Spacecraft Attitude Determination Using Ground-Based Photometry: A Comparison of Estimation Algorithms

Arun J. Bernard* and David K. Geller †
Utah State University

The time history of ground-based photometry measurements, called lightcurves, can be used to determine the attitude of a spacecraft. The extraction of information about the satellite's attitude is realized through what is called lightcurve inversion. Lightcurve inversion utilizes a filter or an optimization method to extract the information on the desired parameters. The accuracy of the lightcurve inversion performance of three filters is compared. These filters are the bootstrap particle filter, the extended Kalman filter and the unscented Kalman filter. The attitude modes of a controlled spacecraft maintaining an inertially fixed attitude and spinning at constant rates are simulated. When the spacecraft is maintaining its inertial attitude, none of the filters are able to accurately estimate the satellite attitude. However, when the spacecraft was rotating the filters were able to estimate the attitude for varying degrees of initial conditions. The extended Kalman filter performs as well as the other two filters when the initial attitude uncertainty is low. When the initial attitude uncertainty increases to high levels, the bootstrap particle filter is the only filter able to accurately determine the spacecraft attitude.

I. Introduction

Photometry is a measure of how bright an object appears to an observer and is a function of the object's physical properties, the object's shape, the light source illuminating the object, and geometry. The ground-based photometry of a spacecraft is using an electro-optical sensor, or telescope, to view the spacecraft and determine its apparent magnitude, which is a quantification of how bright the object appears in the sensor image. As the perceived brightness is a function of geometry, the measurement is directly related to the attitude of the spacecraft. In recent work (see [1–3]), it has been shown that the time history of ground-based photometry measurements, called lightcurves, can be used to determine the attitude of a spacecraft.

The extraction of information about the satellite's attitude, and oftentimes shape, is realized through what is called lightcurve inversion. Lightcurve inversion utilizes a filter or an optimization method to extract the information on the desired parameters. Originally, the process was developed for asteroids. In [4] Kaasalainen et al. demonstrated the use of their optimization methods to extract information about an asteroid's pole direction, rotation rates, light-scattering parameters, and shape. However, with satellites additional complexities are introduced due to factors such as spacecraft symmetry, satellite attitude control systems, and limited observation times. To be able to handle these additional complexities, the lightcurve inversion for satellites is typically carried out using estimation algorithms or filters. The commonly used approaches include using the unscented Kalman filter (UKF) like in [5, 6], or using particle filters like those used in [1, 7].

The unscented Kalman filter and particle filters are frequently the methods of choice for lightcurve inversion as the attitude dynamics and the measurement model are both highly nonlinear. However, previous work by the authors [8] demonstrated that there may be some use for the extended Kalman filter (EKF) in performing lightcurve inversion. This would be desirable as the EKF typically performs much faster than the UKF or particle filters. Thus, the purpose of this work is to compare the abilities of these three filtering techniques in determining the attitude of a satellite.

Particle filters estimate the state using a Monte Carlo–type approach. This means that a large number of particles are used to estimate the state. In this work, the bootstrap particle filter (BPF) is used. The BPF operates by first creating a distribution of particles, then propagating the particles forward using the full nonlinear dynamics. Whenever a measurement is available, the particles are assigned weights, and these weights are used to resample the particles so that there are always the same number of particles, but they are clustered around the areas of greatest likelihood. Linares et. al. in [7], and Holzinger et. al. in [1] claim that a particle filter may be better suited to perform lightcurve inversion than a UKF, as the severe nonlinearities can become non-Gaussian. These non-Gaussian distributions would also result in poor performance of the EKF.

*Graduate student, Mechanical and Aerospace Engineering, Utah State University, 4130 Old Main Hill, Logan, UT 84322-4130
†Professor, Mechanical and Aerospace Engineering, Utah State University, 4130 Old Main Hill, Logan, UT 84322-4130

The extended Kalman filter was developed for use with nonlinear systems and is very commonly used in spacecraft guidance, navigation, and control applications. The EKF works by linearizing the state and measurements about the current state estimate and then uses these linear models to estimate the satellite state. The underlying assumption of the EKF is that these models are fairly representative of the nonlinear dynamics and measurements. When this is not the case, the EKF breaks down and is unable to determine the proper estimate, or the filter diverges. In many instances the linear models hold, and the EKF is able to accurately estimate the state.

The unscented Kalman filter is another commonly used approach for dealing with nonlinear dynamics and measurements. However, instead of linearizing the dynamics and measurement models, the UKF creates a set of points from the covariance matrix called sigma points, and these points utilize the full nonlinear dynamics and measurement models to update the state estimate. According to Van Der Merwe in [9], using the sigma points allows for the mean and the covariance to be known up to the second order terms of a Taylor series expansion of any nonlinear function. The EKF linearizes the system, and thus, only uses the first-order terms of the Taylor series expansion. This means that the UKF may provide more accurate results and perform better than the EKF in cases of extreme nonlinearities.

