Unscented Kalman Filter for 3D Attitude Estimation 16.322 Final Project

Matthew Vernacchia
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Email: mvernacc@mit.edu

***Abstract*—A 3D attitude estimator based on the Unscented Kalman Filter (UKF) using a MEMS IMU and magnetometer is developed. The estimator uses a quaternion representation of attitude, and issues with applying the UKF to quaternion state are explored. The characterization of the sensors' noise, bias and bias walk is also documented.**

I. Introduction

The estimation of an object's 3D pose (rotation and translation) relative to a global reference frame is an important problem in robotics and aerospace engineering. This estimation task is called navigation. Satellites, launch vehicles, manned aircraft and UAVs need pose information in order to control their trajectories and to point sensors, communications equipment and weapons at targets. In this project, I restrict the navigation task to estimating rotation only.

Two broad classes of sensors are used in navigation: proprioceptive and exteroceptive [1], [2]. Proprioceptive sensors measure quantities internal to the robot or vehicle; exteroceptive senors measure the surrounding world.

Accelerometers and rate gyroscopes are common proprioceptive sensors; they measure the acceleration and angular velocity of the vehicle body. The sensors detect the effect of non-inertial fictitious forces on the motion of a small system (oscillating mass, spinning wheel, etc.) within the sensor. Typically, three orthogonal gyroscopes and three orthogonal accelerometers are combined into a single device, called a strapdown Inertial Measurement Unit (IMU) [3].

In articulated robots or vehicles, proprioceptive sensors are also used to sense joint angles and actuator positions.

Navigation can be performed by integrating IMU measurements. However, noise and sensor biases cause the integrated estimate to drift over time. Depending on the quality of the sensors, the estimate becomes useless after tens of seconds (MEMS devices) to several hours (navigation-grade IMUs) [3].

To correct for drift, exteroceptive measurements are incorporated into the estimate. Exteroceptive navigation sensors include:

- *Field sensors* measure orientation relative to a vector field which has a known direction in the global frame. Magnetometers measure the Earth's magnetic field, and accelerometers can measure the gravitational field (on a non-free-falling vehicle). Note that these sensors cannot detect rotation about the vector, e.g. a gravitational sensor cannot provide yaw/heading information.
- *Beacon/Landmark sensors* measure the range and/or bearing to a set of beacons. Examples include Sun sensors [4], GPS/GNSS, and LORAN.
- *Mapping sensors* sense nearby visual or shape features and match them to a map of the environment. Cameras, LiDAR, and ultrasound rangefinders have been used for mapping. The map can be known a priori, or the estimator can perform simultaneous localization and mapping (SLAM) [5].
- *Odometry sensors* measure the incremental motion of the vehicle relative to the environment. On wheeled vehicles, encoders are used to count wheel revolutions. *Visual odometry* uses optical flow algorithms to compute motion from camera data [6].

In this project, I use an IMU and magnetometer.

Several estimators are used in navigation, generally depending on which sensors are available. Non-linear Kalman filters are the typical choice for systems with field, beacon and odometry sensors. Kalman filters work well for systems with approximately Gaussian sensor noise, which these sensors usually have. Mapping sensors require more sophisticated estimators because their measurement distributions are non-Gaussian (e.g. an environment with several similar-looking regions will produce a multi-modal distribution), particle filters are often used in this case.

In this project, I will use an Unscented Kalman filter (UKF). I choose the UKF over the Extended Kalman filter because it offers superior covariance estimation, and avoids the need to determine the Jacobians of the state transition and measurement functions.

II. Hardware

For my sensors, I use an MPU-9150 from InvenSense [7]. SparkFun Electronics sells the sensor conveniently

mounted on a breakout board (part number SEN-11486). The MPU-9150 contains a 3-axis MEMS rate gyro, 3-axis MEMS accelerometer, and a 3-axis Hall effect magnetometer.

For computation I use a BeagleBone Black single board computer from Texas Instruments. The BeagleBone communicates with the sensor board via and Inter-Integrated Circuit (I^2C) bus.

III. Coordinate Systems

I reference the following coordinate systems in my project. The notation for vectors is $\vec{r}^A$ indicates that a vector is written in frame A. The notation for quaternions is q_{A2B} indicates the rotation which transforms vectors from frame A to B.

$$\vec{r}^B = \textsc{RotateFrame}(\vec{r}^A, q_{A2B}) \tag{1}$$

A. North East Down (NED)

The North-East-Down frame is an inertial, Cartesian, right-handed coordinate system.

- The origin is at the location of the sensor (I am only considering rotations, not translations of the sensor).
- The x axis lies in the horizontal plane (i.e. locally tangent to the WGS84 ellipsoid) and points towards geographic (not magnetic) North.
- The z axis points towards the center of the Earth.
- The y axis completes the right-handed system.

B. Sensor

The sensor frame is a non-inertial, Cartesian, right-handed coordinate system. The estimation task is to determine the rotation of the sensor frame relative to the NED frame

IV. Sensor Models

A. Accelerometer

I model the MEMS accelerometer measurement function as

$$\vec{a}^{sensor}_{meas}(t) = \textsc{RotateFrame}(-\vec{g}^{NED}_{Earth}, q_{NED2sensor}(t)) + \vec{\nu}_{accel}(t) \tag{2}$$

where:

- $\vec{a}^{sensor}_{meas}(t)$ is the acceleration measured in the sensor frame at time t.
- $\vec{g}^{NED}_{Earth} = [0, 0, 9.81\,\mathrm{m\,s^{-2}}]$ is the Earth's gravitational field in the NED frame.
- $q_{NED2sensor}(t)$ is the rotation from the NED frame to the sensor frame at time t.
- $\nu_{accel}(t)$ is the accelerometer measurement noise.

Compared to the magnitude of the acceleration due to gravity, the bias of MEMS accelerometers is typically quite small [7], [3]. Therefore I do not include bias terms in my accelerometer model. Further, I assume that the sensor only undergoes rotation, not translation. Any accidental accelerations (e.g. shaking of the sensor during rotation, centripetal acceleration) are modeled as noise. I mounted the sensor near the body's center of mass to minimize centripetal acceleration.

The measurement noise is drawn from

$$\vec{\nu}_{accel} \sim \mathcal{N}(0, \begin{bmatrix} \sigma^2_{accel,d,x} & 0 & 0 \\ 0 & \sigma^2_{accel,d,y} & 0 \\ 0 & 0 & \sigma^2_{accel,d,z} \end{bmatrix}) \tag{3}$$

The measurement noise standard deviation is typically specified as a continuous-time Fourier spectral density $\sigma_{accel,c,*}$, with units of $\mathrm{ms^{-2}}\sqrt{\mathrm{s}}$. This can be converted to discrete time via:

$$\sigma_{accel,d,*} = \sigma_{accel,c,*}\frac{1}{\sqrt{\Delta t}} \tag{4}$$

where Δt is the sampling time interval [8].

B. Rate Gyroscope

I model the MEMS rate gyroscope measurement function as [8]:

$$\vec{\omega}^{sensor}_{meas}(t) = \vec{\omega}^{sensor}(t) + \vec{d} + \vec{b}_{gyro}(t) + \vec{\nu}_{gyro}(t) \tag{5}$$

where:

- $\vec{\omega}^{sensor}_{meas}(t)$ is the angular rate measured in the sensor frame at time t.
- $\vec{\omega}^{sensor}(t)$ is the true angular rate of the sensor with respect to an inertial frame, at time t.
- $\vec{d}$ is a constant bias.
- $\vec{b}_{gyro}(t)$ is a time-evolving bias.
- $\nu_{gyro}(t)$ is the gyro measurement noise.

I model the evolution of the bias as a stationary Gauss-Markov process [8], [9]:

$$\frac{\mathrm{d}\vec{b}_{gyro}}{\mathrm{d}t} = -\alpha\vec{b}_{gyro}(t) + \vec{\eta_{u,c}}(t) \tag{6}$$

where α is the inverse of the correlation time, and $\vec{\eta_{u,c}}(t)$ is the zero mean Gaussian bias walk noise. The discrete-time version of this equation is [8]:

$$\vec{b}_{gyro}(t + \Delta t) = (1 - \alpha\Delta t)\vec{b}_{gyro}(t) + \vec{\eta_{u,d}}(t) \tag{7}$$

The sensor's correlation time and bias walk noise standard deviation are estimated in section V.

