle x} & q_{\triangle w} \end{bmatrix} \qquad (4)$$

2.3.2 Measurement Model

The measurement model describes how the accelerometer would measure the gravity vector given an orientation of the IMU as estimated in the state. This model assumes that translational acceleration of the IMU is small compared to gravity. Since an acceleration of $9.8\frac{m}{s^2}$ is relatively fast, the assumption holds in many cases. The measured acceleration is the downward pointing gravity vector rotated by $\boldsymbol{q}_k$ as computed by quaternion conjugation using equation 1. The Jacobians are shown in equations 7 and 8.

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{v}_k) = \boldsymbol{q}_k^{-1} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g \end{bmatrix} \otimes \boldsymbol{q}_k + \boldsymbol{v}_k \qquad (6)$$

$$H = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} = 2g \begin{bmatrix} q_3 & -q_4 & q_1 & -q_2 \\ -q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & -q_4 \end{bmatrix} \qquad (7)$$

$$R = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{v}} = \boldsymbol{I} \qquad (8)$$

2.4. Unscented Kalman Filter

2.4.1 Unscented Transform

The Unscented Transform is used here to deal with nonlinearity of the process and measurement. This procedure represents a distribution as a set of 2d + 1 *sigma points*, where n is the number of dimensions in the state. For a Gaussian we track the mean and covariance to approximate the state distribution. The first sigma point is located at the mean of the distribution, and along each dimension, 2 of them lie on opposite sides of the mean, at a distance proportional to standard deviation in that direction. To get the standard deviation from the state covariance $\hat{\boldsymbol{P}}$ and the process noise covariance W we use the Cholesky decomposition of $\hat{\boldsymbol{P}} + \boldsymbol{W}$. The set of sigma points around $\hat{\boldsymbol{x}}$ is called X.

In our case, our state is a 4-dimensional quaternion with 3 degrees of freedom, so our covariances are represented as 3x3 matrix. In order to generate our sigma points, we turn our sigma points $\boldsymbol{X}$ into axis angle form, add the noise, and convert them back into quaternion before going to the next step.

2.4.2 Prediction Step

After sigma points $\boldsymbol{X}$ are created, they are sent through the process model individually to create the set $\boldsymbol{Y}$. The process model is as described above in the EKF, equation 3. Since we're using the UKF we don't need the Jacobians in equation 4 and equation 5. This allows us to keep the nonlinearity of the true process model. However we still need the prior estimates and covariance, $\boldsymbol{x}_{ap}$ and $\boldsymbol{P}_{ap}$.

$$Q = \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{-w_1 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{-w_2 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{-w_3 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} \\ \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right) + \frac{w_2^2 \cos\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{w_2 w_1 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{w_3 w_1 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} \\ \frac{w_1 w_2 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right) + \frac{w_2^2 \cos\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{w_3 w_2 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} \\ \frac{w_1 w_3 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \frac{w_2 w_3 \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} & \sin\left(\frac{\|\boldsymbol{w}\|}{2}\right) + \frac{w_3^2 \cos\left(\frac{\|\boldsymbol{w}\|}{2}\right)}{2\|\boldsymbol{w}\|} \end{bmatrix} \tag{5}$$

The weighted mean of this set is then $\boldsymbol{x}_{ap}$, and the weighted covariance is $\boldsymbol{x}_{ap}$. Since angles, and therefore orientations, are periodic, we can't simply compute their sum and take an average. We need to do an iterative procedure to compute these two quantities. We guess the average, compute the error from each vector to the average, and find the average error. We rotate the guess of the mean by this average error and repeat. Once the norm of the average error is small enough, we terminate, and the guess of the average is considered close enough to the true average. The covariance $\boldsymbol{x}_{ap}$ is computed from the set of error vectors in axis-angle form.

2.4.3 Update Step

After the set $\boldsymbol{Y}$ produced during the prediction step, we pass each of these sigma points through the the measurement model, equation 6, to generate the set $\boldsymbol{Z}$. These are the expected measurements that would be obtained from each of the sigma points. We take the mean and covariance of these to get $\boldsymbol{z}_{ap}$ and $\boldsymbol{P}_{zz}$, and these can be computed in a standard fashion since they are vectors.

