Relative Pose Estimation of a Spin-Stabilized Spacecraft

Aero-Astro 16.322 Stochastic Estimation and Control
Final Project

Created By
Duncan Miller
December 5, 2014

Massachusetts Institute of Technology
Cambridge, MA

Abstract

This paper presents filtering solutions for estimating the relative trajectory of a spin stabilized satellite. The SPHERES satellites have been selected as the hardware testbed for implementing a Multiplicative Extended Kalman Filter and novel Multiplicative Unscented Kalman Filter. Relative state measurements are provided by imaging fiducial markers on the target. The results from this analysis show that when the dynamic nonlinearities and non-Gaussian noises are NOT ignored, the Unscented Kalman Filter performs measurably better than the MEKF. In future ISS test sessions, an Unscented Kalman Filter is expected to now be implemented in order to improve satellite estimation and increase the probability of successful satellite docking.

Duncan Miller, *16.322 Student*
duncanlm@mit.edu

Contents

Duncan Miller, *16.322 Student*
duncanlm@mit.edu

1 Project Motivation

Determining, with confidence, the relative state of a free-floating body has a wide variety of applications. The inspection of drifting or tumbling satellites is of particular interest to the Defense Advanced Research Projects Agency (DARPA). The target could be a military asset, an asteroid, space debris, a comet, or an uncooperative agent. Once characterized the target can be subsequently docked to, serviced or otherwise studied. For example, DARPA Phoenix has proposed a satlet assembly architecture that requires high precision relative sensing in order to aggregate small modules in larger operable satellites.

Visual sensing using known geometries already has heritage in space but is still a problem of high interest. Identifing known geometries through image processing of fiducial markers is well known - indeed it has been in use on the International Space Station (ISS) for several years (Figure 1). However, traditionally the problem is formuated in the context of a dynamic observer and a static target (for navigation). The applications of a static observer and a dynamic (e.g. spin-stabilized) target are less well known. Therein lies the research gap which is the focus of this project.

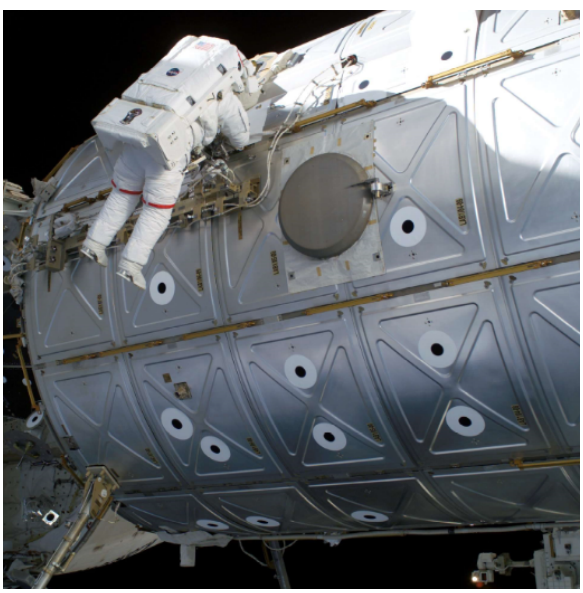

Figure 1: Concentric circle fiducial markers have previously been used on the exterior of the ISS

1.1 Instantiation

The Synchronized Position Hold Engage Reorient Satellites (SPHERES) has been selected as a platform for this investigation. SPHERES is an on-orbit controls testbed operated inside the ISS that provides long duration zero gravity. Athough SPHERES was launched in 2006, the satellites have been continously upgraded with high risk technology payloads. The newest forthcoming payloads are the SPHERES Universal Docking Ports (UDPs) which enable satellite docking and undocking (Figure 2). The author intends to apply the estimators derived within this report to UDP testing operations on Station.

Figure 2: The SPHERES Universal Docking Ports can be used for sensing and estimation by virtue of an onboard camera.

The UDPs have an integrated camera that can be utilized for high precision relative sensing. This is achieved by identifying fiducial markers on the target satellite opposite the opposing camera. A high level concept of operations that the estimation filters enable is shown in Figure 3. In this scenario, the target SPHERE B is in a stablized spin about the camera axis so the fiducials are always in sight by SPHERE A.

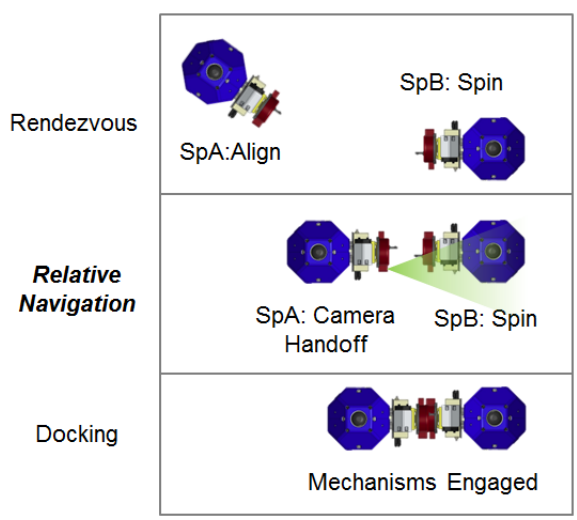

Figure 3: The proposed concept of operations for relative pose estimation.

2 Project Scope

For this project, I have developed a six degree of freedom stochastic simulator of a rigid body spacecraft. With this we explore the performance of two filters in the presence of nonlinear dynamics and non-Gaussian random noise in both simulation and real hardware:

1. A Multiplicative Extended Kalman Filter
2. An Unscented Kalman Filter using the same Multiplicative parameterization of quaternions

This project is of intellectual merit for three reasons. First, it is relevant to the class material covered this semester. I first researched the 6DOF equations of motion for a free-floating satellite. Then I formulated the dynamics as a 13-state time-varying estimation problem that can be reparameterized as a 12-state system which preserves the quaternion norm. Second, this project tackles both the nonlinearities in dynamics and non-Gaussian random noise in measurements. This is necessary in order to measure the performance of the two filters being considered. Finally, the results presented herein consist of Montecarlo simulations and real world data to show that the designed filters work effectively. Using this knowledge, the filtering techniques will be applied directly to the SPHERES testbed on the International Space Station in the scope of a grander mission.

This project delivers (1) a stochastic propagator of nonlinear 6DOF dynamics, (2) a custom visualization of the simulated state, (3) results and conclusions from both simulated and hardware filtering, (4) an auto-coded version of both filters for integration onto a real-world testbed, (5) a body of Matlab code, plots, and config files generated for this assignment.

3 Dynamics

The state of a spacecraft in inertial space can be represented as a 13 element state vector consisting of position, velocity, attitude and rotation rate elements. For a six degree-of-freedom spacecraft, this representation is sufficient to model trajectories, disturbances, and control inputs without any ambiguity. In the scope of the project presented herein, the state to be estimated is a relative state. The position, velocity, attitude and rotation are defined relative to an inertially fixed observer.

Thus, $\mathbf{r}$ is the x,y,z displacement vector from the observer to the target. $\mathbf{v}$ is the x,y,z velocty of the target relative to the observer. $\mathbf{q}$ is a quaternion transform that relates the orientation of the observer's frame *to* the target's frame. The definition of a quaternion is reviewed in Appendix A.1. $\boldsymbol{\omega}$ defines the body-fixed rotation rates of the target relative to the inertial observer. The assembled state vector to be estimated is collected as $\mathbf{x}$.

$$\mathbf{r} = [r_x \quad r_y \quad r_z]^T \tag{1}$$
$$\mathbf{v} = [v_x \quad v_y \quad v_z]^T \tag{2}$$
$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \tag{3}$$
$$\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T \tag{4}$$
$$\boxed{\mathbf{x} = [\mathbf{r} \quad \mathbf{v} \quad \mathbf{q} \quad \boldsymbol{\omega}]^T} \tag{5}$$

The second order dynamics can be rewritten as a set of first order differential equations. The continous time, stochastic, nonlinear dynamics are collected as follows. The process noise ($\mathbf{W}_v$ and $\mathbf{W}_\omega$) enter as acceleration inputs on $\dot{\mathbf{v}}$ and $\dot{\boldsymbol{\omega}}$. Although the forces and torques from the spacecraft thrusters are included in the continous time dynamics, they are dropped as 0 in the scope of this project. Thus, we describe the free floating nonlinear dynamics of a tumbling spacecraft as [3] [1]

$$\dot{\mathbf{r}} = \mathbf{v} \tag{6}$$
$$\dot{\mathbf{v}} = \frac{1}{m}(\mathbf{W_v} + \mathbf{F_T}) \tag{7}$$
$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} = \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega}\\ 0\end{bmatrix} \otimes \mathbf{q} \tag{8}$$
$$\dot{\boldsymbol{\omega}} = \mathbf{J^{-1}}(-\boldsymbol{\omega} \times \mathbf{J}\omega + \mathbf{W}_\omega + \mathbf{F_R}) \tag{9}$$

Where we have defined $\mathbf{J}$ as the moment of inertia tensor along the geometric axes of the SPHERES. The definition of the inertia tensor is covered in Appendix A.2.

$$\mathbf{J} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{10}$$

For convenience, I have introduced the outer-product matrix for quaternion kinematrics as

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & \omega_1 \\ \omega_3 & 0 & -\omega_1 & \omega_2 \\ -\omega_2 & \omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{11}$$

In addition, the cross product matrix $[\boldsymbol{\omega}\times]$ condenses the notation for subsequent state space representations.

$$[\boldsymbol{\omega}\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{12}$$

4 Measurement Modeling

The measurement inputs to the filter are the position and quaternion of the target relative to the observer's camera frame. The observer is assumed to be inertially still and has the role of identifying and tracking the tumbling spacecraft prior to docking.

$$\mathbf{y}_k = \begin{bmatrix} \bar{\mathbf{r}}_k \\ \bar{\boldsymbol{\sigma}}_k \end{bmatrix} \tag{13}$$

These measurements are the output of Haralick's exterior orientation problem. The methodology is derived in Tweddle 2010 [8] for a monochrome camera. Although out of the scope of this project, I have expanded on these principles for an RGB solution, which is what has been applied in Section 9. It has been shown that unique identification of four points on a plane is sufficient to determine the translation ($\mathbf{r}$) and rotation ($\mathbf{q}$) between the observer and target frames without ambiguity.

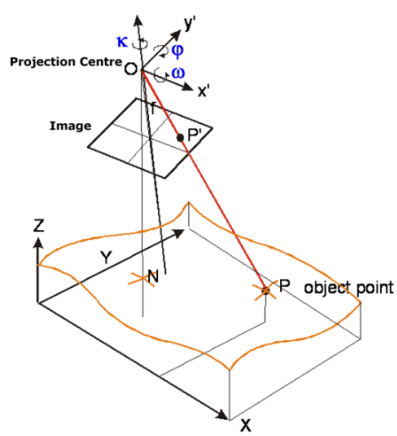

Figure 4: Graphic showing how 3D points are projected onto a 2D image plane for processing [4]

The solution to the exterior orientation problem is obtained by solving a least squares, iterative, nonlinear equation. In addition to the solution, a mean squared reprojection error is also obtained with is proportional to the confidence of the estimate. This is detailed further in Section 7 but in the end, the measurements are treated as given for the purposes of this project.

5 Multiplicative Extended Kalman Filter

A Multiplicative Extended Kalman Filter (MEKF) has been implemented to estimate the full state dynamics of the spin stabilized satellite. Since measurements are only position and attitude, the information from the dynamic equations can be used to obtain a smoothed estimate of the full state (including velocities) that filters out measurement and process noise. The MEKF is reviewed herein.

The continous, nonlinear dynamics (as defined in Section 3) can be expressed in general as a stochastic nonlinear differential equation and nonlinear measurement equation with the following notation.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{B}_w \mathbf{w} \tag{14}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{15}$$

5.1 Reparameterize

The discrete-time Extended Kalman Filter (EKF) has been introduced to account for nonlinearities in the dynamic model. The nonlinear dynamics (expressed as $\mathbf{f}$) can be linearized at every time step around the current best estimate. In this way, the same optimality prinicples employed for the traditional Kalman filter can be used for nonlinear systems to drive the estimation error to zero.

However, the EKF is not mathematically tuned to handle quaternion dynamics. Instead, an MEKF has been introduced in the literature to account for the fact that unit quaterions must maintain unit magnitude. Even small error quaternions must preserve unity (and should not be driven to zero). This motivates the reparameterization of the error quaternion as a three-element set of Modified Rodrigues Parameters (MRP), defined exactly as follows.

$$\boldsymbol{\sigma} = \frac{4}{1+q_4} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{16}$$

$$\dot{\boldsymbol{\sigma}} = \frac{d}{dt}(\boldsymbol{\sigma}) = \frac{4}{(1+q_4)^2} \begin{bmatrix} \dot{q}_1(1+q_4) - \dot{q}_4 q_1 \\ \dot{q}_2(1+q_4) - \dot{q}_4 q_2 \\ \dot{q}_3(1+q_4) - \dot{q}_4 q_3 \end{bmatrix} \tag{17}$$

In this way, the zero vector ($\boldsymbol{\sigma} = \mathbf{0}$) represents no error (the goal). The Modified Rodrigues Parameterization is valid for angle errors less than a full rotation. This is adequate since the error quaterion is reset to 0 after every measurement update.