C. Magnetometer

I model the magnetometer measurement function as [10]:

$$\vec{h}^{sensor}_{meas}(t) = (I+D)^{-1}(\textsc{RotateFrame}(\vec{h}^{NED}_{Earth}, q_{NED2sensor}(t)) + \vec{b}_{mag}) + \vec{\nu}_{mag}(t) \tag{8}$$

where:

- $\vec{h}^{sensor}_{meas}(t)$ is the magnetic field measured in the sensor frame at time t.
- $\vec{h}^{NED}_{Earth}$ is the Earth's magnetic field in the NED frame. This value varies with time and geographic location. In Cambridge, MA in 2015, it is $\vec{h}^{NED}_{Earth} = [19.5\,\mu\mathrm{T}, -5.1\,\mu\mathrm{T}, 48.2\,\mu\mathrm{T}]$ [11].
- $\vec{b}_{mag}$ is a constant bias which rotates with the sensor.

- D is a constant, symmetric 3×3 matrix. Its diagonal entries correspond to scale factor errors, and its off-diagonal entries to non-orthogonality errors.
- $\vec{\nu}_{mag}(t)$ is the magnetometer measurement noise.

Although the bias and scale factor terms are assumed to be constant in time, they depend on the environment around the magnetometer. Soft iron distortions (nearby ferromagnetic, low magnetic coercivity materials), distort the Earth's magnetic field and change the value of D. Hard iron distortions (nearby permanent magnets or electric currents) add to the Earth's magnetic field and change the value of $\vec{b}_{mag}$ [12]. It is important to locate the magnetometer away from time-varying magnetic sources (e.g. motors). It is best to perform the magnetometer calibration steps described in section V-B after the magnetometer has been installed.

V. Sensor Characterization

A. Allan Variance

I analyzed the noise properties of my sensors using the Allan variance technique [3]. The Allan variance of a signal is computed according to algorithm 1. The Allan variance is computed for a range of averaging times τ, typically 10 times the measurement interval to 0.1 times the duration of the signal. The Allan deviation ($\sqrt{\text{AVAR}(\tau)}$) is then plotted against τ on a log-log scale.

Algorithm 1 Allan Variance

function AVAR(τ)
  Divide the data into n bins of time length τ
  for each bin b_i **do**
    $a_i \leftarrow \text{MEAN}(b_i)$
  end for
  return $\frac{1}{2(n-1)} \sum_{i=1}^{n-1} (a_{i+1} - a_i)^2$
end function

The resulting Allan deviation plot has the general shape shown in figure 1. Given the time scales of my measurements, I expect to capture the white noise and bias stability regions. The Allan deviation plots for my sensors are shown in figures 2 to 4.

The values of the sensor's noise parameters can be read from the Allan Deviation plots. The continuous-time measurement noise Fourier spectral density $\sigma_{*,c,*}$ is the value at which the linear -0.5 slope portion of the plot passes through $\tau = 1\,\text{s}$ [3]. I perform a linear best fit on the Allan Deviation for $\tau \in [0.1\,\text{s}, 10\,\text{s}]$, and find the value of the best-fit line at $\tau = 1\,\text{s}$. For comparison, I also compute the discrete-time measurement noise $\sigma_{*,d,*}$ by taking the standard deviation of the signal. The results of both calculations are presented in table I. For all sensors, the continuous- and discrete-time noise estimates are in agreement, i.e. equation 4 is approximately correct. The accelerometer noise parameters are close to the manufacturer's specified values, while the gyro is 2 to 4 times nosier that the specification. The manufacturer does not specify noise parameters

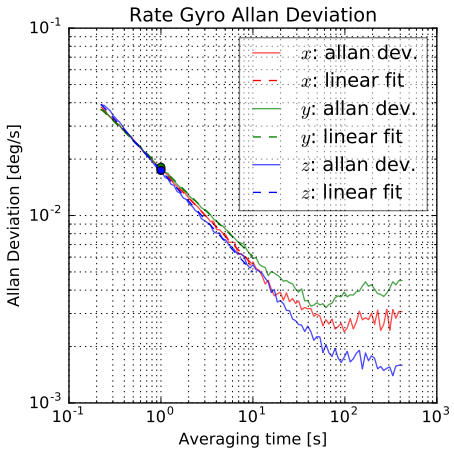

Fig. 2. Allan deviation analysis of MPU-9150 rate gyroscope data.

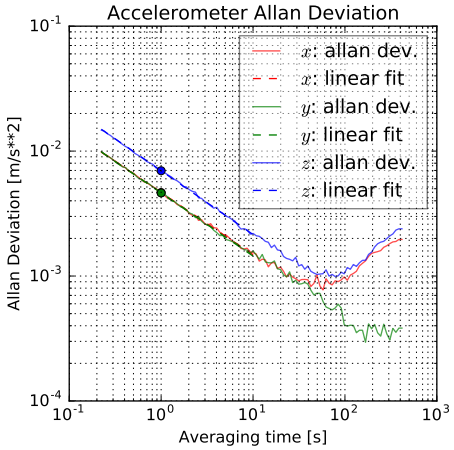

Fig. 3. Allan deviation analysis of MPU-9150 accelerometer data.

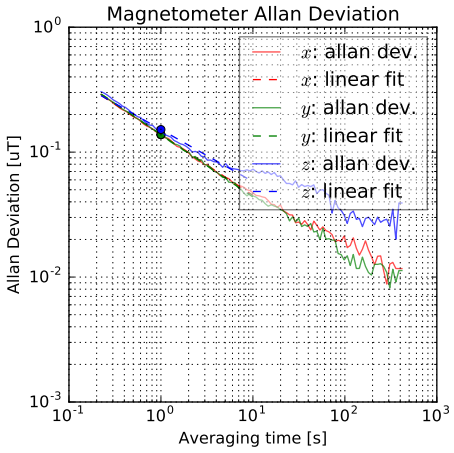

Fig. 4. Allan deviation analysis of MPU-9150 magnetometer data.

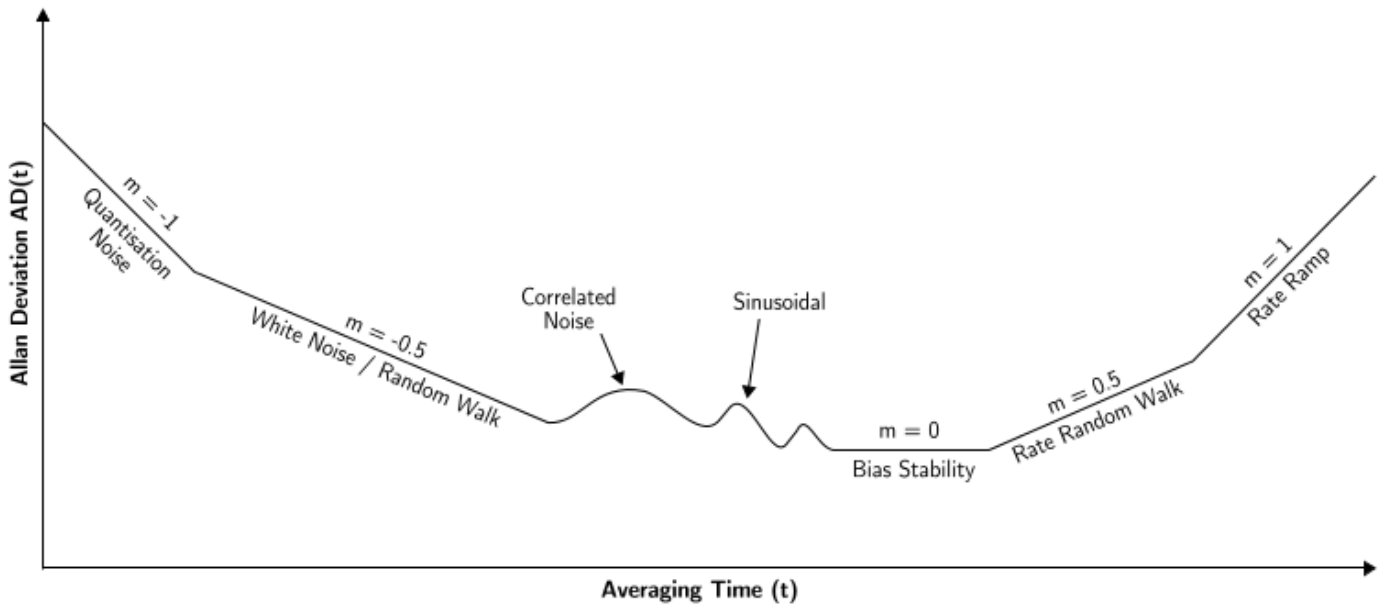

Fig. 1. The typical regions of a $\log(\text{ADEV}(\tau))$ vs τ plot, and their relation to signal noise characteristics. Reprinted from [3].

for the magnetometer, but my values are about 9 times the values reported for a similar MEMS Hall effect sensor in [13].