Next, we compute the cross correlation $\boldsymbol{P}_{xz}$, which relates noise in state to noise in the measurement. Next are:

$$\boldsymbol{P}_{nu} = \boldsymbol{P}_{zz} + \boldsymbol{R} \tag{9}$$

$$\boldsymbol{K} = \boldsymbol{P}_{xz} * \boldsymbol{P}_{nu}^{-1} \tag{10}$$

Finally we can complete the update step as:

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_{ap} \otimes \text{aa2quat}(\boldsymbol{K} * nu) \tag{11}$$

$$\hat{\boldsymbol{P}} = \boldsymbol{P}_{ap} - K * P_{nu} * \boldsymbol{K}' \tag{12}$$

where aa2quat is an axis-angle to quaternion conversion.

2.4.4 Alternative Formulation: 7 dimensional state

We can also take the state to be orientation along with angular velocity, and assume that angular velocity stays constant in between steps. Then we update the angular velocity during the update step, with the measurement model being the identity. A similar EKF with a 7 dimensional state was not implemented due to the difficulty of taking a Jacobian of a quaternion product composed with a axis-angle to quaternion conversion.

2.5. MadgwickAHRS

MadgwickAHRS produces an estimate of the quaternion for the orientation of the global frame relative to the sensor frame using an integration of the angular velocity which is corrected by a gradient descent problem using accelerometer and magnetometer. The final orientation estimation is shown in equations 13-16,

$$\hat{\boldsymbol{q}}_{est,t} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} \tag{13}$$

$$\boldsymbol{q} = \int \dot{\boldsymbol{q}}_{est,t} \tag{14}$$

$$\dot{\boldsymbol{q}}_{est,t} = \frac{1}{2}\hat{\boldsymbol{q}}_{est,t-1} \otimes s_{\omega_t} - \beta \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \tag{15}$$

$$\nabla \boldsymbol{f} = {\boldsymbol{J}_{g,b}}^T(\hat{\boldsymbol{q}}_{est,t-1}) f_{g,b}(\hat{\boldsymbol{q}}_{est,t-1}, s_{\hat{\boldsymbol{a}}}, s_{\hat{\boldsymbol{m}}}) \tag{16}$$

where $\hat{\boldsymbol{q}}_{est,t}$ is the estimated orientation at time t, $\hat{\boldsymbol{q}}_{est,t-1}$ is the previous estimated orientation at time t-1, s_{ω_t} is a 4-dimensional angular velocity vector at time t with a 0 real component, $s_{\hat{\boldsymbol{m}}}$ is a 4-dimensional magnetic field vector at time t with a 0 real component, $s_{\hat{\boldsymbol{a}}}$ is a 4-dimensional acceleration vector at time t with a 0 real component, $f_{g,b}$ is the solution surface of the gradient descent algorithm, ${J_{g,b}}^T$ is its Jacobian, and $\beta = \sqrt{\frac{3}{4}}\omega_\beta$, where ω_β is normalized gyroscope error for each axis. For the gyroscope used in this paper, we use a value of $\beta = 0.003$. The full Jacobian and solution surface equations are fully derived by Madgwick in [5]. A MATLAB library for MadgwickAHRS [5] was used to implement MadgwickAHRS.

2.6. Performance factors

2.6.1 Biases

Kalman filters all assume zero mean, white Gaussian noise. White Gaussian noise is usually a reasonable assumption, but for cheap IMUs, biases may be present in the measurements. These are typically modeled as random walk processes [6]. Biases are typically fixed, but there are components that may be time varying due to conditions like temperature.

If stationary data is available, we can estimate the biases by assuming that on average, angular velocity should be zero mean, and acceleration be the gravity vector, taken as [0 0 9.81].