The MRPs can be calculated from each measurement by determing the quaternion product of the measurement and the complex conjugate of the current state estimate.

$$\bar{\boldsymbol{\sigma}}_{k-1} = \bar{\mathbf{q}}_k \otimes \mathbf{q}^*_{ref_{k-1}} \tag{18}$$

Also, at the end of each iteration, once the MRPs have been filtered, the current best estimate of the target's quaternion can be recovered by taking the quaternion product of the error quaterion and the previous estimate.

$$\mathbf{q}_k = \delta\mathbf{q}(\boldsymbol{\sigma}_k) \otimes \mathbf{q}_{ref_{k-1}} \tag{19}$$

5.2 Linearize

According to the MEKF, at each time step, the dynamics must be linearized around the current best estimate and subsequently discretized in order to apply the Kalman Filter equations.

$$\mathbf{A}_k = \left.\frac{\partial \mathbf{f}(\mathbf{a})}{\partial \mathbf{x}}\right|_{\mathbf{a}=\hat{\mathbf{x}}_{k-1}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \\ & & \ddots \end{bmatrix}\right._{\mathbf{x}=\hat{\mathbf{x}}(t)} \tag{20}$$

$$\mathbf{C}_k = \left.\frac{\partial \mathbf{h}(\mathbf{a})}{\partial \mathbf{x}}\right|_{\mathbf{a}=\hat{\mathbf{x}}_{k-1}} \tag{21}$$

In the scope of this problem, the translational dynamics are trivially linear. However, the rotational dynamics (both the kinematics of the MRPs and Euler's equation) are nonlinear and have been linearized analytically. The kinematics of the MRPs can be written as a function of the angular velocities as follows according to [6] [5].

$$\dot{\boldsymbol{\sigma}} = \frac{1}{2}\left[\mathbf{I}_{3\times3}\left(\frac{1-\boldsymbol{\sigma}^T\boldsymbol{\sigma}}{2}\right) + [\boldsymbol{\sigma}\times] + \boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]\boldsymbol{\omega} \tag{22}$$

By inspection, it is clear that the linearized version of this can be reduced to

$$\dot{\boldsymbol{\sigma}} \approx \frac{1}{2}[\boldsymbol{\sigma}\times]\boldsymbol{\omega} + \frac{1}{4}\mathbf{I}_{3\times3}\boldsymbol{\omega} \tag{23}$$

Linearizing Euler's equation can also be done with some effort. First, it is easiest to expand the vector form of Euler's rotational dynamics into explicit equations. Refer to Appendix A.3 for details of the Jacobian. The linearized continous A matrix evaluated at the best estimate is called $\mathbf{A}_{\boldsymbol{\omega}}$.

$$\dot{\omega} = \mathbf{J}^{-1}\begin{bmatrix} \omega_2(I_{zz}\omega_3 - I_{xy}\omega_1 - I_{yz}\omega_2) - \omega_3(I_{yy}\omega_2 - I_{xy}\omega_1 - I_{yz}\omega_3) \\ \omega_3(I_{xx}\omega_1 - I_{xy}\omega_2 - I_{xz}\omega_3) - \omega_1(I_{zz}\omega_3 - I_{xz}\omega_1 - I_{yz}\omega_2) \\ \omega_1(I_{yy}\omega_2 - I_{xy}\omega_1 - I_{yz}\omega_3) - \omega_2(I_{xx}\omega_1 - I_{xy}\omega_2 - I_{xz}\omega_3) \end{bmatrix} \tag{24}$$

$$\dot{\omega} \approx \mathbf{A}_{\omega}\boldsymbol{\omega} \tag{25}$$

The linearized continous dynamics can be assembled as

$$\dot{\mathbf{x}} = \mathbf{A}_k\mathbf{x} + \mathbf{B}_w\mathbf{w} \tag{26}$$

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{\boldsymbol{\sigma}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{1}{2}[\boldsymbol{\omega}\times] & \frac{1}{4}\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{A}_{\boldsymbol{\omega}} \end{bmatrix}\begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \boldsymbol{\sigma} \\ \boldsymbol{\omega} \end{bmatrix} \tag{27}$$

$$+ \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \frac{1}{m}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}^{-1} \end{bmatrix}\begin{bmatrix} \mathbf{w_v} \\ \mathbf{w}_{\boldsymbol{\omega}} \end{bmatrix} \tag{28}$$

To apply the discrete MEKF equations, we must descritize the dynamics to obtain $\mathbf{A}_d$ and $\mathbf{W}_d$.

$$\mathbf{x}_k = e^{\mathbf{A}\Delta t}\mathbf{x}_{k-1} + \int_0^{\Delta t} e^{\mathbf{A}\tau}\mathbf{B}_w\mathbf{w}d\tau \tag{29}$$

$$\mathbf{x}_k = \mathbf{A}_d\mathbf{x}_{k-1} + \mathbf{w}_k \tag{30}$$

This can be achieved numerically using the Hamiltonian matrix, $\mathbf{S}$ at each time step.

$$\mathbf{S} = \begin{bmatrix} -\mathbf{A}_k & \mathbf{B}_w\mathbf{W}_c\mathbf{B}_w \\ \mathbf{0}_{12\times12} & \mathbf{A}_k^T \end{bmatrix} \tag{31}$$

$$\mathbf{Z} = e^{\mathbf{S}\Delta t} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{0}_{12\times12} & \mathbf{Z}_{22} \end{bmatrix} \tag{32}$$

$$\mathbf{A}_d = \mathbf{Z}_{22}^T \tag{33}$$

$$\mathbf{W}_d = \mathbf{Z}_{22}^T\mathbf{Z}_{12} \tag{34}$$

5.3 Propagate

During each iteration, the state must be propagated from the previous estimate. Euler integration of the nonlinear differential equation provides the state, and the propagated covariance can also be found.

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \cdot \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}) \tag{35}$$

$$\mathbf{Q}_{k|k-1} = \mathbf{A}_d\mathbf{Q}_{k-1|k-1}\mathbf{A}_d^T + \mathbf{W}_d \tag{36}$$

Again, we note the key assumptions about the driving noise $\mathbf{W}_d$, modeled as white.

$$E[\mathbf{w}_k] = 0 \quad \forall k \tag{37}$$

$$E[\mathbf{w}_{k_1}\mathbf{w}_{k_2}^T] = W_{k_1}\Delta(k_1 - k_2) \tag{38}$$

Where

$$\Delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{39}$$

5.4 Measurement Update

The measurement update equations can be use the position and quaternion solutions from the exterior orientation problem. The measurement noise modeling is derived in Section 7.

$$\mathbf{y}_k = \begin{bmatrix} \bar{\mathbf{r}}_k \\ \bar{\boldsymbol{\sigma}}_k \end{bmatrix} = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{40}$$

$$= \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}\begin{bmatrix} \mathbf{r}_k \\ \mathbf{v}_k \\ \boldsymbol{\sigma}_k \\ \boldsymbol{\omega}_k \end{bmatrix} + \begin{bmatrix} \mathbf{v}_r \\ \mathbf{v}_\sigma \end{bmatrix} \tag{41}$$

With this, we are now able to compute successively the Kalman gain, $\mathbf{L}_k$ and the updated state and covariance estimates.

$$\mathbf{L}_k = \mathbf{Q}_{k|k-1}\mathbf{C}_d^T[\mathbf{C}_d\mathbf{Q}_{k|k-1}\mathbf{C}_d^T\mathbf{R}_k]^{-1} \tag{42}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k(\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})) \tag{43}$$

$$\mathbf{Q}_{k|k} = (\mathbf{I} - \mathbf{L}_k\mathbf{C}_d)\mathbf{Q}_{k|k-1} \tag{44}$$

However, in implementing the above equations, I occasionally encountered numerical stability issues when

the covariance diverged due to large condition numbers. Thus, using the fact that the covariance is always positive symmetric definite, we can implement a two step correction consisting of a numerically robust version of the covariance update equation. This has been show to be less sensitive to arithmetic truncation, especially when R is small.

$$\mathbf{Q}_{k|k} = (\mathbf{I} - \mathbf{L}_k\mathbf{C}_k)\mathbf{Q}_{k|k-1}(\mathbf{I} - \mathbf{L}_k\mathbf{C}_k)^T + \mathbf{L}_k\mathbf{R}_k\mathbf{L}_k^T \tag{45}$$

$$\mathbf{Q} = \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T) \tag{46}$$

6 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) provides an alternative filtering technique compared with the MEKF. While the MEKF ignores the nonlinearities of the model, the UKF propagates a set of sample points through the noninear model, thereby obtaining a better characterization of the mean and covariance. My implementation of this filter is achieved using the same dynamics and MRPs presented in the previous section. The generalized UKF method is reviewed herein.

At each iteration, first, a set of 2n sigma points is generated about the current best estimate.

$$\mathbf{x}_{k-1}^i = \hat{\mathbf{x}}_{k-1|k-1} \pm \tilde{\mathbf{x}}^i \tag{47}$$

$$\tilde{\mathbf{x}}^i = \sqrt{n\mathbf{Q}_{k-1|k-1}}_i, \quad i = 1...n \tag{48}$$

Where $\sqrt{n\mathbf{Q}}_i$ is the ith row of the square root matrix. Each sigma point is subsequently propagated through the discrete, nonlinear dynamic equations.

$$\hat{\mathbf{x}}_k^i = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^i, \cancel{\mathbf{u}_{k-1}}, \cancel{t_{k-1}}) \tag{49}$$

The propagated sigma points $\hat{\mathbf{x}}_k^i$ can be used to obtain a better approximation of the propagated mean and covarience.

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\hat{\mathbf{x}}_k^i \tag{50}$$

$$\mathbf{Q}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}(\hat{\mathbf{x}}_k^i - \hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{x}}_k^i - \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{w}_{k-1} \tag{51}$$

New sigma points can be generated to obtain a more accurate measurement prediction. In the scope of this project, the measurement update equation is linear and this step is trivially unncessary.

$$\mathbf{x}_{k|k-1}^i = \hat{\mathbf{x}}_{k|k-1} \pm \tilde{\mathbf{x}}^i \tag{52}$$

$$\tilde{\mathbf{x}}^i = \sqrt{n\mathbf{Q}_{k|k-1}}_i, \quad i = 1...n \tag{53}$$

$$\hat{\mathbf{y}}_k^i = \mathbf{h}(\hat{\mathbf{x}}_k^i, t_k) = \mathbf{C}\hat{\mathbf{x}}_k^i \tag{54}$$

$$\hat{\mathbf{y}}_k = \frac{1}{2n}\sum_{i=1}^{2n}\hat{\mathbf{y}}_k^i \tag{55}$$

The estimated covariance and cross-covariance can be obtained using the sample variance equations.

$$\mathbf{Q}_{yy} = \frac{1}{2n}\sum_{i=1}^{2n}(\mathbf{y}_k^i - \mathbf{y}_k)(\mathbf{y}_k^i - \mathbf{y}_k)^T + \mathbf{R}_k \tag{56}$$

$$\mathbf{Q}_{xy} = \frac{1}{2n}\sum_{i=1}^{2n}(\mathbf{x}_k^i - \mathbf{x}_{k|k-1})(\mathbf{y}_k^i - \mathbf{y}_k)^T \tag{57}$$

Finally, we have the Kalman gain and update equations as follows.

$$\mathbf{L}_k = \mathbf{Q}_{xy}\mathbf{Q}_{yy}^{-1} \tag{58}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{59}$$

$$\mathbf{Q}_{k|k} = \mathbf{Q}_{k|k-1} - \mathbf{Q}_{xy}\mathbf{Q}_{yy}^{-1}\mathbf{Q}_{xy}^T \tag{60}$$

$$= \mathbf{Q}_{k|k-1} - \mathbf{L}_k\mathbf{Q}_{yy}\mathbf{L}_k^T \tag{61}$$

7 Noise Modeling

7.1 Process Noise

The process noise enters the dynamics in the linear and angular rotational acceleration equations. $\mathbf{W}_v$ and $\mathbf{W}_{\boldsymbol{\omega}}$ are the disturbance accelerations that are applied to both free floating spacecraft. Since the state vector being modeled is a relative state, the process noise can be grouped onto the target spacecraft in the scope of relative estimation.

The process noise has been modeled as Gaussian white noise. For an orbiting spacecraft it can include microforces such as solar pressure, gravity gradients and atmospheric drag. However, in the instantiation of this estimation problem (i.e. SPHERES inside of the International Space Station) there are air drafts from circulating fans that provide the largest magnitude disturbances. To make the results the most interesting, this is what has been modeled.

$$E[\mathbf{W}_v(\tau_1)\mathbf{W}_v(\tau_2)^T] = \mathbf{W}_{c1}\delta(\tau_1 - \tau_2) \tag{62}$$

$$E[\mathbf{W}_\omega(\tau_1)\mathbf{W}_\omega(\tau_2)^T] = \mathbf{W}_{c2}\delta(\tau_1 - \tau_2) \tag{63}$$

$$E[\mathbf{W}_v(\tau_1)\mathbf{W}_\omega(\tau_2)^T] = \mathbf{0}_{3\times 3} \tag{64}$$

7.2 Random Measurement Noise

Three families of measurement noise have been implemented in order to achieve the most realistic performance of the filters. First, the traditional random noise has been included in the measurements. However, the noise on the measurements has been modeled as nonlinear/non-Gaussian. Due to nonlinearities in both the dynamics and (now) the measurements we can conclude that the MEKF will not be an unbiased estimator for this problem.