For sensor with bias walk, the Allan Deviation plot will have a flat 0 slope region. The signal's bias instability $\sigma_{*,bi,*}$ is the minimum value of ADEV, and the correlation time $1/\alpha$ is the value of τ at which the minimum occurs [3]. The continuous-time bias walk noise $\vec{\eta_u}(t)$ has standard deviation [8]:

$$\text{Var}[\eta_{*,u,c,*}] = \sigma^2_{*,u,c,*} = 2\alpha\sigma^2_{*,bi,*} \tag{9}$$

This is converted to discrete time by [8]:

$$\text{Var}[\eta_{*,u,d,*}] = \sigma^2_{*,u,d,*} = (\Delta t)(\sigma^2_{*,u,c,*}) \tag{10}$$

The results of the bias walk analysis are presented in table II. The accelerometer and gyro Allan Deviation plots have a flat region, indicating that a bias walk process exists for these sensors. The magnetometer Allan Deviation plot does not have a flat region, and continues at a slope of ≈ -0.5 for all observed τ. This indicates that the magnetometer does not have a bias walk process, only a white noise process. This confirms my assumption that the magnetometer bias is independent of time. The accelerometer bias is very small ($\sigma_{accel,u,c,*} \approx 10^{-4}$) compared to the quantity to be measured ($|\vec{g}_{Earth}| \approx 10$), justifying my decision to neglect bias in the accelerometer measurement model.

TABLE II
SENSOR BIAS WALK PARAMETERS

Sensor	Bias Instability	Correlation Time
MPU-9150 Gyro x axis	$2.40 \times 10^{-3} \circ \text{s}^{-1}$	98.3 s
MPU-9150 Gyro y axis	$3.23 \times 10^{-3} \circ \text{s}^{-1}$	62.3 s
MPU-9150 Gyro z axis	$1.40 \times 10^{-3} \circ \text{s}^{-1}$	332.0 s
MPU-9150 Accel x axis	$7.74 \times 10^{-4}\,\text{m s}^{-2}$	57.7 s
MPU-9150 Accel y axis	$2.95 \times 10^{-4}\,\text{m s}^{-2}$	167.4 s
MPU-9150 Accel z axis	$9.91 \times 10^{-4}\,\text{m s}^{-2}$	78.2 s
MPU-9150 Mag	-	> 400 s

B. Magnetometer calibration

Calibrating the magnetometer entails estimation of the parameters $\vec{b}_{mag}$ and D. Both approaches require knowledge of the magnitude Earth's local magnetic field $|\vec{h}_{Earth}|$, and do not require knowledge of attitude. Because rotation preserves magnitude, all measurements of the Earth's field should have magnitude $|\vec{h}_{Earth}|$ if the proper calibration factors are applied. The magnitude errors between each of the calibrated measurements and $|\vec{h}_{Earth}|$ can be used as a cost to find the optimal $\vec{b}_{mag}$ and D [10].

The calibration parameters can either be estimated from a batch of data prior to running the attitude estimator, or they can be estimated in real time as new magnetometer measurements arrive. One *a priori* algorithm uses non-linear least squares to minimize the sum of the squares of the magnitude errors. A real-time algorithm uses a modification of the UKF measurement-update equations [10].

The calibration process can be visualized as mapping a 3D ellipsoid to a sphere. If we plot the set of uncalibrated measurements as points in 3-space, they lie roughly on the surface of an ellipsoid. Calibration finds the translation ($\vec{b}_{mag}$)

TABLE I
SENSOR MEASUREMENT NOISE PARAMETERS. SPECIFIED PARAMETERS FROM [7] UNLESS OTHERWISE NOTED.

Sensor	FFT noise (meas)	FFT noise (spec)	RMS noise (meas)	RMS noise (spec)
MPU-9150 Gyro x axis	$0.0179\,^\circ/\sqrt{\text{s}}$	$0.005\,^\circ/\sqrt{\text{s}}$	$0.125\,^\circ\,\text{s}^{-1}$ at 46 Hz	$0.06\,^\circ\,\text{s}^{-1}$ at 92 Hz
MPU-9150 Gyro y axis	$0.0162\,^\circ/\sqrt{\text{s}}$	$0.005\,^\circ/\sqrt{\text{s}}$	$0.106\,^\circ\,\text{s}^{-1}$ at 46 Hz	$0.06\,^\circ\,\text{s}^{-1}$ at 92 Hz
MPU-9150 Gyro z axis	$0.0180\,^\circ/\sqrt{\text{s}}$	$0.005\,^\circ/\sqrt{\text{s}}$	$0.141\,^\circ\,\text{s}^{-1}$ at 46 Hz	$0.06\,^\circ\,\text{s}^{-1}$ at 92 Hz
MPU-9150 Accel x axis	$4.56 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$3.9 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$0.0315\,\text{m}\,\text{s}^{-2}$ at 46 Hz	$0.039\,\text{m}\,\text{s}^{-2}$ at 92 Hz
MPU-9150 Accel y axis	$4.63 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$3.9 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$0.0315\,\text{m}\,\text{s}^{-2}$ at 46 Hz	$0.039\,\text{m}\,\text{s}^{-2}$ at 92 Hz
MPU-9150 Accel z axis	$6.66 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$3.9 \times 10^{-3}\,\text{ms}^{-2}\sqrt{\text{s}}$	$0.0453\,\text{m}\,\text{s}^{-2}$ at 46 Hz	$0.039\,\text{m}\,\text{s}^{-2}$ at 92 Hz
MPU-9150 Mag x axis	$0.136\,\mu\text{T}\sqrt{\text{s}}$	Unspec'd	$0.913\,\mu\text{T}$ at 46 Hz	$0.1\,\mu\text{T}$ at 100 Hz [13]
MPU-9150 Mag y axis	$0.143\,\mu\text{T}\sqrt{\text{s}}$	Unspec'd	$0.923\,\mu\text{T}$ at 46 Hz	$0.1\,\mu\text{T}$ at 100 Hz [13]
MPU-9150 Mag x axis	$0.151\,\mu\text{T}\sqrt{\text{s}}$	Unspec'd	$0.983\,\mu\text{T}$ at 46 Hz	$0.1\,\mu\text{T}$ at 100 Hz [13]

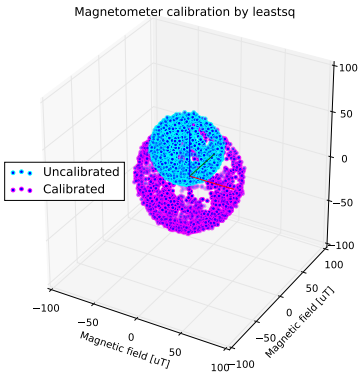

Fig. 5. Visualization of the magnetometer calibration process in 3D magnetic measurement space.

and rotation and scaling (D) which transform that ellipsoid to a sphere of radius $|\vec{h}_{Earth}|$ centered at the origin (see figure 5).

Whichever algorithm is used, it must be provided with measurements from a set of attitudes which have good coverage of $\text{SO}(3)$ [12]. In the ellipsoid-fitting problem, there is insufficient information if we have noisy points from only a small patch of the ellipse. Analogously, the magnetometer calibration problem is difficult with measurements from a small range of attitudes. For a practical example of how calibration can fail, note that the magnitude error minimization is trivially solved by setting $D = -I$ (map all points to the origin) and $\vec{b}_{mag}$ to any vector with magnitude $|\vec{h}_{Earth}|$. With insufficiently diverse measurements, I found that both the least squares and UKF-based algorithms fall into this trap.

I choose to perform *a priori* calibration, and use `scipy.optimize.leastsq` to solve the nonlinear least squares problem.

VI. ESTIMATOR DESIGN

Because my system has nonlinear measurements and approximately Gaussian measurement noise, I use an Unscented Kalman Filter (UKF) as my estimator.