2.6.2 Noise Covariances

The Kalman filters rely on noise covariances Q and R, where Q represents uncertainty in the process model and R represents uncertainty in the measurement. In the 4D EKF and UKF, Q is gyroscope noise and some level of uncertainty from non-constant angular velocity. R is acceleration noise (along with magnetometer noise when it is available). In the 7D UKF, Q represents only the uncertainty from non-constant angular velocity, and R represents noise in all measurements. In theory these IMU noises can be found by looking at stationary data. However, we found better value of Q and R through a grid search.

2.6.3 Use of Magnetometer

The Kalman Filters mentioned above do not use the magnetometer at all. They use acceleration in the update step with gravity as the reference to the global frame. However, multiple yaw angles will produce the same measurement, meaning yaw is unobservable from acceleration measurements alone. A magnetometer gives relative field strengths in 3 axes, so it can use magnetic north as the reference. We expect incorporating magnetometer data into the Kalman filter measurement will prevent the yaw from drifting over time.

We implement a 4D UKF with magnetometer, with the an unchanged process model and modified measurement model. The measurement model for the magnetometer is that same as the model for the accelerometer, but the gravity vector is replaced with the magnetic north vector. We take this as the first magnetometer measurement in the data set.

2.7. Unsuccessful Methods

2.7.1 Bias estimation

In some applications, stationary data is not available. We propose a filter which estimates the accelerometer bias b_a and/or gyro bias b_g by modifying the process and measurement models and including b_a and b_g in the state. For both biases, the process model is zero-mean random walk. The models are shown in equations 17-20. However, implementations of these models did not converge to reasonable biases.

$$a_m = g + b_a \tag{17}$$

$$g_m = w + b_g \tag{18}$$

$$b_a^{(t+1)} = b_a^{(t)} \tag{19}$$

$$b_g^{(t+1)} = b_g^{(t)} \tag{20}$$

2.7.2 System Identification for Relative Pose

A method based on the UKF was proposed to estimate relative pose (position and orientation) between two IMUs fixed to a rigid body by using accelerometer and gyro measurements from both. While the complete formulation is not shown here, the fiter tracked the relative pose as the state. The process model assumed that rigid body rotated around one IMU, and the measurement model gave the expected measurement of the second IMU. The measurement model required knowledge of orientation with respect to the global frame as well as angular acceleration, which we determined by pre-processing the data using one of the filters above. Then our ”system identification UKF” was run to estimate the relative pose. When implemented, the filter produced non-positive definite covariance matrices which prohibited generation of sigma points and caused this filter to fail.

3. Experiments

Estimated orientations are displayed as the components of the axis-angle representation. The axis-angle representation lends itself for comparison, as the magnitude of each component is proportional to the magnitude of rotation. This is in contrast to quaternions which always have a norm of 1.

3.1. Different Filters

The following filters are compared: a simple integrator, EKF, UKF, EKF with magnetometer, UKF with magnetometer, UKF with 7 dimensional state (UKF7), and MadgwickAHRS. The EKF and UKF both follow the model equations 3 and 6. The UKF with a 7 dimensional state is described in section 2.4.4.

3.2. Process and Measurement Noise Covariances

Using measured noise covariance from a stationary sensor measurement, we got roughly *diag*(7e-5,6e-5,7e-5) and *diag*(3e-5,3e-5,3e-5) for $\boldsymbol{Q}$ and $\boldsymbol{R}$. However, these produced poor results. In order to determine the best values for Q and R, we run a parameter search where:

$$\boldsymbol{Q} = \alpha \boldsymbol{I}_3 \tag{21}$$

$$\boldsymbol{R} = \beta \boldsymbol{I}_3 \tag{22}$$

$$\tag{23}$$

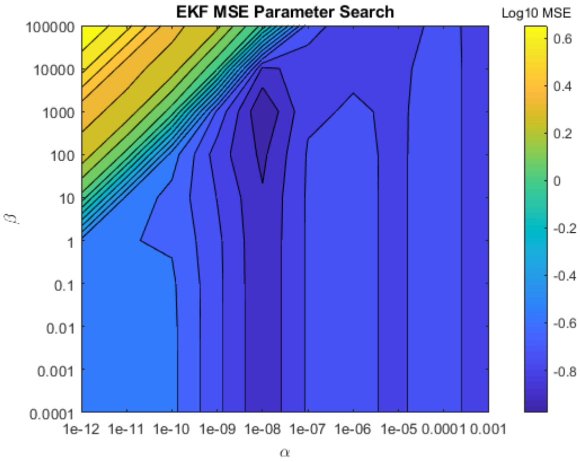

Figure 1. A parameter search over Q and R matrices produces $\alpha = 1 \times 10^{-8}$ and $\beta = 1 \times 10^{3}$ for EKF.

where $\boldsymbol{I}_3$ is the 3x3 identity matrix, α is a scalar, and β is a scalar. α and β perform a logarithmic sweep over the ranges [1e-12, 1e-3] and [1e-4, 1e5] respectively.

3.2.1 EKF

Performing the parameter search for the EKF produces the plot shown in figure 1. The optimal values of α and β are 1×10^{-8} and 1×10^{3} corresponding to the process and measurement noise, respectively. This indicates that the gyroscope in the process model is trusted much more than the accelerometer in the measurement model.

3.2.2 UKF

Since the UKF follows the same process and measurement model as the EKF, we would expect the optimal values of α and β to be the same. However, running the same parameter search yields the plot in figure 2. For a given ratio of α or β, the minimum error gives a constant ratio of α or β. The filter seem to be not sensitive to the absolute magnitude of the covariances. The relative magnitude of the covariances are very large, as in the EKF.

The optimal values of α and β are 1×10^{-12} and 1×10^{-4} for the process and measurement noise respectively.

3.2.3 UKF with 7 Dimensional State

The first 3 dimensions of the process covariance correspond to the change in orientation while the latter 3 dimensions of correspond to the constant angular velocity model. Similary, the noise covariance has dimensions corresponding to the accerlerometer and gyroscope. This presents a minimum of 4 degrees of freedom for the two covariance matrices, making a extensive parameter search infeasible. Instead the covariance matrices were found heuristically by sweeping across one degree of freedom at a time. The process noise covariance matrix is $10^{-5} \times diag(10, 10, 1, 1, 1, 1)$ and the measurement noise covariance matrix is $10^{2} \times \boldsymbol{I}_6$.

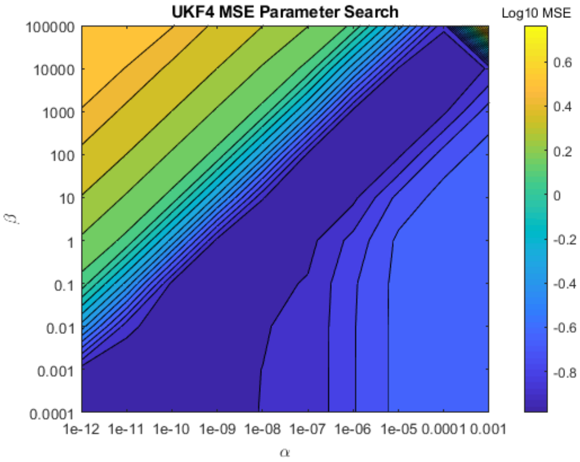

Figure 2. A parameter search over Q and R matrices produces $\alpha = 1 \times 10^{-12}$ and $\beta = 1 \times 10^{-4}$ for UKF.

3.3. Effect of Magnetometer

The addition of a magnetometer in the measurement model as described in section 2.6.3 almost eliminates the drift in yaw in the data set without motion, as seen in figure 3. The effect is not as pronounced in dynamic data set as filters without the magnetometer generally do not drift. The addition of the magnetometer causes a slight penalty in the high frequency response of the filter.