The derivation for the nonlinear noise measurements begins with assumption that the positions of the fiducial markers on the image plane suffer from small Gaussian noise perturbations. That is, the true pixel position of the concentric circles can be distrubed normally with some variance that is a function of the atmospheric lighting, image blur, software threasholding and CMOS noise. However, we can conclude that there is a nonlinear transform from pixel noise to quaternion measurement noise in two of the three axes.

Figure 5 depicts the noise resulting from the twist case. w represents the pixel noise error which is applied in the image plane around the first Euler angle (x-axis twist). Using small angle approximations, the transfomed noise in the first Euler angle (ϕ) is

$$\phi_w = \sin^{-1}\frac{w}{d} \approx \frac{w}{d} \tag{65}$$

Where d has been defined to be the distance between the center of the concentric circle (black dot) and the center of the fiducial plane. This is a linear transform to the first Euler angle.

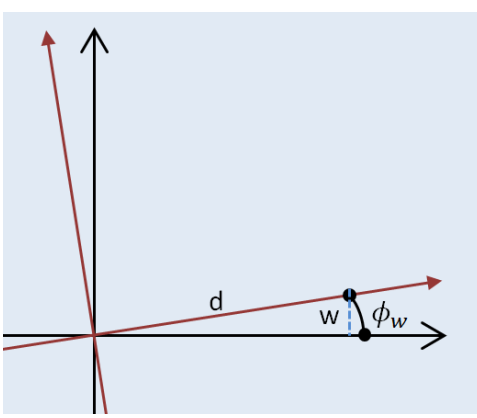

Figure 5: This graphic shows higher measurement sensitivity (lower noise) for an x-axis twist (blue is the image plane).

However, the two tilt degrees of freedom are much more sensitive to Euler angle noise. Athough the pixel noise is the same, Figure 6 shows that the same pixel noise w results in a larger angle noise in the second and third Euler angles. A small angle approximation provides the following relation.

$$\cos\theta_w = \frac{d-w}{d} = 1 - \frac{w}{d} \tag{66}$$

$$\cos\theta_w \approx 1 - \frac{\theta_w^2}{2} = 1 - \frac{w}{d} \tag{67}$$

$$\theta_w, \psi_w \approx \sqrt{\frac{2w}{d}} \tag{68}$$

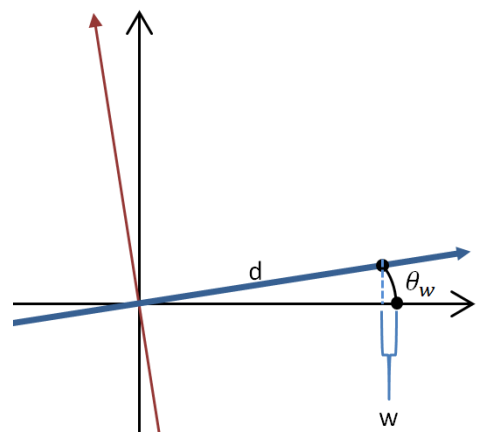

Figure 6: This graphic shows lower measurement sensitivity (higher noise) for a y- or z-axis tilt (blue is the image plane).

Since the relation w/d is smaller than 1, it is clear that the measurement noise on the second (and third) Euler angles are subject to a nonlinear amplifying transform. After the Euler angle noise has been generated, it is further transformed into an error quaterion as derived in Appendix A.1. This error quaternion is integrated into the measurement according to the previously introduced multiplicative outer product.

$$\mathbf{q}_{meas}(k) = \delta\mathbf{q}_w \otimes \mathbf{q}_{truth}(k) \tag{69}$$

Thus, in modeling $\mathbf{R}$ within the filters, we have weighted the measurement states according to a variable η^2 which is a measure of the confidence of the measurement. In the hardware implementation, this is an output of the iterative solution to the exterior orientation problem and is an input for every filter step.

7.3 Type I Errors

In probability theory, Type I errors (or false positives) can lead to a large divergence of the measurement state. In the scope of this problem, it is concievable (in fact, occasionally expected) that the camera tracking algorithm

falsely identifies a blob as a fiducial marker. This can result in a measurement that may be significantly offset from the current state estimate. In the simulation, this has been emulated by increasing the variance of the discrete random noise by a factor of 5 approximately 5% of the time.
Athough the filters presented herein do not actively perform outlier rejection, I chose to model this in the system in order to make conclusions on the robustness of the filter.

7.4 Type II Errors

In probability theory, Type II errors (or false negatives) arise when a measurement exists but the algorithm chooses not to use it. In the scope of this problem, it is concievable that a measurement may not be acquired every time step. This can be caused by camera disturbances, motion blur, or external objects obscuring the markers. As a result, the filter propagates for longer time periods between measurements. This has been implemented in my simulation by dropping measurements approximately 5% of the time to get a more realistic performance from the filters.

8 Simulation Results

The scenario studied is a spin stabilized case about the x-axis of the target SPHERES satellite. The intial conditions ensure that the SPHERES is aligned with the fiducials pointed roughly towards the camera and the products of inertia of the SPHERES ensure that the nonlinear dynamics are interesting.

$$\mathbf{r}_0 = [1 \quad 0 \quad 0]^T m \tag{70}$$
$$\mathbf{v}_0 = [-0.1 \quad 0 \quad 0]^T m/s \tag{71}$$
$$\mathbf{q}_0 = [0 \quad 0 \quad 1 \quad 0]^T \tag{72}$$
$$\boldsymbol{\omega}_0 = [0.1 \quad -0.02 \quad -0.0001]^T rad/s \tag{73}$$
$$\mathbf{x}_0 = [\mathbf{r_0} \quad \mathbf{v_0} \quad \mathbf{q_0} \quad \boldsymbol{\omega}_0]^T \tag{74}$$

In the scenario considered, I have propagated for 20 seconds with a time step of 0.01. Longer times were also studied and the results were comparable. The noise covariances, $\mathbf{R}$ and $\mathbf{W}_c$, were also varied. In the presented solution, I have set

$$\mathbf{R} = \begin{bmatrix} 0.05\eta^2\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \eta^2\mathbf{I}_{3\times3} \end{bmatrix} \tag{75}$$
$$\mathbf{R}_k = \mathbf{R} \tag{76}$$
$$\mathbf{W}_c = 10^{-4} \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 0.002\mathbf{I}_{3\times3} \end{bmatrix} \tag{77}$$
$$\mathbf{W}_k \approx \mathbf{W}_c \Delta t \tag{78}$$

Due to the conflicting units between states, some states have smaller or larger variances. The matrices were selected based on expected disturbances. I also note that $\mathbf{R}$ is a function of η^2, an output of the exterior orientation problem and generally small (≈ 0.005).

8.1 Representative Convergence

Here we present representative convergence of the filter for all 13 states. Appendix C contains larger versions of these plots. It can be seen in Figure 20 that the noise is relatively white except for the Type I errors which are the obvious state outliers. The large turquoise bullet represents the initial guess of the state, which quickly converges to the truth state.

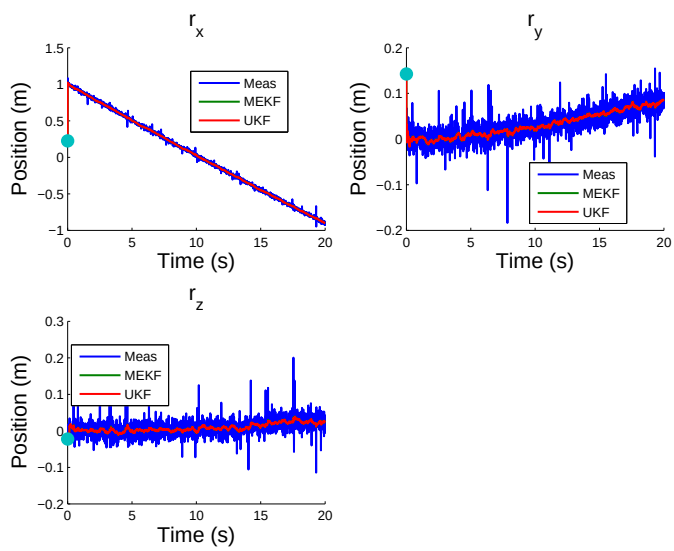

Figure 7: Representative tracking of position states with full noise modeling.

For the position and velocities, subject to process noise, the dynamics are linear and it is obvious that the MEKF and UKF behave similarly and very well. Their differences in pure linear estimation really can only be discerned through Montecarlo simulations.

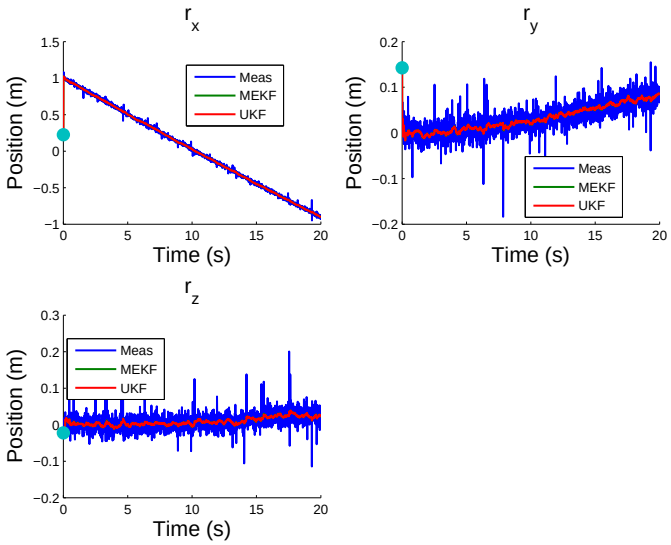

Figure 8: Representative tracking of velocity states with full noise modeling.

The merits of the Unscented Transform manifest themselves in the estimation of the nonlinear rotational dynamics with non-Gaussian noise. In Figure 22, it can be seen that the magnitude of the noise varies periodically. Since this plot is busy and small, a larger version is reproduced in the Appendix and the error vector is plotted in Figure 11.

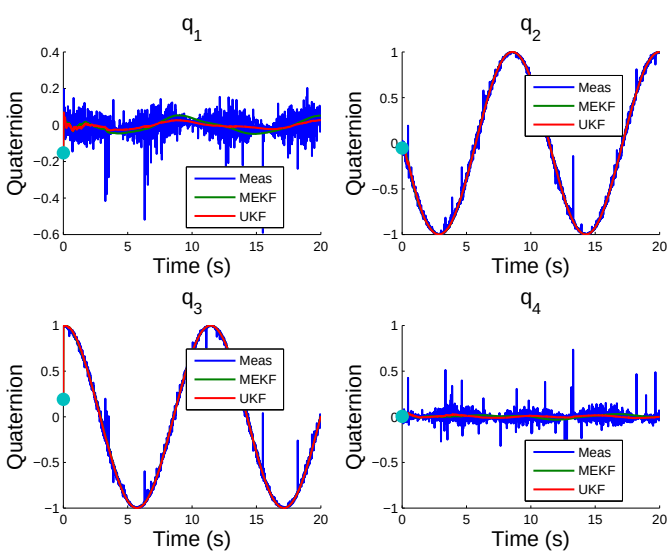

Figure 9: Representative tracking of quaternion states with full noise modeling.

It can be seen that the MEKF suffers from higher frequency, larger magnitude oscillations in the ω estimate compared to the UKF.

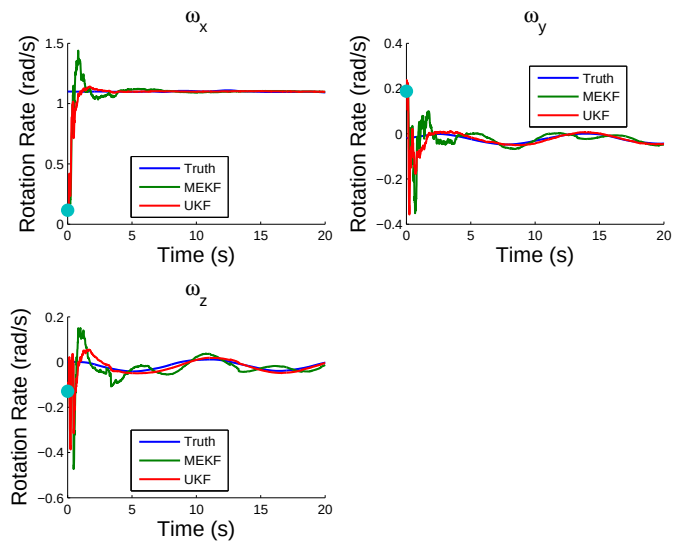

Figure 10: Representative tracking of angular velocity states with full noise modeling.