A. Quaternion State in the UKF

The use of quaternions to represent attitude state requires some modification to the standard UKF algorithm. First, the UKF assumes that the state space is a vector space, i.e. that all combinations of settings of the state variables are valid states. While the quaternions $\mathbb{H}$ are a 4-D vector space, only unit quaternions represent valid rotations in $\text{SO}(3)$. Therefore, the attitude state space is the 3-D surface of the unit sphere in $\mathbb{H}$, i.e. it is a manifold, not a vector space.

This invalidates the UKF's representation of uncertainty. The distribution of states should only include unit quaternions, so level-sets of likelihood should be 3-D patches on the surface of the unit sphere (figure 6). This requires a 3-D representation of the attitude covariance. However, in the UKF the dimension of the covariance is equal to the dimension of the state. Therefore, the level-sets of likelihood would be 4-D ellipsoids, and the distribution of states would include many non-unit quaternions (figure 7).

To avoid this, the unit quaternions must be mapped to a 3-D vector space. Fortunately, the unit quaternions form a Lie group ($\text{SO}(3)$), so their logarithms form a Lie algebra ($\text{so}(3)$), which is a vector space (this duality is referred to as the exponential map). The logarithm of a quaternion is defined as follows: for a rotation of θ radians about a unit axis $\hat{n}$:

$$q = [\cos\left(\frac{\theta}{2}\right), \hat{n}\sin\left(\frac{\theta}{2}\right)] \tag{11}$$

$$\log(q) = [\log(|q|), \theta\hat{n}] \tag{12}$$

$$= [0, \theta\hat{n}] \tag{13}$$

The logarithm of a unit quaternion has real part 0, and complex parts equal to the axis-angle representation of the rotation. Because the real part is always zero, it can be dropped to represent the log-unit-quaternions as 3-vectors in

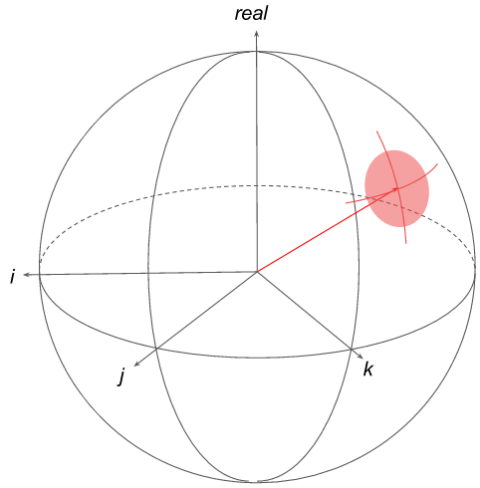

Fig. 6. An illustration of the 4-D unit sphere in $\mathbb{H}$, a unit quaternion (red), and its 3-D 1-σ uncertainty region (pink). The region lies on the unit sphere.

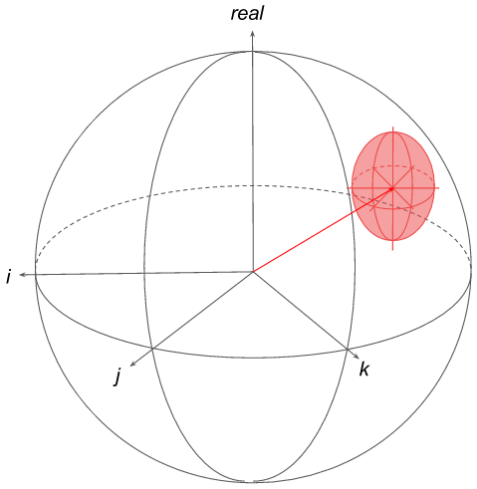

Fig. 7. A unit quaternion (red), and its 4-D 1-σ uncertainty region. The region contains points which are not on the unit sphere.

$\mathrm{so}(3) = \mathbb{R}^3$.

Another issue is that the UKF must average state vectors. However, the unit quaternions are a redundant representation of $\mathrm{SO}(3)$, and a quaternion and its negative represent the same rotation. Before averaging quaternions, one must check their vector dot product, and negate one of the quaternions if it is negative.

B. Kraft's quaternion UKF algorithm

Kraft's quaternion UKF [14] represents attitude uncertainty with a 3-D covariance, and uses the exponential map to convert 3-D attitude disturbance vectors to disturbance quaternions $d\mathcal{Q}^{(i)}$. Its propagation function is presented in algorithm 2. If there are n non-attitude states, then the covariance Q has dimension $(n+3) \times (n+3)$, the state estimate $\vec{x}^{est}$ has dimension $n+4$, and the sigma points $\vec{\mathcal{X}}^{(i)}$ have dimension $n+3$. States $0, 1, 2, 3$ are the attitude quaternion, other states are vector components or scalars. The diagonal entries $Q[0,0], Q[1,1], Q[2,2]$ represent the attitude estimate variance about the body-frame x, y, z axes, with units of rad^2. W is the process noise covariance, and R is the measurement noise covariance.

Algorithm 2 Quaternion UKF Dynamics Propagation.

function PROPAGATEDYNAMICS($\vec{x}^{est}_{k-1|k-1}, Q_{k-1|k-1}, W$)
  $q^{est}_{k-1|k-1} \leftarrow \vec{x}^{est}_{k-1|k-1}[0:4]$
  Compute the sigma points $\vec{\mathcal{X}}^{(i)}_{k-1|k-1}$ from the covariance $Q_{k-1|k-1} + W$
  $d\mathcal{Q}^{(i)}_{k-1|k-1} \leftarrow \mathrm{QUATEXP}(\vec{\mathcal{X}}^{(i)}_{k-1|k-1}[0:3]) \quad \forall i$
  $\mathcal{Q}^{(i)}_{k-1|k-1} \leftarrow q^{est}_{k-1|k-1} \cdot d\mathcal{Q}^{(i)}_{k-1|k-1} \quad \forall i$
  Propagate each state vector $[\mathcal{Q}^{(i)}_{k-1|k-1}, \vec{\mathcal{X}}^{(i)}_{k-1|k-1}[3:]]$ through the state transition function to step $k|k-1$.
  $q^{est}_{k|k-1} \leftarrow \mathrm{QUATAVG}(\mathcal{Q}^{(i)}_{k|k-1} \quad \forall i)$
  $\vec{x}^{est}_{k|k-1}[0:4] \leftarrow q^{est}_{k|k-1}$
  $\vec{x}^{est}_{k|k-1}[4:] \leftarrow \mathrm{MEAN}(\vec{\mathcal{X}}^{(i)}_{k|k-1}[3:] \quad \forall i)$
  $d\mathcal{Q}^{(i)}_{k|k-1} \leftarrow (q^{est}_{k|k-1})^{-1} \cdot \mathcal{Q}^{(i)}_{k|k-1}$
  $\vec{\mathcal{X}}^{(i)}_{k|k-1}[0:3] \leftarrow \mathrm{QUATLOG}(d\mathcal{Q}^{(i)}_{k|k-1})$
  Compute $Q_{k|k-1}$ from $\vec{\mathcal{X}}^{(i)}_{k|k-1}$
  return $\vec{x}^{est}_{k|k-1}, Q_{k|k-1}$
end function

Algorithm 3 Quaternion UKF Measurement Update.