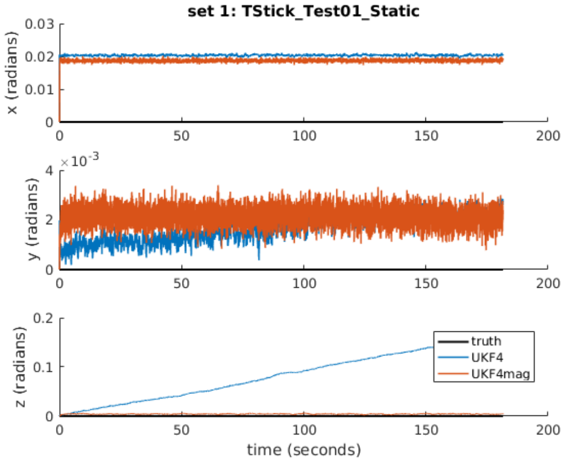

Figure 3. UKF with (orange) and without (blue) magnetometer

3.4. Effect of Removing Bias

For most filters, removing bias in gyro and accelerometer measurements improved performance, particularly in data sets with slow motion. Figure 4 shows results for each filter on the stationary data set without preprocessing to remove bias. Figure 5 shows the same results, but with preprocessing. Without preprocessing, the only filter that stays on track in the z direction is the UKF with magnetometer. After removing bias, they all track well except for the Madgwick filter in which x and y are the same, but z gets worse.

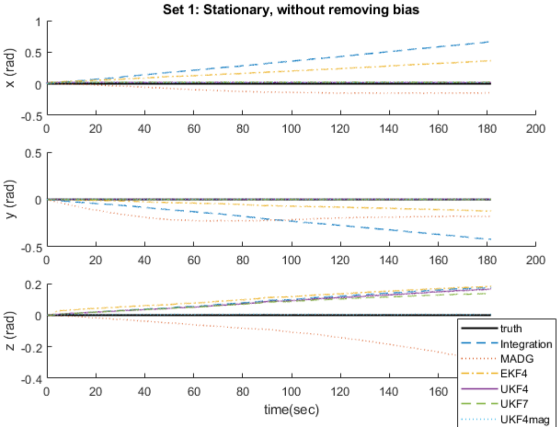

Figure 4. Stationary data set without removing bias.

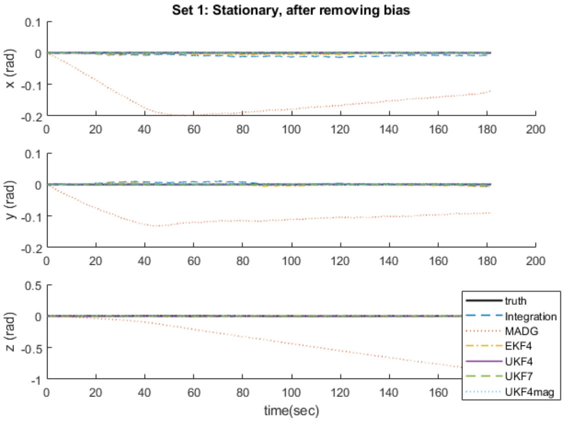

Figure 5. Stationary data set after removing bias.

For other well-behaved data sets, the bias removal had little effect qualitatively. Figure 6 shows free rotation for each filter, without removing bias. All filters track orientation quite well.

3.5. Overall Results

A summary of mean square error with different algorithms and representative data sets are in table 1. In general the UKF performed the best. Most methods perform similarly well most of the time. The difference in methods are usually too small to see, as in figure 6. However, some methods are more computationally expensive than others.