8.2 Representative Errors

In this section, we plot the estimation errors relative to the known truth states. The UKF performs better across the board in the nonlinear states. It is noted that the UKF does suffer from higher frequency errors in the initial few steps compared with the MEKF but this is eventually smoothed out for the better.

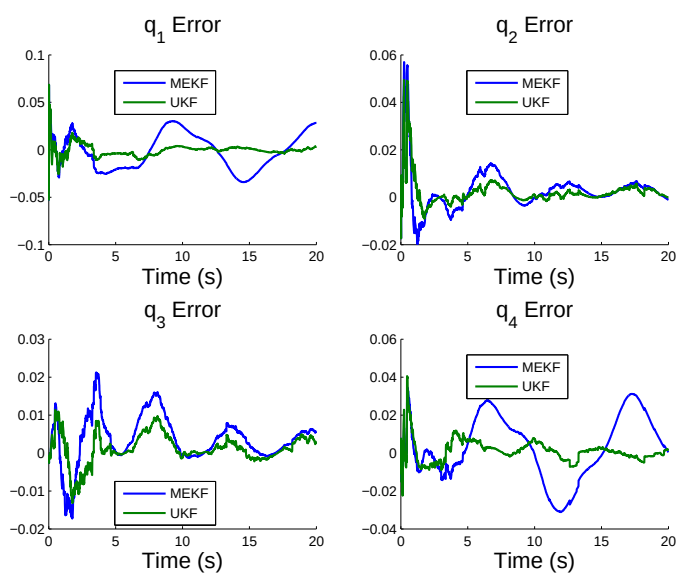

Figure 11: Representative quaternion error with full noise modeling.

The effects of the Type I errors get smoothed and integrated into the angular rate estimates. It can be seen that

the estimates suffer latch ups (sharp discontinuities) when multiple Type I errors are observed successively (seen in Figure 22). However, it is clear that the UKF recovers much better than the MEKF from Type I errors.

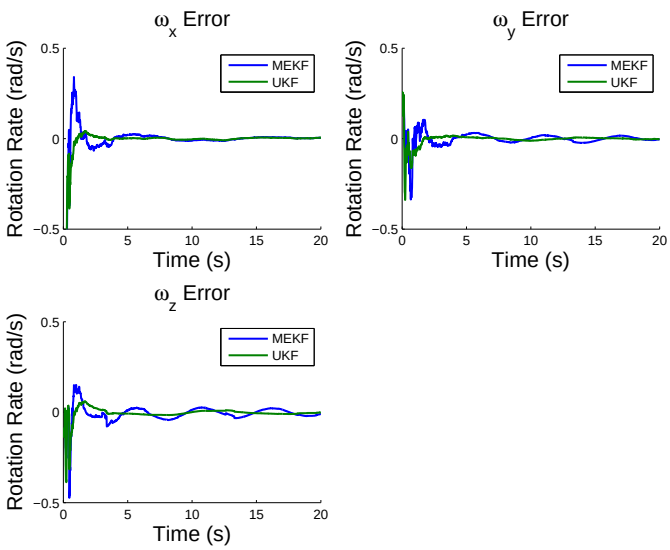

Figure 12: Representative angular velocity error with full noise modeling.

8.3 Modeling Variations

As an itellectual exercise, I have also moldeled the system with varying degrees of noise to better understand the robustness of the controllers. First, I considered a simplified white noise case where the noise on the quaternion measurement is directly white (does not undergo the nonlinear transform described previously). In addition there are no Type I errors. In this case the MEKF and UKF perform nearly identically.

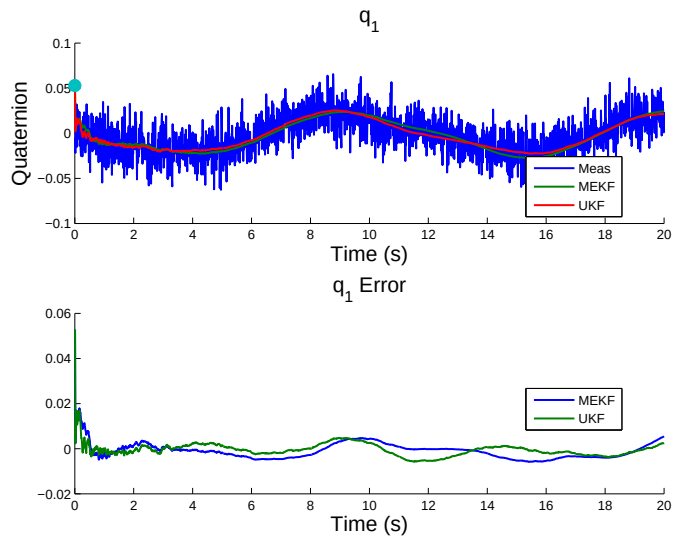

Figure 13: Representative angular velocity error with full noise modeling.

In a second noise variation investigation, I changed the noise covariances within the filters by a factor of two relative to the true W and R. In a real world filter, the process noise and measurement noise intensities will not be known exactly. In fact, they may be different by a factor of 2 or more. I ran the MEKF and UKF filters and both demonstrated robustness to variation.

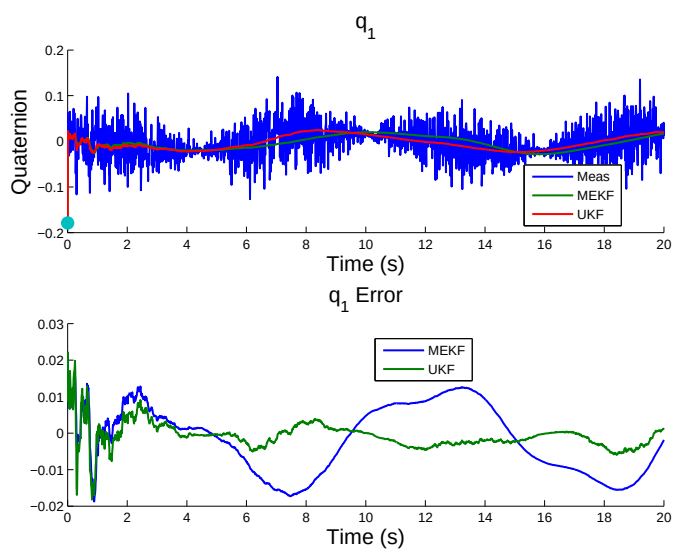

Figure 14: Representative angular velocity error with full noise modeling.

8.4 Montecarlo Simulation

Since a single filter run is based on stochastic noise, it is natural to perform a Montecarlo simulation in order to

judge and compare the expected filter performance. To this end, I have simulated 100 scenarios with identical noise intensities. Using the results from each run, we can compute a metric called the Mean Squared Error (MSE). This can be calculated by summing over all time for each run for each filter. Since the system is subject to non-linear dynamics and non-gaussian noise, the MEKF is no longer a non-biased estimate of the state, which manifests itself in the derivation of the MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{x}}_i - \mathbf{x}_i)^2 \tag{79}$$

$$MSE = Var(\hat{\mathbf{x}}) + (Bias(\hat{\mathbf{x}}, \mathbf{x}))^2 \tag{80}$$

However, the MSE is not the best performance parameter that can be used because its magnitude is a function of the base units of measurement. Thus, we introduce the normalized root mean squared deviation as follows.

$$NRMSD(\hat{\mathbf{x}}_i) = \frac{\sqrt{MSE(\hat{\mathbf{x}}_i)}}{\mathbf{x}_{max_i} - \mathbf{x}_{min_i}} \tag{81}$$

The NRMSD can be calculated for each state and averaged over all montecarlo runs. From the NRMSD (Figure 15), we can make conclusions about the performance of each filter.

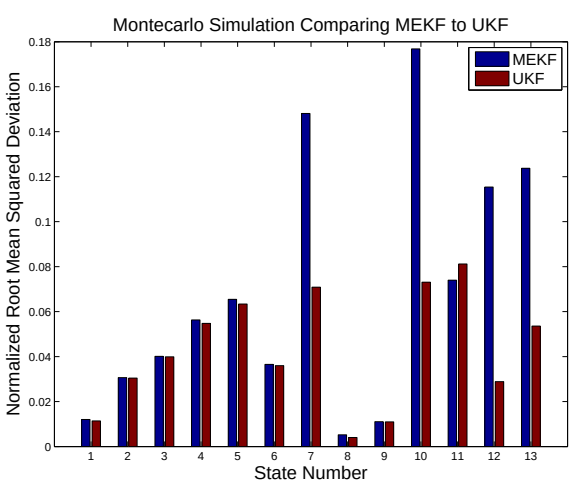

Figure 15: The non-dimensional performance of the MEKF and UKF filters (lower is better)

It is obvious that for the linear translational dynamics, the performance of the MEKF relative to the UKF is nearly identical. However, in the presence of nonlinearities and non-Gaussian noise, it becomes obvious that the UKF has significant advantages, especially in two of the quaternion states (q_1 and q_4) and two of the of the angular velocity rates (w_y and w_z which are nonzero precisely due to nonlinearities).

9 Hardware Testing

As a final step of validating the performance of these filters, I have run them in real-time on actual hardware in the lab. To achieve this, I have used the Matlab Coder toolbox to auto-code the Matlab function into C++ files. The SPHERES-VERTIGO processor runs C++ so it was simple enough to integrate the proposed MEKF and UKF filters within the SPHERES software framework. As shown in the video in class, the hardware demonstration involved floating the SPHERES on a 3DOF air carriage and collecting state measurements during a "flyby" maneuver. Since on the ground we are restricted to 3DOF, a spin stabilized scenario was infeasible to implement.

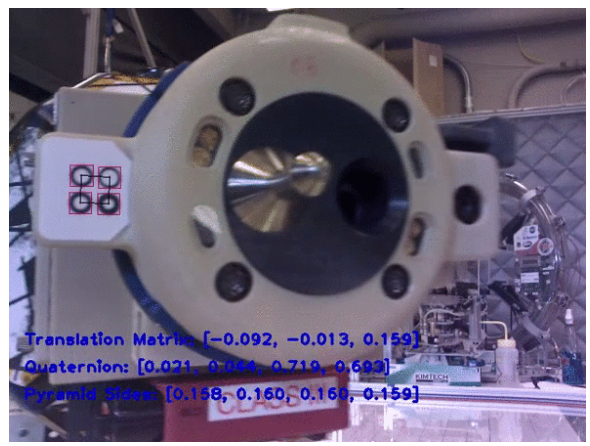

Figure 16: One instant of a measurement solution (target is locked)

The camera was capable of acquiring a lock on the target spacecraft (Figure 16) during the pass. From this, the full state data was able to be estimated by both filters. Although we are missing a "truth" state, I was able to verify though the video that the direction and order of magnitude of the position and velocity states was correctly being estimated. Moreover, the difference between MEKF and UKF is plotted in Figure 30 and it can be seen that they tend to converge over time (for the linear translational dynamics).

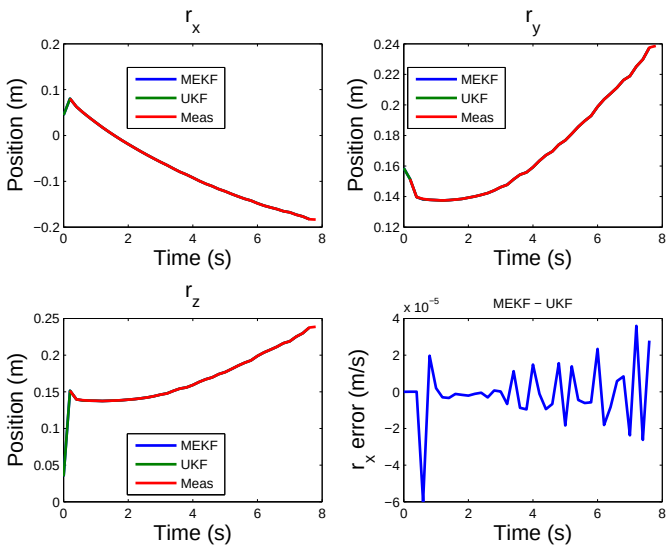

Figure 17: Quaternion estimate of the state in the SPHERES hardware demo

Just by visual inspection of the video capture, it is clear that the rotation rates are small and do not vary by more than 10% throughout the run. However, the MEKF estimator exhibits high estimate oscillations in comparison to the UKF until converging to a more reasonable solution.

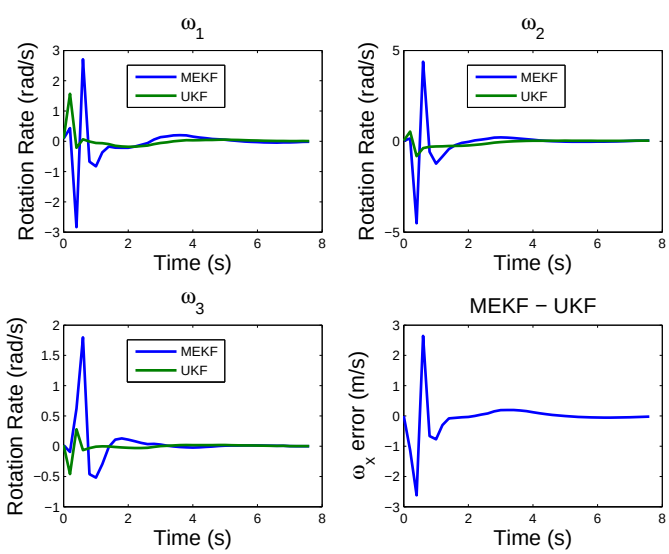

Figure 18: Angular rate estimate of the state in the SPHERES hardware demo

Finally, I note that the distrubances experienced in this hardware demo are on the same order of magnitude expected on the ISS, if not a bit larger than expected. The air carriages are not perfect isolators and the glass table is never perfectly balanced, which introduces process noises that we wouldn't expect in zero gravity. The complete results of the hardware testing can be found in Appendix C.2.