function UPDATEMEASUREMENT($\vec{x}^{est}_{k|k-1}, Q_{k|k-1}, \vec{y}_k, R$)
  $q^{est}_{k|k-1} \leftarrow \vec{x}^{est}_{k|k-1}[0:4]$
  Compute the sigma points $\vec{\mathcal{X}}^{(i)}_{k|k-1}$ from the covariance $Q_{k|k-1}$
  $d\mathcal{Q}^{(i)}_{k|k-1} \leftarrow \mathrm{QUATEXP}(\vec{\mathcal{X}}^{(i)}_{k|k-1}[0:3]) \quad \forall i$
  $\mathcal{Q}^{(i)}_{k|k-1} \leftarrow q^{est}_{k|k-1} \cdot d\mathcal{Q}^{(i)}_{k|k-1} \quad \forall i$
  Map each state vector $[\mathcal{Q}^{(i)}_{k|k-1}, \vec{\mathcal{X}}^{(i)}_{k|k-1}[3:]]$ through the measurement function to get the measurement sigma points $\vec{\mathcal{Y}}^{(i)}_k$.
  $\vec{y}^{est}_k = \mathrm{MEAN}(\vec{\mathcal{Y}}^{(i)}_k \quad \forall i)$
  Compute Q_{yy} from $\vec{\mathcal{Y}}^{(i)}_k$ and R.
  Compute Q_{xy} from $\vec{\mathcal{X}}^{(i)}_{k|k-1}$ and $\vec{\mathcal{Y}}^{(i)}_k$.
  $K \leftarrow Q_{xy} Q^{-1}_{yy}$
  $d\vec{\mathcal{X}} \leftarrow K(\vec{y}_k - \vec{y}^{est}_k)^T$
  $dq = \mathrm{QUATEXP}(d\vec{\mathcal{X}}[0:3])$
  $q^{est}_{k|k} \leftarrow q^{est}_{k|k-1} \cdot dq$
  $\vec{x}^{est}_{k|k}[0:4] \leftarrow q^{est}_{k|k}$
  $\vec{x}^{est}_{k|k}[4:] \leftarrow \vec{x}^{est}_{k|k-1}[4:] + d\vec{\mathcal{X}}[3:]$
  $Q_{k|k} \leftarrow Q_{k|k-1} - K Q_{yy} K^T$
  return $\vec{x}^{est}_{k|k}, Q_{k|k}$
end function

C. Sensor Bias Estimation

I augment the system state $[q_{NED2sensor}, \vec{\omega}^{sensor}]$ with the gyroscope bias parameters $\vec{b}_{gyro}$ and run the UKF on the augmented state. This accounts for the time-varying nature of the gyroscope bias.

I also attempted to estimate the magnetometer bias in-real time as described in [10]. However, this approach requires the system to move through many orientations to build up information about the magnetometer bias. Many trajectories which I may be interested in estimating do not have a sufficient variety of orientations. Instead, I chose to pre-calibrate the magnetometer (see section V-B).

VII. Results

I ran my estimator on data collected by my sensors in two experimental trajectories:

1) static - No motion.
2) xyz90 - Sequential +90° and back rotations about each axis.

I was unable to obtain a ground-truth measurement system, so the true state during these experiments is only roughly known. To compensate, I have also run my estimator on simulated measurements of similar trajectories. The true state trajectory is known for the simulated cases.

All cases start and end with the sensor level and pointing its x axis to magnetic north. Because of the sensor's axis designations and the local magnetic declination, this orientation is $yaw = 15°, pitch = 0°, roll = 180°$ relative to the NED frame. The estimator's initial attitude guess is that the sensor frame is aligned with the NED frame ($q_{init} = [1, 0, 0, 0]$).

The results plots contain Euler angle traces on the bottom-right axes. These are included to display the orientation in a human-readable format, but are not used by the estimator. The estimator represents orientation with a quaternion, the components of which are shown on the top-left axes.

A. Simulation Results

The simulated static trajectory is shown in figure 8. The estimator converges to the correct attitude estimate in approximately 3 s and the body rate converges in 30 s (to a slightly wrong value).

To illustrate the importance of the concerns raised in section VI-A, I also ran a standard UKF on the same data. As shown in figure 9, the estimator fails to converge, and the quaternion has non-unit magnitude. Normalizing the quaternion estimate after each step (figure 10) allows the filter to converge, but the convergence time is significantly longer than with Kraft's filter.

The simulated xyz90 trajectory is shown in figure 11. The initial conditions are the same as for the static case. Note that the yaw uncertainty decreases around $t = 10$ s and $t = 25$ s, when the system experiences large pitch and roll displacements. In these states, the sensor yaw axis is not aligned with gravity, so the accelerometer can provide yaw information, reducing the estimate's yaw variance. Also note that the roll and yaw Euler angle experience a gimbal lock singularity at $t = 25$ s when pitch $= 90°$. The quaternion representation of attitude does not have a singularity.

TABLE III
Attitude estimation performance on static trajectory. Convergence time is first time within 2° of final value. The true values are measured with a level (pitch, roll) and with a compass corrected for declination by [11] (yaw).

Axis	True value	Estimated value	Convergence time
yaw	$(14.8 \pm 2.0)°$	$(17.0 \pm 1.5)°$	1.6 s
pitch	$(0.0 \pm 0.5)°$	$(-1.0 \pm 0.4)°$	0.08 s
roll	$(180.0 \pm 0.5)°$	$(-179.6 \pm 0.4)°$	0.07 s

B. Experimental Results

The experimental static trajectory is shown in figure 12. Note that the z axis body rate is erroneously high due to an erroneously low z axis gyro bias estimate for the first few seconds. The rate-of-change of $q[2]$ (yaw) as it converges from the initial guess ($yaw = 0°$) to its correct value ($yaw = 15°$) causes the filter's z axis body rate estimate to be positive. However, the rate gyro (correctly) reads a body rate of $\approx 0\,°\,s^{-1}$, so the estimator's z axis gyro bias estimate becomes negative to compensate. After yaw becomes constant, it takes some time for the z axis gyro bias estimate to re-converge to its correct value ($\approx 0\,°\,s^{-1}$). I was able to significantly shorten this time by changing the sensor model's rate gyro correlation time from $1/\alpha = 100$ s to a fictitiously low $1/\alpha = 10$ s.

All attitude components attain a final value within 2° of their true value (see table III).

The experimental xyz90 trajectory is shown in figure 13. The 90° motions about the x (at t =5 s to 12 s), y (at t =14 s to 21 s) and z (at t =24 s to 30 s) are correctly detected. Also note that the roll and yaw Euler angle experience a gimbal lock singularity at $t = 17$ s when pitch $= 90°$. As in the simulated case, the quaternion representation of attitude does not have a singularity.

After the trajectory is completed, all attitude components attain a final value within 3° of their true value (see table IV).

VIII. Conclusion

I have successfully implemented Kraft's quaternion based UKF described in [14]. The estimator performs well (attitude error $< 3°$) on simulated and experimental data. To represent attitude uncertainty, Kraft's UKF uses a 3-D exponential

TABLE IV

ATTITUDE ESTIMATION PERFORMANCE AFTER XYZ90 TRAJECTORY. THE TRUE VALUES ARE MEASURED WITH A LEVEL (PITCH, ROLL) AND WITH A COMPASS CORRECTED FOR DECLINATION BY [11] (YAW).

Axis	True value	Estimated value
yaw	$(14.8 \pm 4.0)^\circ$	$(11.8 \pm 1.5)^\circ$
pitch	$(0.0 \pm 0.5)^\circ$	$(-0.7 \pm 0.4)^\circ$
roll	$(180.0 \pm 0.5)^\circ$	$(179.7 \pm 0.4)^\circ$

mapped quaternion covariance instead of directly taking the 4-D covariance of the quaternion state. This representation is theoretically justified because the quaternions which represent valid rotations lie on a 3-D manifold within the 4-D quaternion space $\mathbb{H}$. Practically, I have demonstrated that Kraft's UKF has $\approx 3\times$ faster convergence time that a UKF which uses 4-D quaternion covariance.

I have also demonstrated the characterization of a MEMS IMU and magnetometer. I used Allan variance to assess the white noise and bias walk parameters of the sensors. I employed a nonlinear least squares optimization algorithm to estimate the magnetometer bias and scale factor error.

REFERENCES

[1] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Mar. 2005. [Online]. Available: http://www-users.cs.umn.edu/ trawny/Publications/Quaternions_3D.pdf

[2] J. Kelly and G. S. Sukhatme, "A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors," in *12th International Symposium on Experimental Robotics, 2010*, Delhi, India, Dec 2010. [Online]. Available: http://robotics.usc.edu/publications/711/

[3] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-696, Aug. 2007. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf

[4] Y. Winetraub, U. d. San Bitan, and A. B. Heller, "Attitude determination advanced sun sensors for pico-satellites," Handasaim School, Tel-Aviv University, Israel, Tech. Rep., 2006. [Online]. Available: https://www.agi.com/downloads/corporate/partners/edu/advancedSunSensorProject.pdf

[5] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Proceedings of the International Symposium of Robotics Research (ISRR)*, Flagstaff, AZ, 2011.

[6] M. Dille, B. P. Grocholsky, and S. Singh, "Outdoor downward-facing optical flow odometry with commodity sensors," in *Proceedings Field & Service Robotics (FSR '09)*, July 2009.