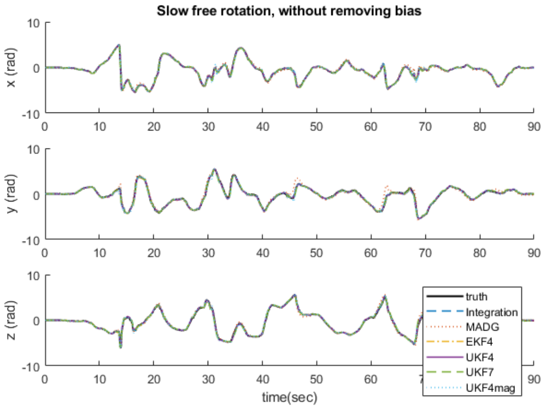

Figure 6. Slow free rotation without removing bias

In particular, the UKF takes more time to run than the other filters, which agrees with the results of LaViola [4] and conflicts with the assertion of Kraft [3]. The UKF with a 7-dimensional state also takes much longer to run than the 4-dimensional state UKF. The UKF with magnetometer does not usually perform better than a UKF without magnetometer, despite having more available information. It does perform better in data sets with slower motion and worse in data set with faster motion.

Surprisingly, the simple integrator performs comparably with the other methods. This is reasonable as the EKF and UKF perform an integration in their process model and their process covariances are very low. However, on longer timescales, the integrator would drift whereas the Kalman filters would remain referenced by gravity.

Two data sets were particularly difficult in that they produced poor results for some filters. Tracking for ”fast rotation about X” was lost for all filters, though UKF with magnetometer followed it for the longest as in figure 7. It is evident from the graph that this rotation was faster than ”fast rotation about Z” in figure 8. The sheer speed and acceleration of the rotation is likely the cause of failure.

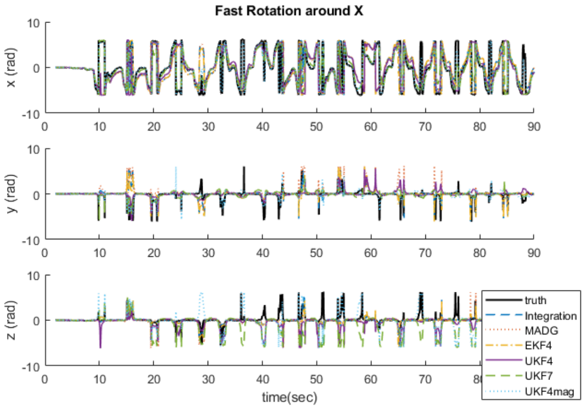

Figure 7. Fast Rotation around X

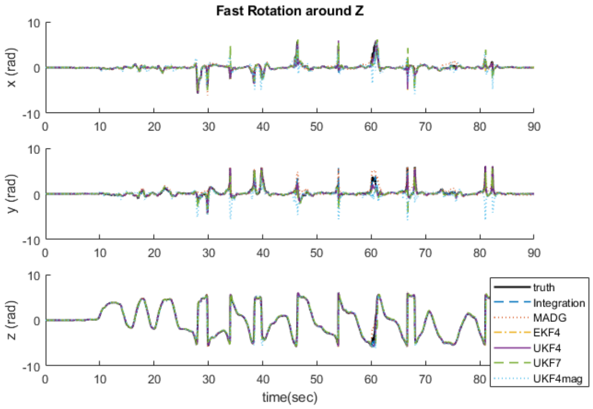

Figure 8. Fast Rotation around Z

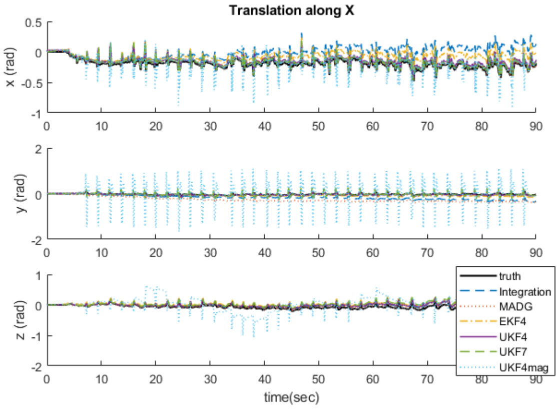

Figure 9. Translation along X

Another challenging data set was ”translation along X”, as shown in 7. Since it is only translation, most filters correctly show only has small deviations in orientation. However, the UKF with magnetometer has serious periodic errors, though it recovers to tracking correctly each time. This may be due to the fact that since the most complicated filter that uses the most data, it's requires more assumptions about the model. The measurement model has the assumption that there is no translation. The other filters do surprisingly well despite the data set violating the model.