10 Summary and Conclusions

In this report, we have investigated the performance of two strategic filtering methods: a Multiplicative Extended Kalman Filter (MEKF) and a Muliplicative Unscented Kalman Filter (UKF). Specifically, we have simulated the stochastic dynamics of a spin stabilized satellite and have applied advanced filtering techniques to estimate the target state in the presence of measurement uncertainties. After performing a thorough Montecarlo simulation, we were able to conclude that the UKF performed measurably better when estimating nonlinear states with non-Gaussian noise. In addition, I applied my solution to a real hardware testbed (SPHERES) and showed that the UKF had preferred performance.

The results from this project have direct application on upcoming SPHERES test session on the ISS. Future work on this project will address some simplifications and assumptions necessarily made here for simplicity.

- *Actuator Modeling*: The external forces and torques generated by the satellites have been neglected in this analysis. In the future if this filter is used during satellite docking maneuvers, it will be preferred to feed forward the thruster firing commands in the filter.
- *Free Floating Observer*: The model presented herein assumes a static observer and dynamic targer. However, for satellite-satellite docking, both agents are dynamic. It may make sense to separate the dynamics of both bodies for more accurate modelling.
- *Process Noise*: The process noise on the ISS is expected to be marginally different than the process noise for the flat floor demonstration. A thorough characterization of the intensity should be estimated prior to use.
- *Sensor Noise*: Similarly, the sensor noise on the ISS is expected to be a function of the lighting conditions and other disturbances. A thorough characterization of the intensity should be estimated prior to use.
- *Time Lag*: There is no guarantee that the measurements will be acquired regularly or that the Δt will be perfect because Linux is a non-real time operating system. There will be some non-deterministic time lag in the system between measurement update and thruster actuation. This should be taken

into account as much as is reasonably possible prior to use.

The filters may also be improved by implementing an outlier rejection algorithm that can identify and discard Type I errors which provided a certain amount of estimation latch-up. Also, it should be noted that the improvements from the UKF do not come for 'free'. The additional sigma points added computation time for propagation and filtering on the order of n. In the end, I was impressed by the versatility of these filtering techniques and intend to use them in future estimation problems throughout my career. The methodology presented in this paper may be scaled to future manned or unmanned missions to LEO, the Moon, Mars and beyond.

References

[1] Sebasti Xamb Descamps. Euler and the dynamics of rigid bodies. *Quaderns dHistria de lEnginyeria*, Volum IX, 2008.

[2] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Stanford University*, 2006.

[3] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. *European Control Conference (ECC)*, 2013.

[4] Dr. Fuh. Analytic photogrammetry. Digital Camera and Computer Vision Laboratory. Department of Computer Science and Information Engineering. National Taiwan University.

[5] Rush D. Robinett Hanspeter Schaub and John L. Junkins. Adaptive external torque estimation by means of tracking a lyapunov function. 1996.

[6] F. Landis Markley. Attitude error representations for kalman filtering. In *CASI NASA Archive*.

[7] Philippe Martin. Generalized multiplicative extended kalman filter for aided attitude and heading reference system. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[8] Brent Tweddle. Computer vision based navigation for spacecraft proximity operations. Master's thesis, Massachusetts Institute of Technology, 2010.

Appendices

A Additional Mathematical Definitions

A.1 Quaternion

A unit quaternion is a four element representation of the attitude of an object. It consists of a vector and scalar elements that are related to the Euler axis and Euler angle of rotation as follows. Also as a unit quaternion, it obeys the unit length constraint.

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_v \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{e} \sin \phi/2 \\ \cos \phi/2 \end{bmatrix} \tag{82}$$

$$|\mathbf{q}|^2 = |\mathbf{q}_v|^2 + q_4^2 = 1 \tag{83}$$

Quaternion multiplication is a noncommunative operation that can be defined as either a matrix-vector product or compacted in vector notation. I have defined $\mathbf{v}_a$ and $\mathbf{v}_b$ as the vector parts of the quaternions $\mathbf{q}_a$ and $\mathbf{q}_b$ with q_4 as the scalar element of each.

$$\mathbf{q}_a \otimes \mathbf{q}_b = \begin{bmatrix} q_{a4} & -q_{a3} & q_{a2} & q_{a1} \\ q_{a3} & q_{a4} & -q_{a1} & q_{a2} \\ -q_{a2} & q_{a1} & q_{a4} & q_{a3} \\ -q_{a1} & -q_{a2} & -q_{a3} & q_{a4} \end{bmatrix} \begin{bmatrix} q_{b1} \\ q_{b2} \\ q_{b3} \\ q_{b4} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_a \times \mathbf{v}_b + q_{4a}\mathbf{v}_b + q_{4b}\mathbf{v}_a \\ q_{4a}q_{4b} - \mathbf{v}_a \bullet \mathbf{v}_b \end{bmatrix} \tag{84}$$

We also introduce the tranform from 1-2-3 Euler angles to quaternions which is needed for measurement noise generation. The solution is provided in [2].

$$\mathbf{q}_{123}(\phi, \theta, \psi) = \begin{bmatrix} c_{\phi/2}c_{\theta/2}c_{\psi/2} + s_{\phi/2}s_{\theta/2}s_{\psi/2} \\ -c_{\phi/2}s_{\theta/2}s_{\psi/2} + s_{\phi/2}c_{\theta/2}c_{\psi/2} \\ c_{\phi/2}c_{\phi/2}s_{\theta/2} + s_{\phi/2}c_{\theta/2}s_{\psi/2} \\ c_{\phi/2}c_{\theta/2}s_{\psi/2} - s_{\phi/2}s_{\theta/2}c_{\psi/2} \end{bmatrix} \tag{85}$$

A.2 Inertia Definition

The inertia tensor collects the mass moments of inertia of a rigid body in matrix form. The moment of inertia I_{xx}, I_{yy}, I_{zz} measure the resistance to rotational accelerations along the x,y,z axes. The products of inertia, the off-diagonal elements of the inertia tensor, are a measure of the induced acceleration around the second axis provided an acceleration input in the first axis. The inertia tensor is a symmetric matrix.

$$(I_{xx})_O = \int_m (y^2 + z^2)dm \qquad (I_{yy})_O = \int_m (x^2 + z^2)dm \qquad (I_{zz})_O = \int_m (x^2 + y^2)dm$$

$$(I_{xy})_O = (I_{yx})_O = \int_m (xy)dm \qquad (I_{xz})_O = (I_{zx})_O = \int_m (xz)dm \qquad (I_{yz})_O = (I_{yz})_O = \int_m (yz)dm$$

A.3 Full Linearization of the Euler Dynamics

Proceeding with the linearization about $\hat{\omega}$ of Euler's rotational dynamics, we can find the Jacobian, evaluate it at the current best estimate and multiply by the inverse of the inertia tensor to solve for the rotational acceleration. I have verified that this produces the same results as first multipying by the inverted inertia and then taking the Jacobian.

$$\dot{\omega} \approx \mathbf{J}^{-1} \begin{bmatrix} I_{xy}\omega_3 - I_{xz}\omega_2 & I_{zz}\omega_3 - I_{yy}\omega_3 - 2I_{yz}\omega_2 - I_{xz}\omega_1 & I_{zz}\omega_2 + 2I_{yz}\omega_3 - I_{yy}\omega_2 + I_{xy}\omega_1 \\ I_{xx}\omega_3 + 2I_{xz}\omega_1 + I_{yz}\omega_2 - I_{zz}\omega_3 & I_{yz}\omega_1 - I_{xy}\omega_3 & I_{xx}\omega_1 - 2I_{xz}\omega_3 - I_{xy}\omega_2 - I_{zz}\omega_1 \\ I_{yy}\omega_2 - 2I_{xy}\omega_1 - I_{xx}\omega_2 - I_{yz}\omega_3 & I_{yy}\omega_1 2I_{xy}\omega_2 - I_{xx}\omega_1 + I_{xz}\omega_3 & I_{xz}\omega_2 - I_{yz}\omega_1 \end{bmatrix}_{\boldsymbol{\omega}=\hat{\boldsymbol{\omega}}} \tag{86}$$

B Visualization

A custom visualization was developed for this project to show the user or audience how the trajectory progresses through time. A screen shot is shown in Figure 19 and a video has been presented during class. This helps in the debugging process by seeing the mean motion in real time.

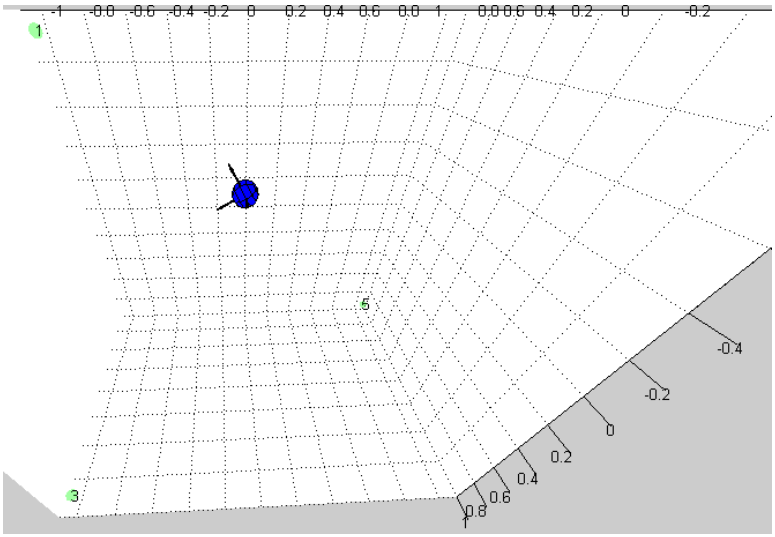

Figure 19: simulator

C Additional Figures

Additional individual figures per state have been generated and populated in the published html report located in the html subfolder.

C.1 Additional Simulation Results

C.1.1 Tracking

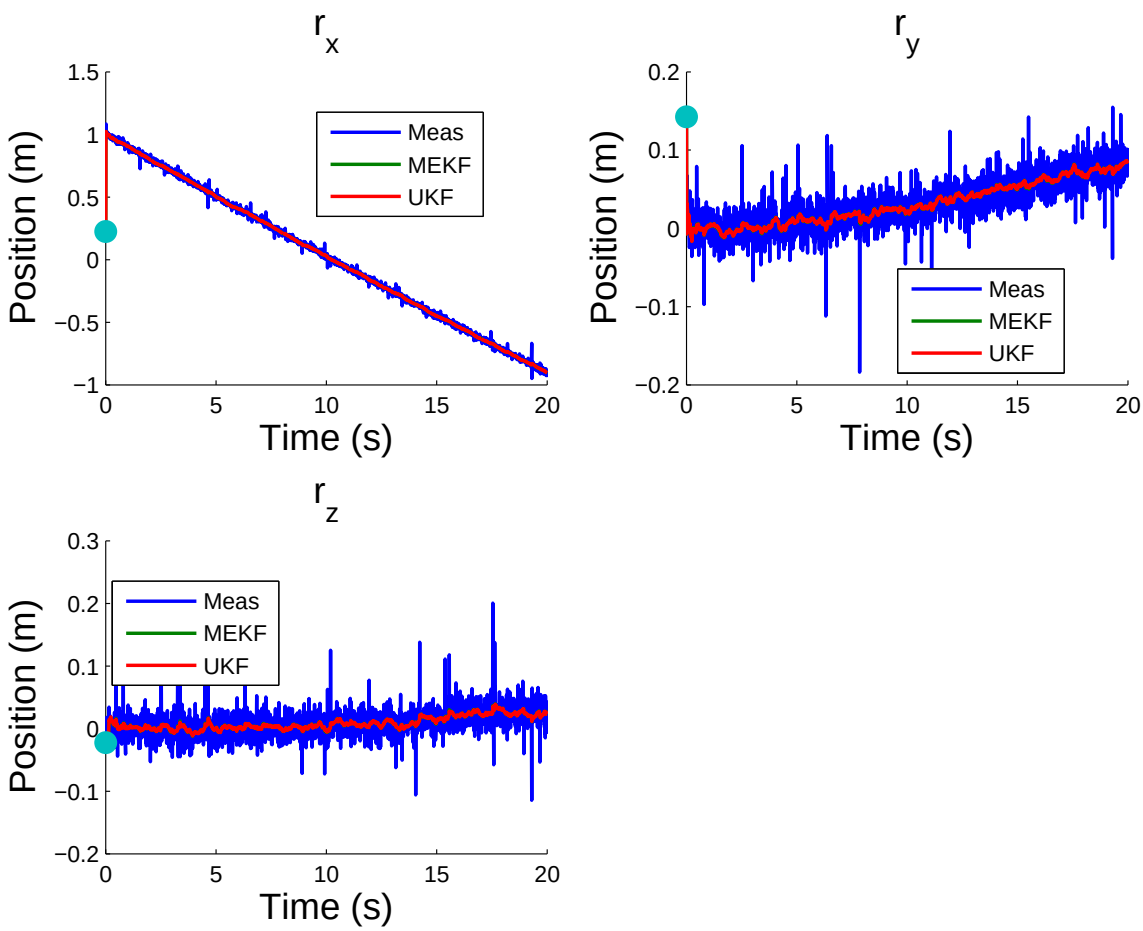

Figure 20: Representative tracking of position states with full noise modeling.