[7] *MPU-9150 Product Specification*, 4th ed., InvenSense, Sunnyvale, CA, US, 2013. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/IMU/MPU-9150-Datasheet.pdf

[8] J. Crassidis, "Sigma-point kalman filtering for integrated gps and inertial navigation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, no. 2, pp. 750–756, April 2006.

[9] M. Park, "Error analysis and stochastic modeling of mems based inertial sensors for land vehicle navigation applications," Master's thesis, Department of Geomatics Engineering, University of Calgary, 2004.

[10] J. L. Crassidis, K.-L. Lai, and R. R. Harman, "Real-time attitude-independent three-axis magnetometer calibration," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 115–120, Jan 2005. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.6278

[11] (2015) World magnetic model. US National Geophysical Data Center and the British Geological Survey. [Online]. Available: https://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml

[12] M. Stanley. (2011, March) Hard and soft iron magnetic compensation explained. MEMS Industry Group. [Online]. Available: http://blog.memsindustrygroup.org/2011/03/22/hard-and-soft-iron-magnetic-compensation-explained/

[13] A. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 7, pp. 1346–1356, July 2006.

[14] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, vol. 1, July 2003, pp. 47–54.

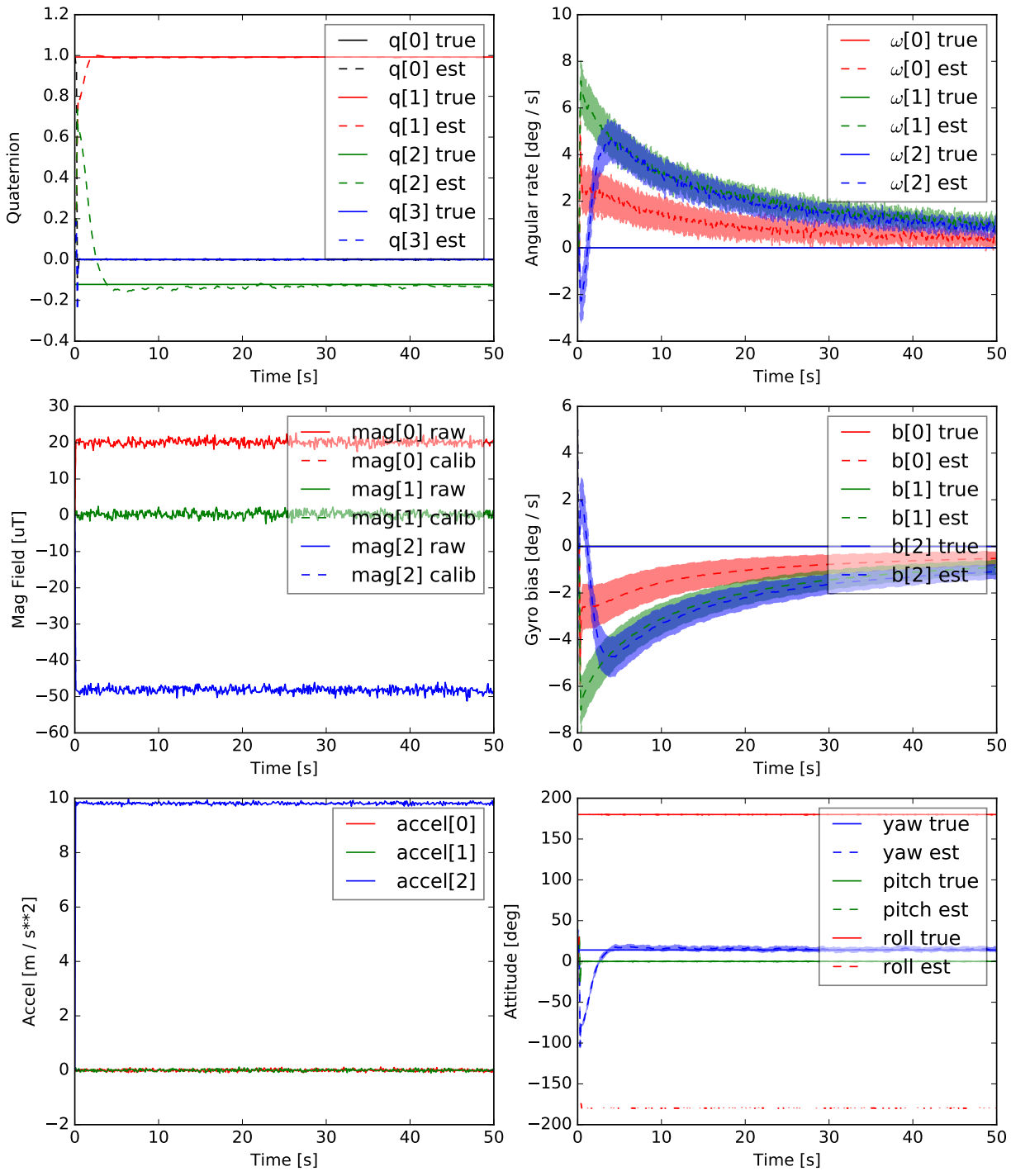

Fig. 8. Estimator outputs, simulated measurements, and true state for the static trajectory using Kraft's quaternion UKF. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.

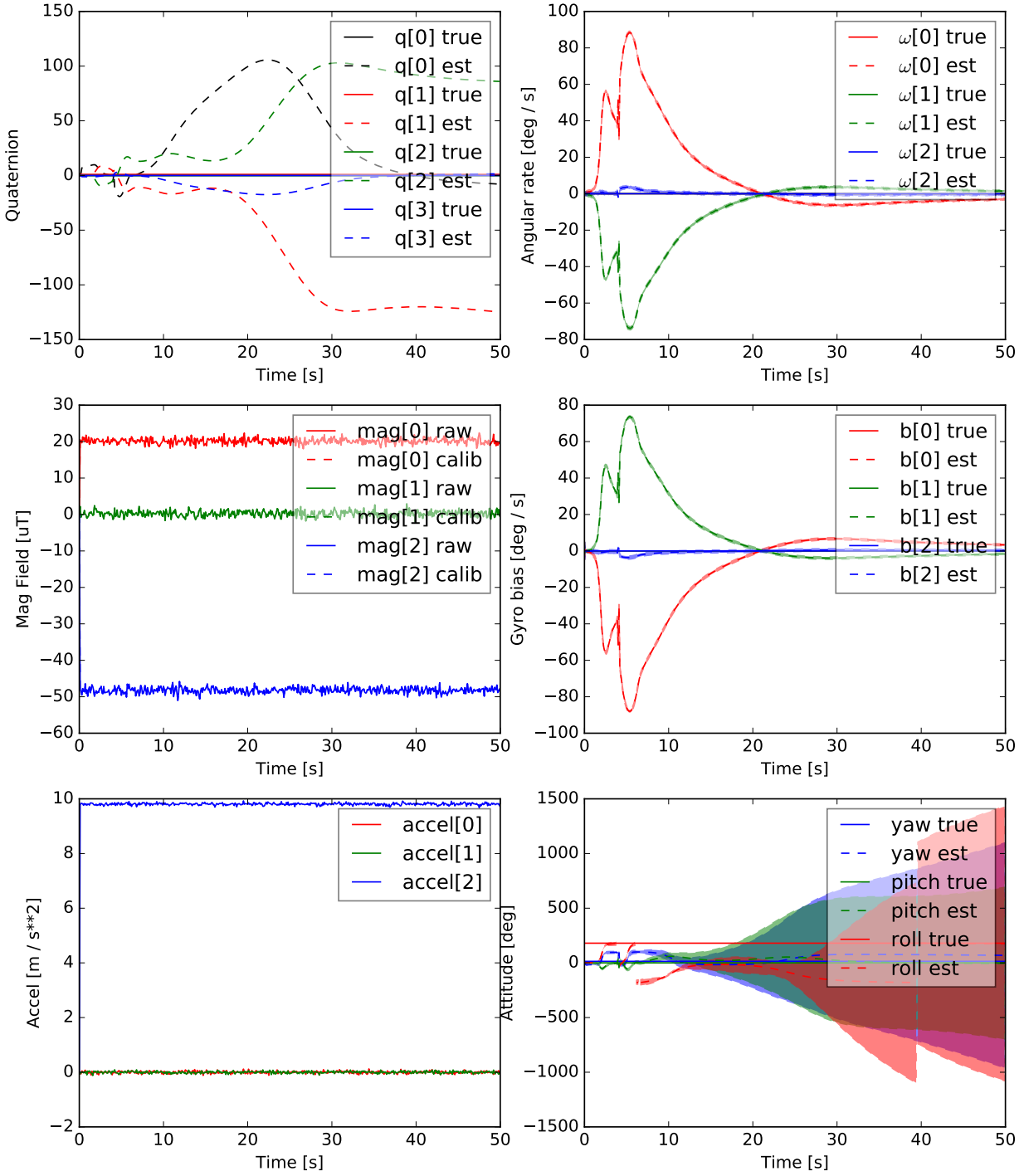

Fig. 9. Estimator outputs, simulated measurements, and true state for the static trajectory using a standard UKF *without* quaternion normalization. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.