4. Conclusion

We tried several different versions of nonlinear extensions of Kalman filtering on several datasets with different types of motion. By examining MSE and looking at plots of axis-angle, we conclude that all methods worked relatively well. Overall, variations of the UKF performed better for most situations. Factors that improved performance include tuning Q and R (surprisingly, to use the gyroscope much more than the accelerometer), preprocessing data to remove bias, and using the magnetometer (for some cases). Since most filters perform similarly in most cases, robustness and computational efficiency become more important factors for practical use.

References

[1] S. J. Julier and J. K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[2] R. E. Kalman et al. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 5(2):102–119, 1960.

[3] E. Kraft. A quaternion-based unscented kalman filter for orientation tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 1, pages 47–54, 2003.

[4] J. J. LaViola. A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440. IEEE, 2003.

[5] S. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25, 2010.

[6] F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1156, Oct 2008.

[7] A. Szczesna, P. Skurowski, P. Pruszowski, D. Peszor, M. Paszkuta, and K. Wojciechowski. Reference data set for accuracy evaluation of orientation estimation algorithms for inertial motion capture systems. In *International Conference on Computer Vision and Graphics*, pages 509–520. Springer, 2016.

Data Set Algorithm w/o Hard Bias	Stationary	Slow Rotation Around X	Slow Rotation Around Z	Fast Rotation Around X	Fast Rotation Around Z	Translation Through X	Free Motion
Integration	7.08×10^{-2}	5.42×10^{-1}	1.32×10^{-1}	5.24	4.52×10^{-2}	2.42×10^{-2}	1.00×10^{-2}
Madgwick	2.28×10^{-2}	8.20×10^{-1}	7.77×10^{-1}	5.44	2.28×10^{-1}	2.20×10^{-2}	1.36×10^{-1}
EKF	2.12×10^{-2}	4.07×10^{-1}	2.91×10^{-1}	5.50	7.47×10^{-2}	1.08×10^{-2}	7.29×10^{-3}
UKF	3.22×10^{-3}	1.01×10^{-1}	2.10×10^{-1}	5.85	6.41×10^{-2}	3.06×10^{-3}	6.22×10^{-3}
UKF7	2.67×10^{-3}	3.28×10^{-2}	1.13×10^{-1}	4.52	3.03×10^{-1}	7.26×10^{-3}	1.43×10^{-2}
UKF w/ mag	1.21×10^{-4}	3.72×10^{-2}	1.78×10^{-1}	1.84	6.70×10^{-1}	8.13×10^{-2}	2.46×10^{-2}
Algorithm w/ Hard Bias							
Integration	3.64×10^{-5}	1.63×10^{-2}	1.48×10^{-2}	4.79	1.67×10^{-2}	7.53×10^{-4}	8.58×10^{-3}
Madgwick	8.86×10^{-2}	6.98×10^{-1}	7.73×10^{-1}	5.04	2.19×10^{-1}	1.45×10^{-2}	1.17×10^{-1}
EKF	1.18×10^{-5}	2.95×10^{-2}	7.27×10^{-2}	5.48	2.97×10^{-2}	4.01×10^{-4}	8.59×10^{-3}
UKF	5.03×10^{-6}	2.06×10^{-2}	1.01×10^{-1}	5.69	1.54×10^{-2}	1.22×10^{-4}	5.62×10^{-3}
UKF7	1.66×10^{-6}	4.91×10^{-3}	3.63×10^{-2}	4.57	2.03×10^{-1}	3.15×10^{-3}	1.32×10^{-2}
UKF w/ mag	8.40×10^{-6}	1.46×10^{-2}	2.63×10^{-1}	6.64×10^{-1}	7.40×10^{-1}	1.00×10^{-1}	2.35×10^{-2}

Table 1. Mean squared error for each algorithm. Highlighted cells indicate the best performing algorithm per column.