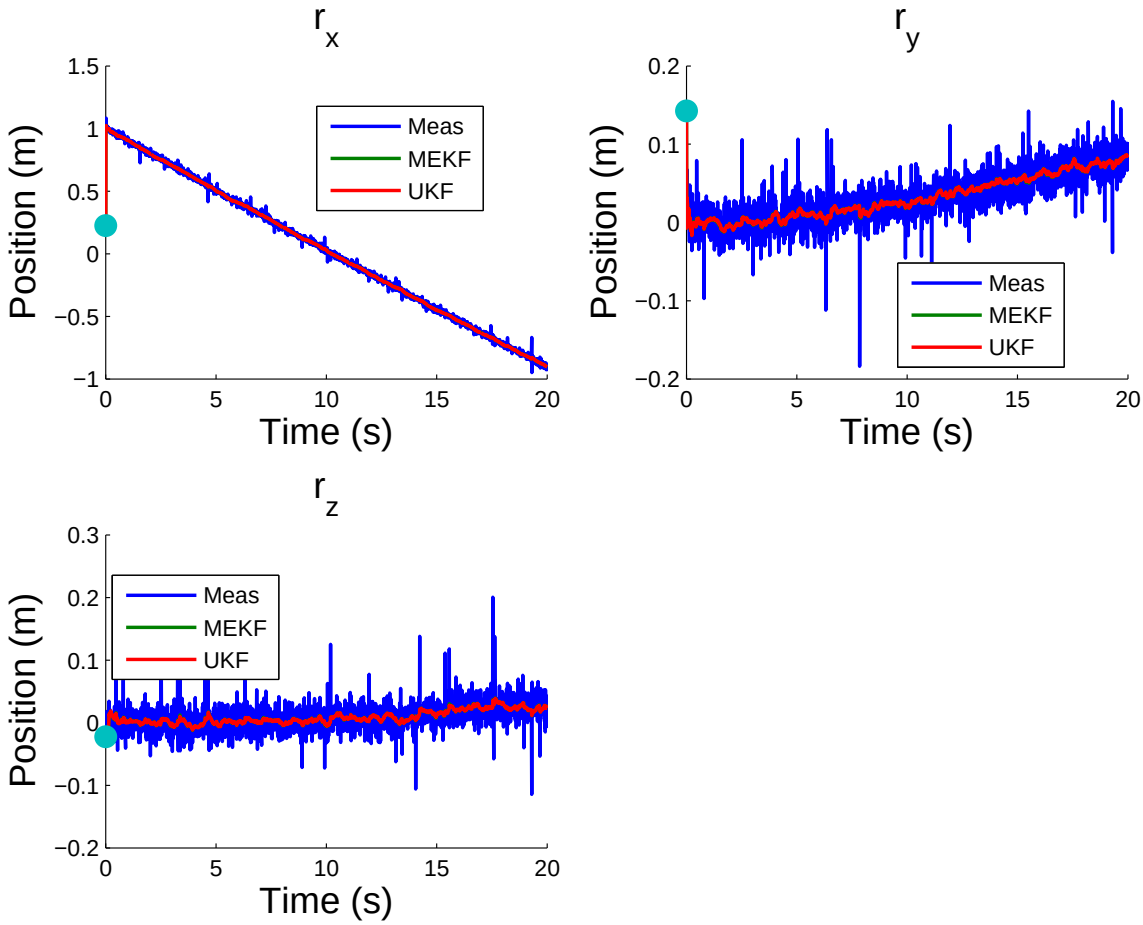

Figure 21: Representative tracking of velocity states with full noise modeling.

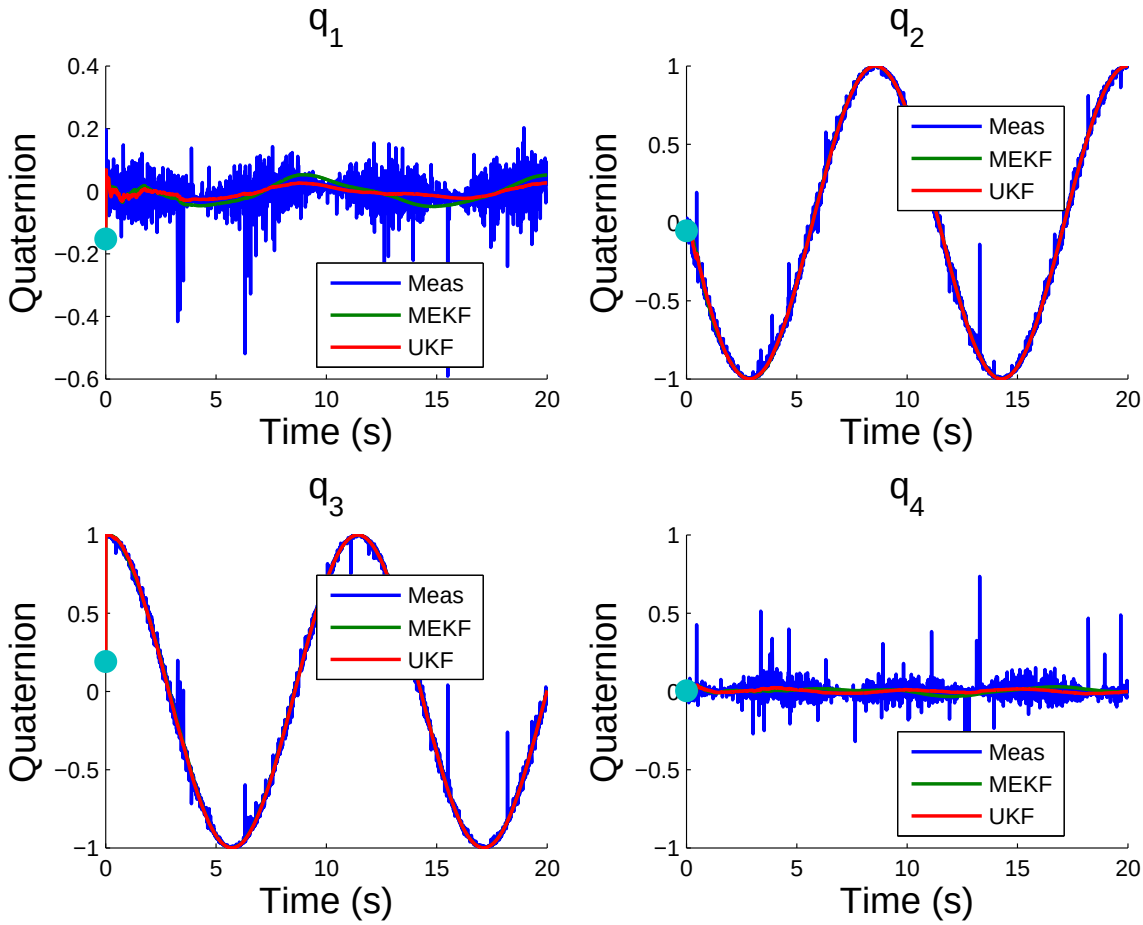

Figure 22: Representative tracking of quaternion states with full noise modeling.

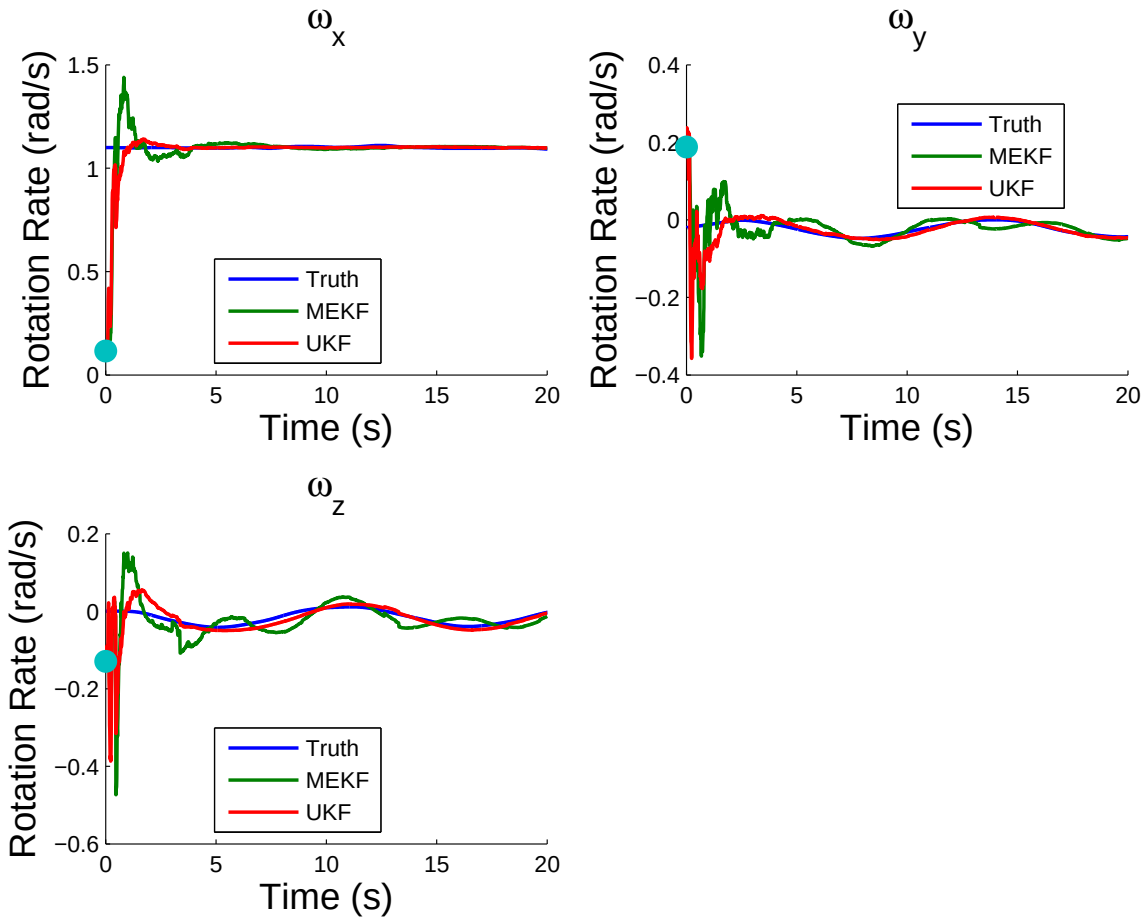

Figure 23: Representative tracking of angular velocity states with full noise modeling.

C.1.2 Errors

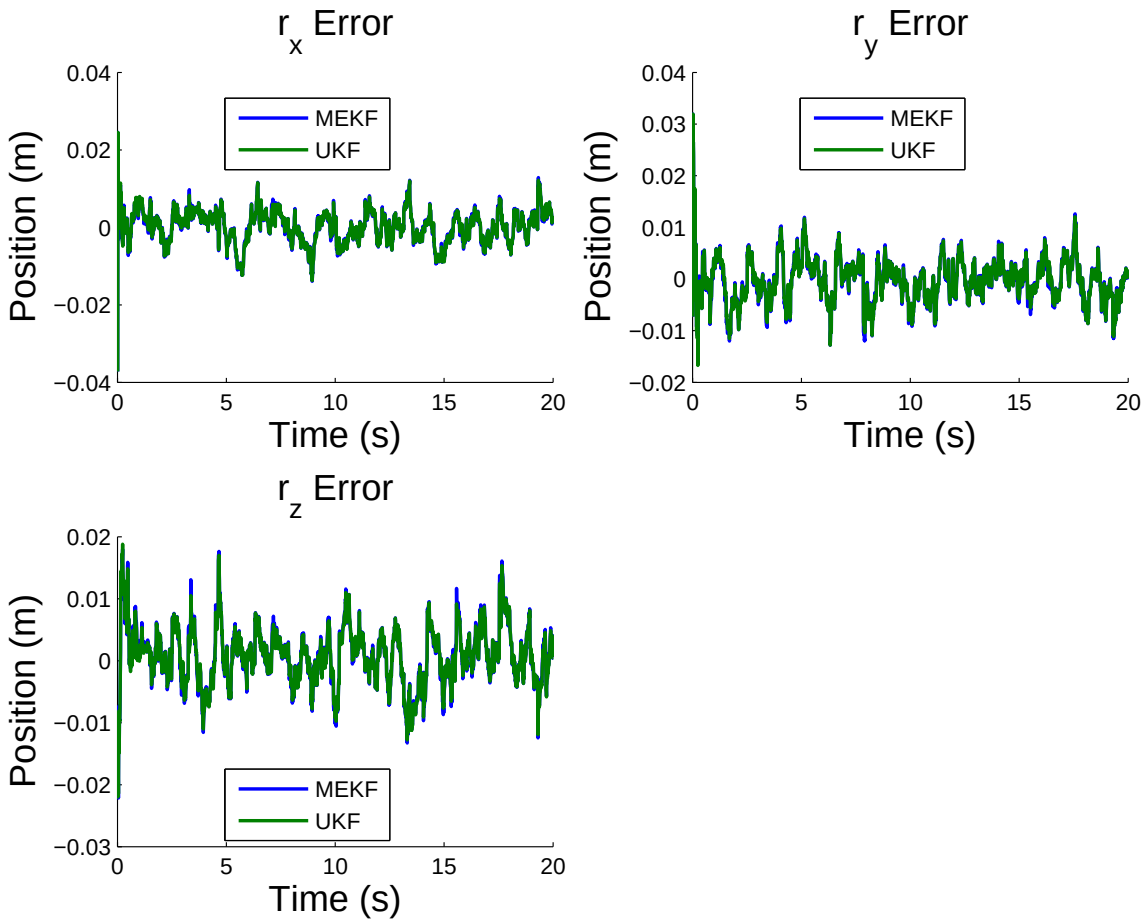

Figure 24: Representative angular velocity error with full noise modeling.