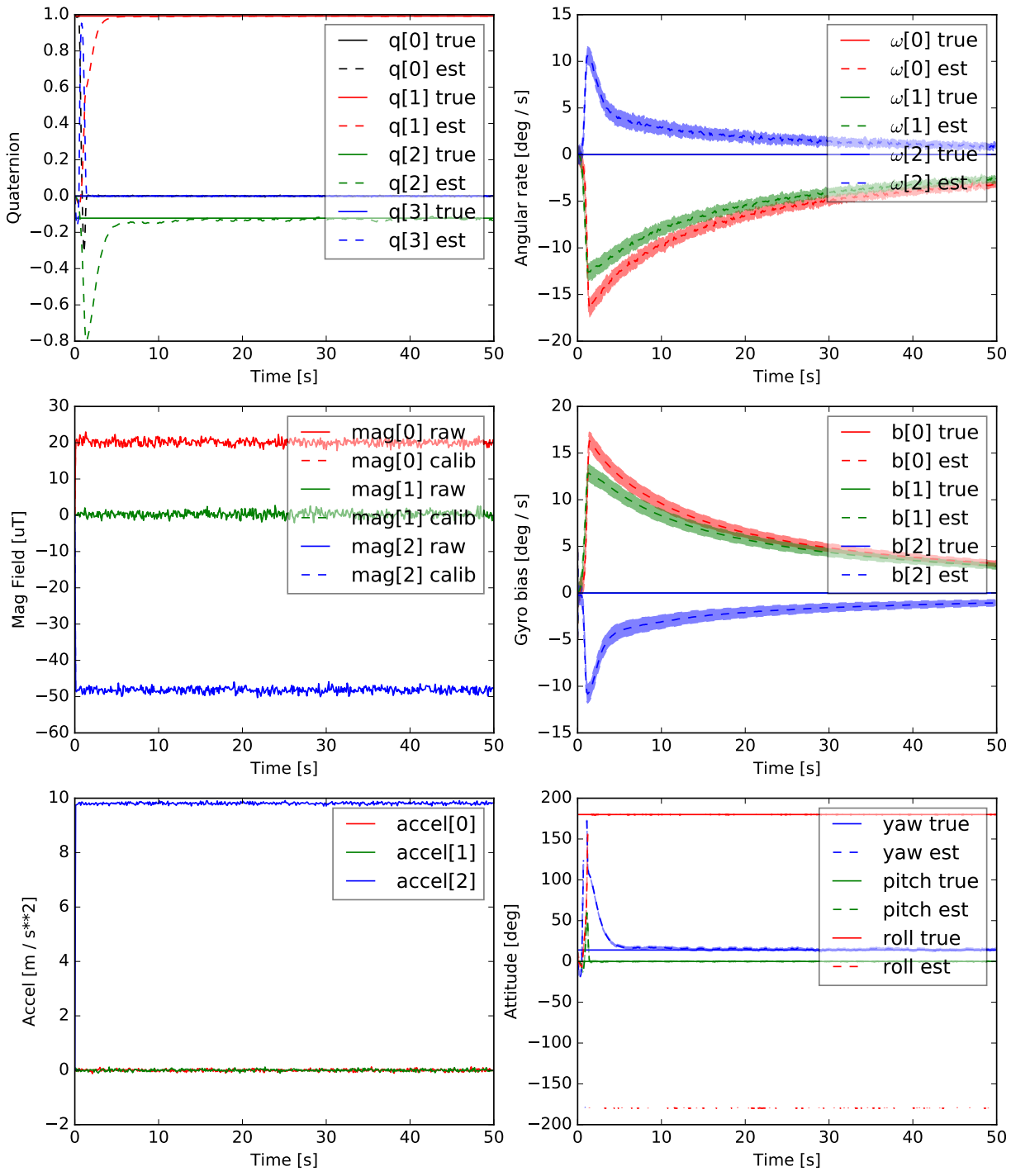

Fig. 10. Estimator outputs, simulated measurements, and true state for the static trajectory using a standard UKF with quaternion normalization. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.

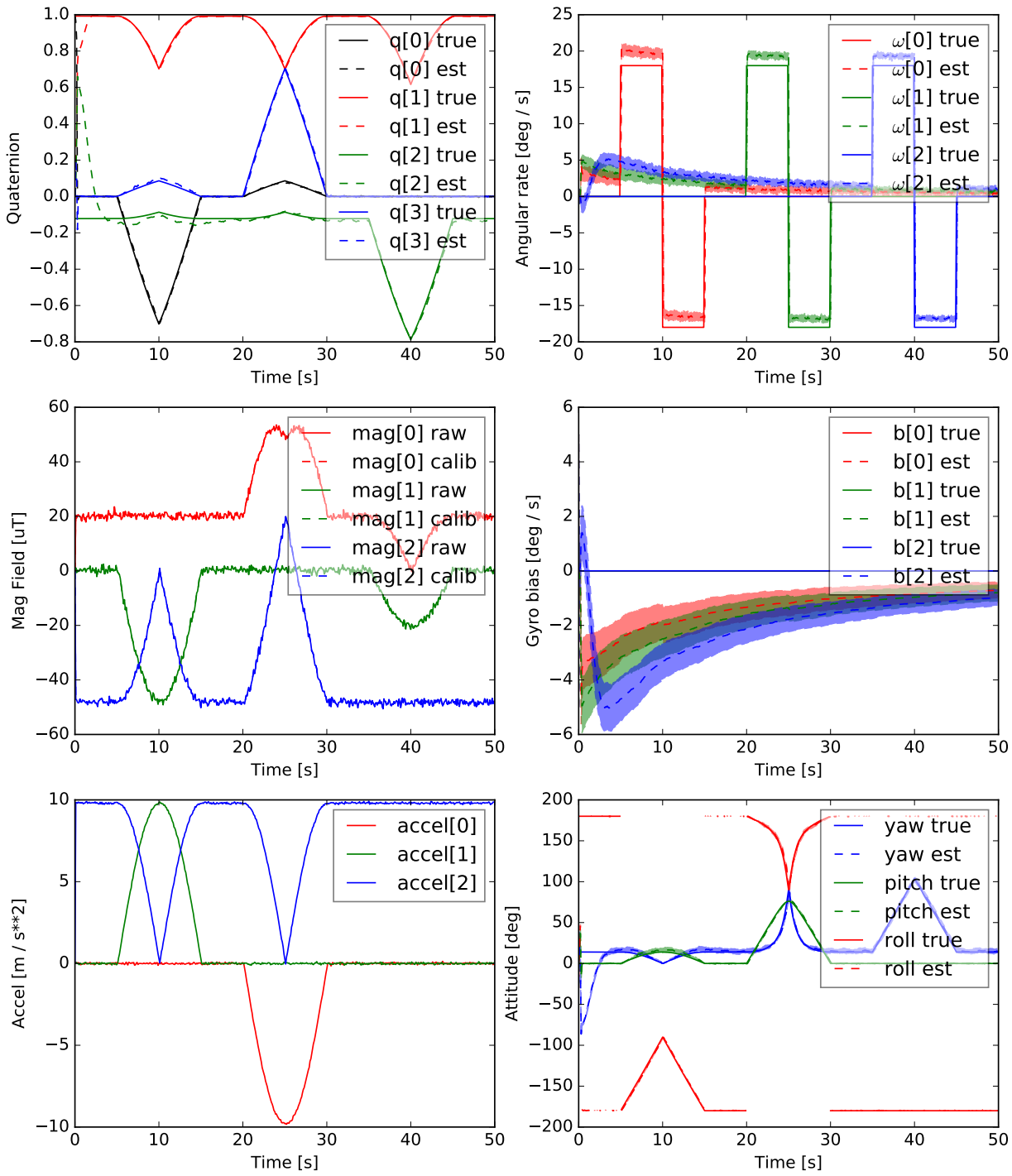

Fig. 11. Estimator outputs, simulated measurements, and true state for the xyz90 trajectory using Kraft's quaternion UKF. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.

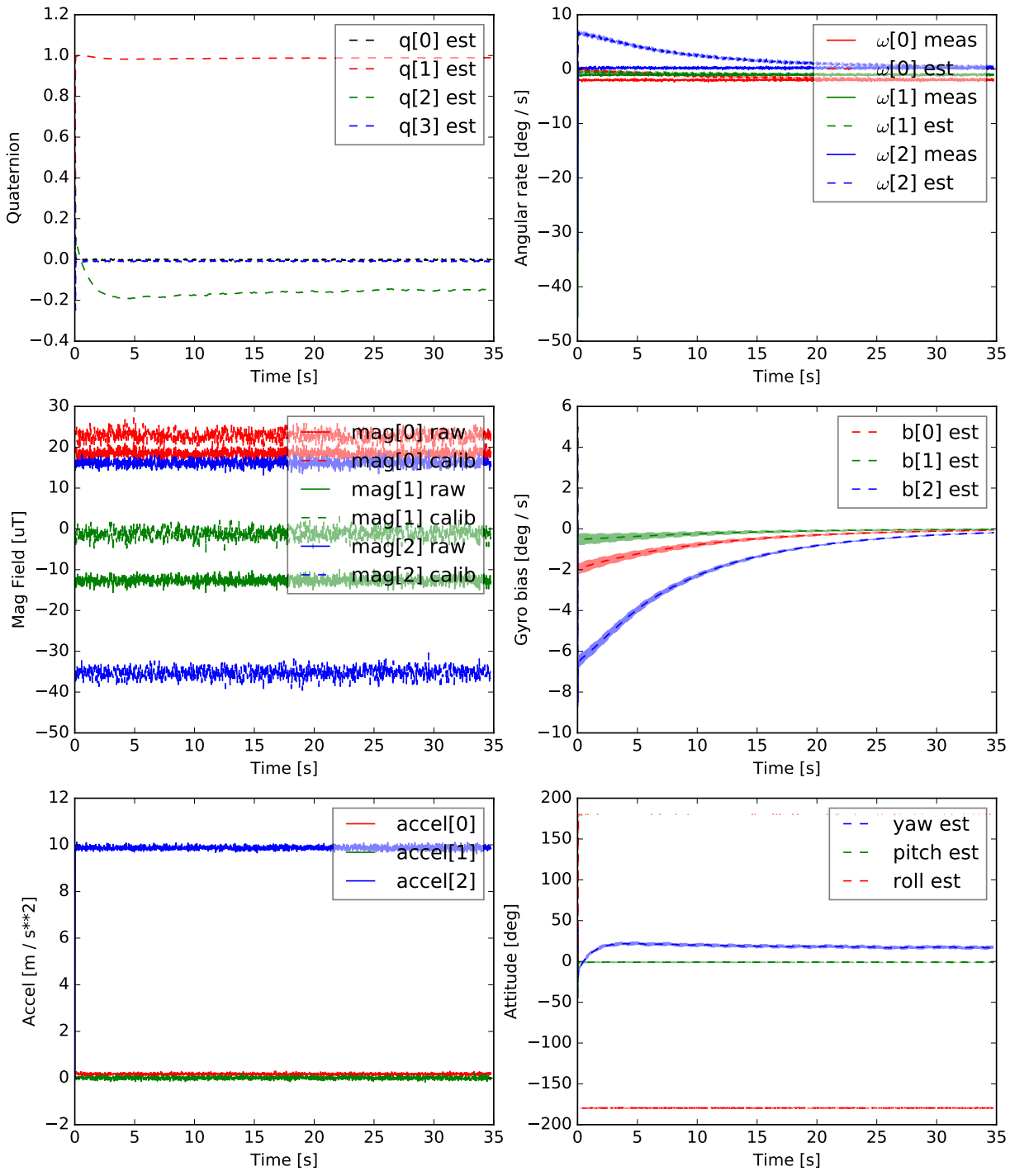

Fig. 12. Estimator outputs and experimental measurements for the static trajectory using Kraft's quaternion UKF. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.

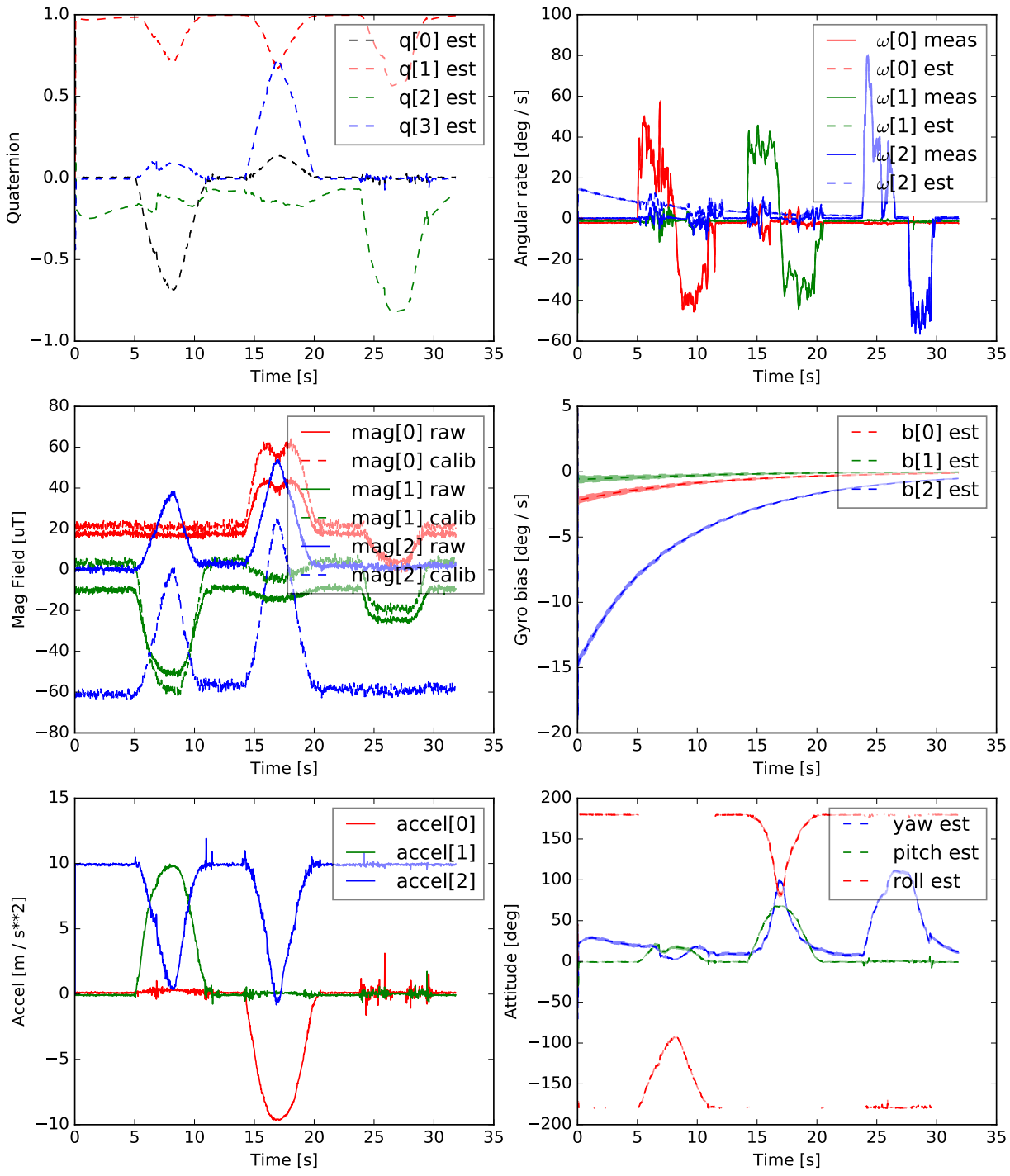

Fig. 13. Estimator outputs and experimental measurements for the xyz90 trajectory using Kraft's quaternion UKF. Estimated states are shown as dashed lines. The 1-σ estimate uncertainty is shown as a shaded region around the dashed lines.