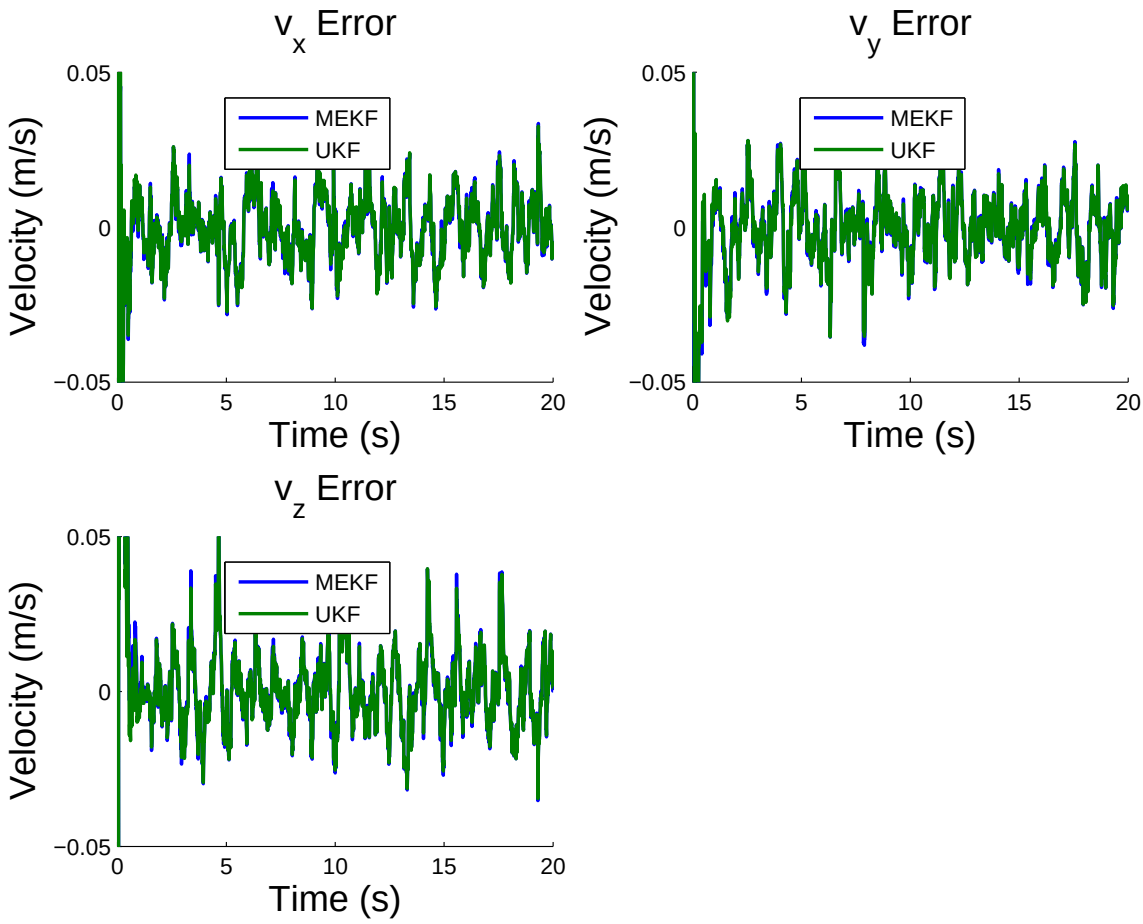

Figure 25: Representative angular velocity error with full noise modeling.

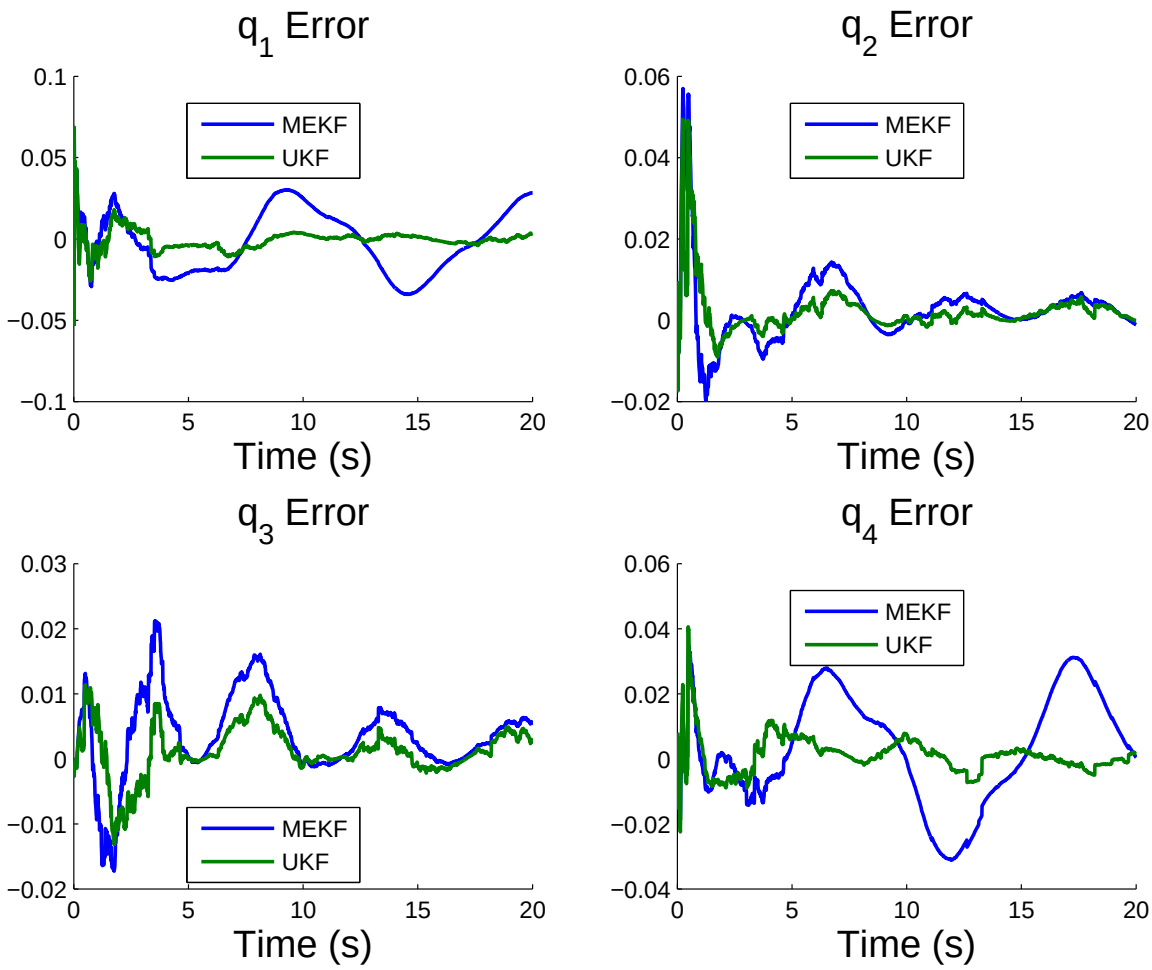

Figure 26: Representative angular velocity error with full noise modeling.

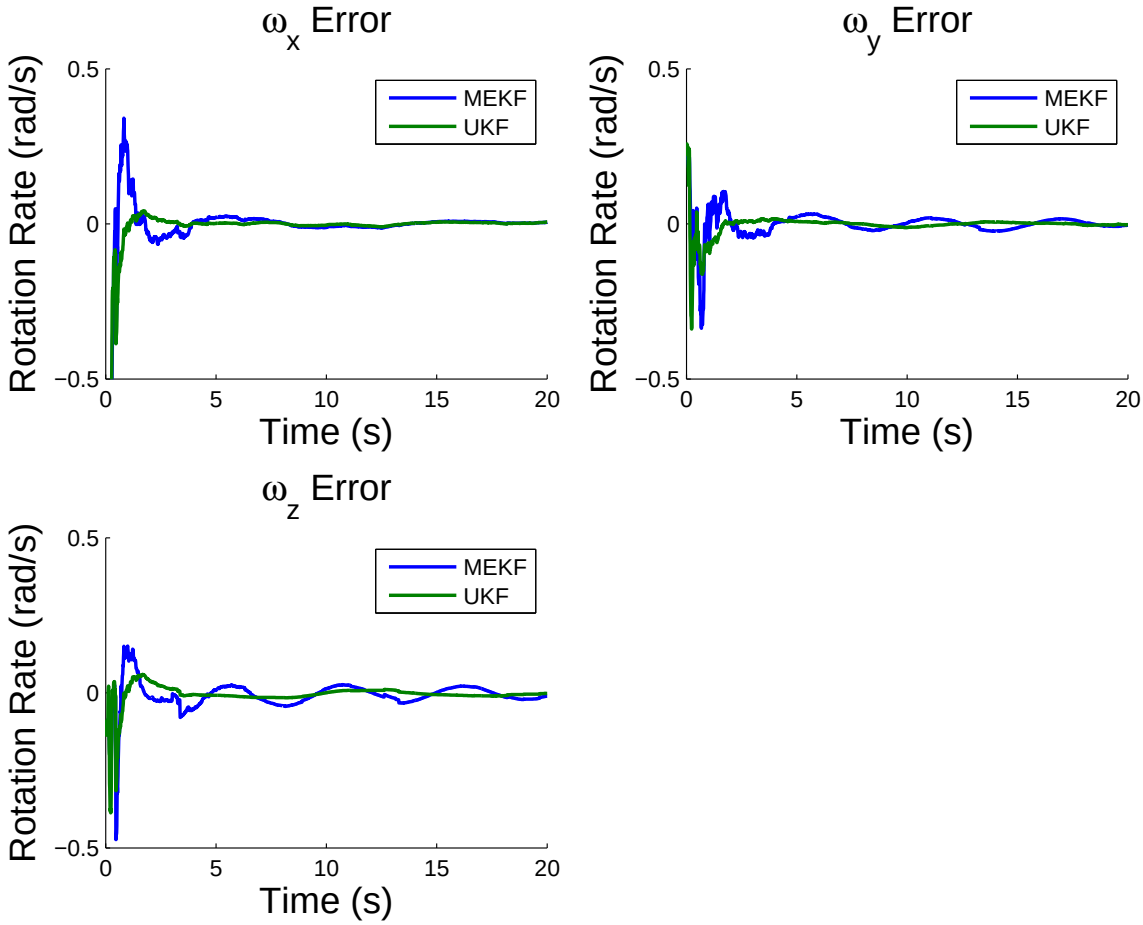

Figure 27: Representative angular velocity error with full noise modeling.

C.2 Hardware Testing Results

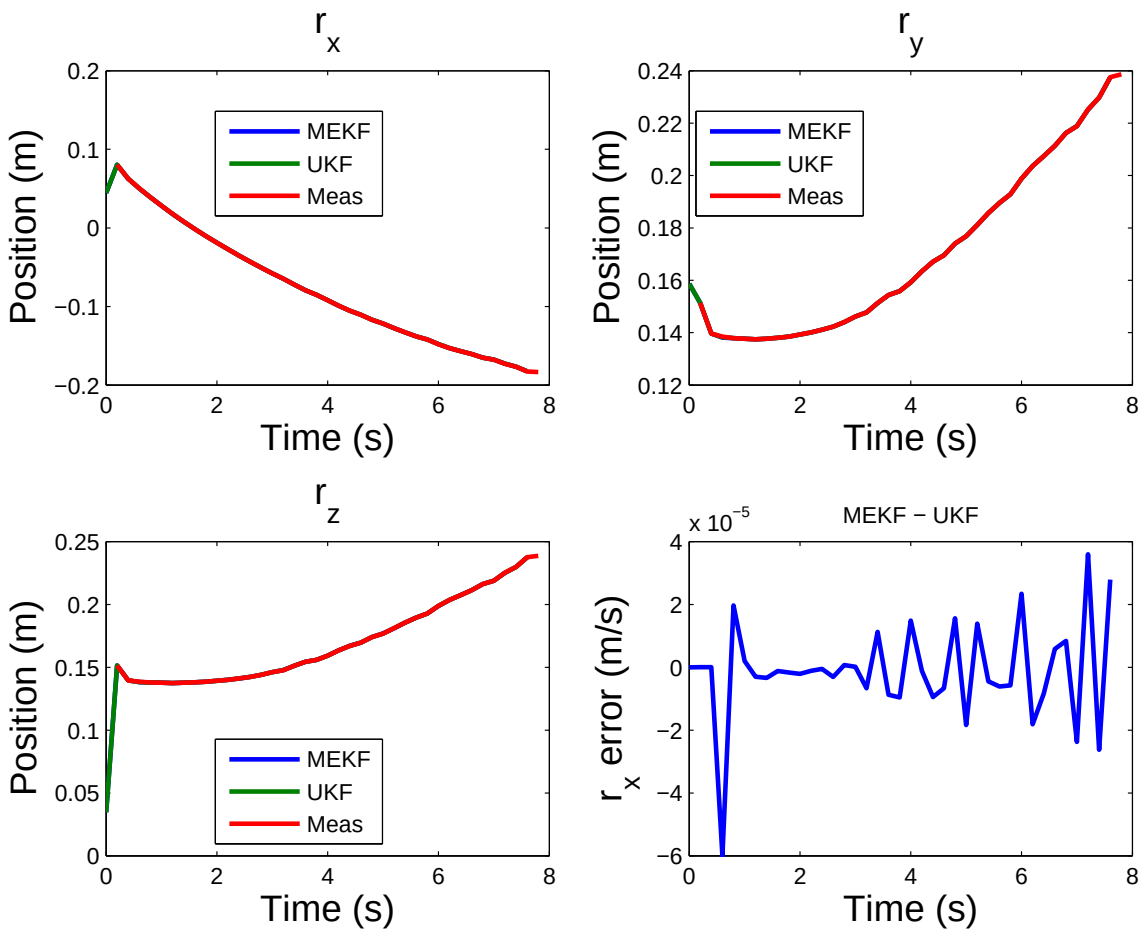

Figure 28: Position estimate of the state in the SPHERES hardware demo

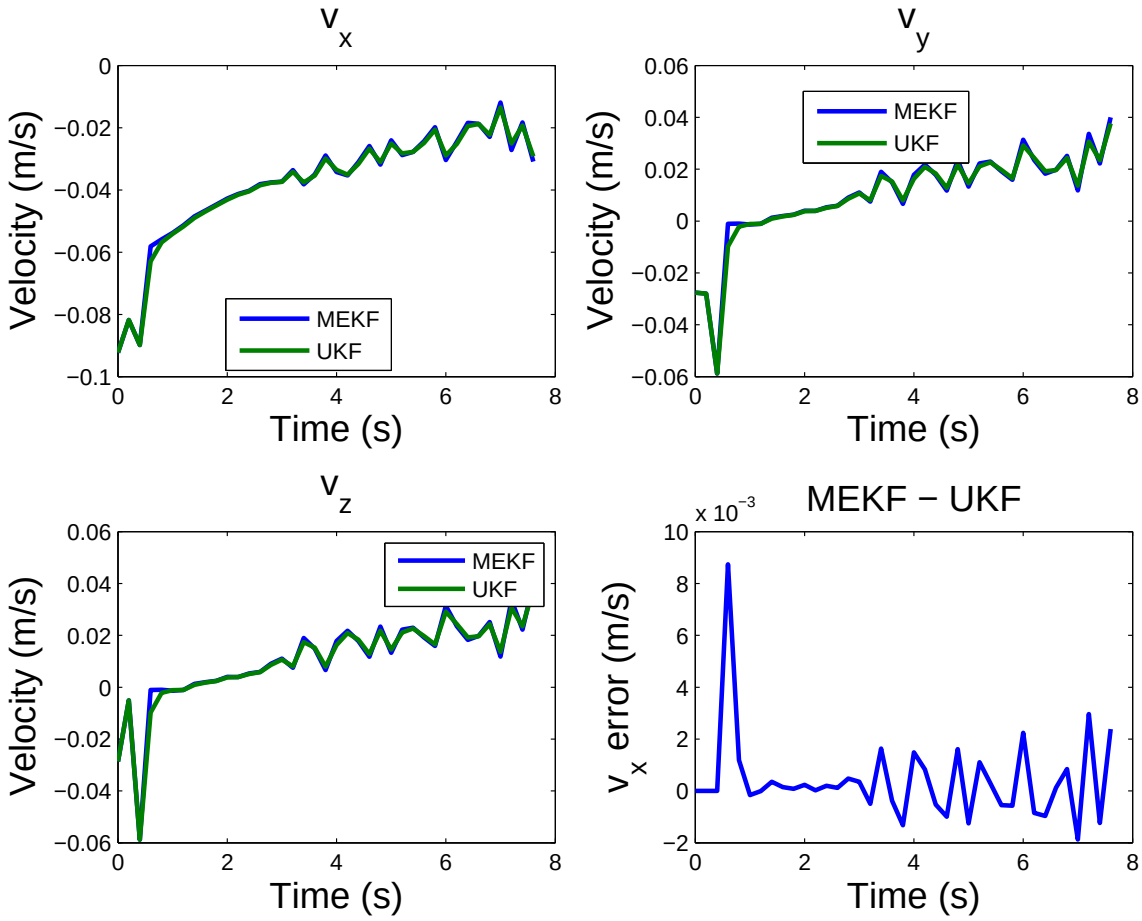

Figure 29: Velocity estimate of the state in the SPHERES hardware demo

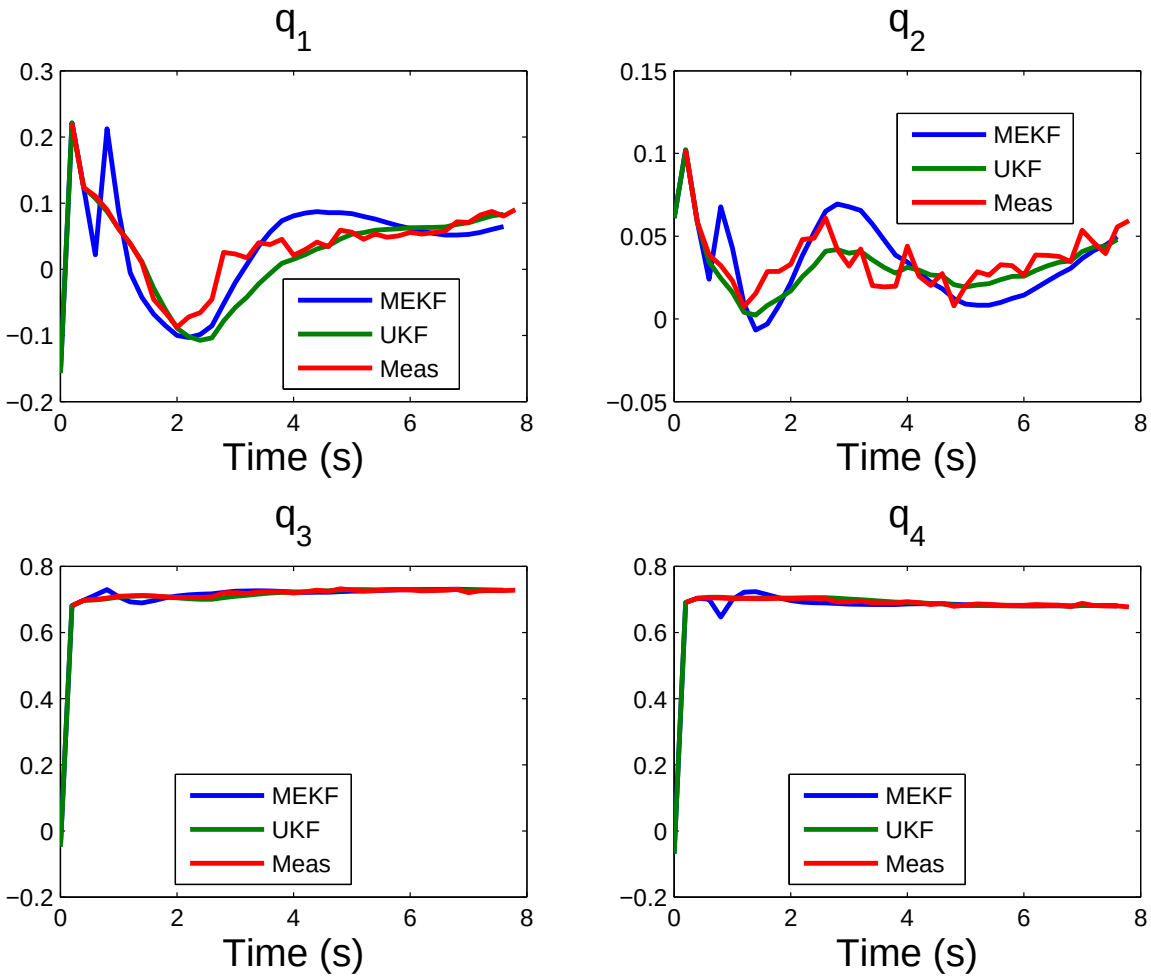

Figure 30: Quaternion estimate of the state in the SPHERES hardware demo

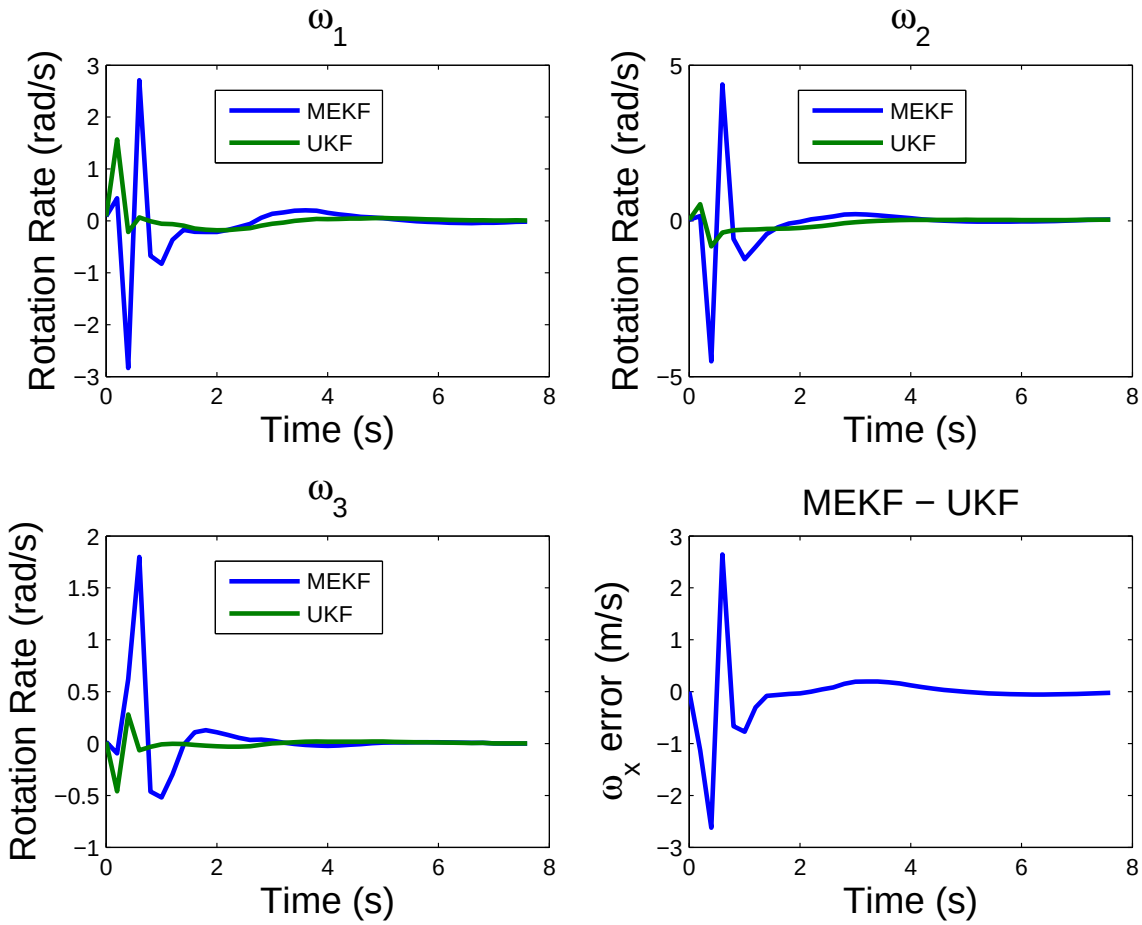

Figure 31: Angular rate estimate of the state in the SPHERES hardware demo