This work serves to compare the performance of these three filters in performing satellite attitude determination using lightcurves. The satellite dynamics will be restricted to a satellite that is being controlled by an attitude control system, and it is assumed that there are no uncertainties in the shape model or knowledge of the physical parameters of the spacecraft. The performance of the different filters are compared by evaluating the accuracy of their results. The following sections describe the models and formulations used for the dynamics, measurements, and estimation algorithms and then present the results and conclusions.

II. Dynamics models

This section presents the dynamics governing the motion of the spacecraft. These dynamics consist of two parts, the translational motion of the spacecraft in orbit and the rotational motion of the spacecraft, which is expressed through the dynamics of the attitude.

A. Orbital dynamics

The orbital motion of the spacecraft being examined in this paper is limited to motion due to the Earth's gravitational field. The differential equations describing this motion are presented below.

$$\dot{\bar{\boldsymbol{r}}} = \bar{\boldsymbol{v}} \tag{1}$$

$$\ddot{\bar{\boldsymbol{r}}} = \bar{\boldsymbol{a}}_{grav}(\bar{\boldsymbol{r}}) \tag{2}$$

In the above equations, $\bar{\boldsymbol{r}}$ and $\bar{\boldsymbol{v}}$ are respectively the position and velocity of the spacecraft, and $\bar{\boldsymbol{a}}_{grav}$ is an $N \times N$ gravity model.

B. Rotational dynamics

There a number of different ways to represent the attitude of a spacecraft (for examples, see [10]). This paper utilizes the quaternion attitude representation as it avoids some of the singularities associated with other representations. A quaternion is a four-element vector with unit norm that contains information about both the axis of rotation and the magnitude of the rotation about the angle. Here the quaternion representation contains the scalar element as the first element of the quaternion, and the vector components as the last three components. Furthermore, the earth-centered inertial (ECI) frame is the frame of reference from which the satellite body attitude is determined.

$$q_{I\to b} = \begin{bmatrix} q_0 \\ \bar{\boldsymbol{q}} \end{bmatrix} \tag{3}$$

$$\bar{\boldsymbol{q}} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \tag{4}$$

In the above expressions, the I refers to the inertial frame, and the b refers to the spacecraft body frame. Unless otherwise denoted, superscripts will be used to relate which frame the vectors are coordinatized in. The general differential equations with quaternions for the rotational motion of a spacecraft are presented below.

$$\dot{q}_{I\to b} = \frac{1}{2} \begin{bmatrix} 0 \\ \bar{\omega}^b \end{bmatrix} \otimes q_{I\to b} \tag{5}$$

$$\dot{\bar{\omega}}^b = J^{-1}[\boldsymbol{T} - \bar{\omega}^b \times J\bar{\omega}^b] + \bar{\boldsymbol{w}} \tag{6}$$

In the above equations, $\bar{\omega}$ is the angular velocity, and J is the inertia matrix of the spacecraft. The term $\boldsymbol{T}$ is the sum of the torques acting on the spacecraft, including gravity gradient, aerodynamic drag, and control torques, and w is a random angular acceleration term with power spectral density, Q_w.

Equations 5 and 6 are the full rotational dynamics equations for any space object. However, satellites are often capable of maintaining their own attitude through an attitude control system. For simulation purposes each spacecraft is defined as having an operational control system. This means that the satellite will maintain its attitude as either an inertial hold, where its attitude stays fixed, or rotating at a constant rate. The differential equations that describe this controlled motion are presented below.

$$\dot{q}_{I\to b} = \frac{1}{2}\begin{bmatrix} 0 \\ \bar{\omega}^b \end{bmatrix} \otimes q_{I\to b} \tag{7}$$

$$\dot{\bar{\omega}}^b = \mathbf{0} \tag{8}$$

III. Measurement Model

The attitude of the spacecraft is estimated by analyzing lightcurves, which are the time history of the apparent magnitude of the object. The governing equation for the calculating the apparent magnitude of the spacecraft is:

$$M_\Lambda = -2.5\log_{10}\left(\frac{I_\Lambda}{\Im_{sun}}\right) - 26.74 \tag{9}$$

Where $\Im_{sun}$ is the average illumination intensity of the sun at a given distance. The apparent magnitude of the satellite is dependent upon the amount of light that is reflected by the satellite. To simplify calculations, it is assumed that the satellite shape is convex, and the shape of the satellite is represented by a series of flat plates or facets. Each facet is defined by an area, A_i, and a unit normal vector, $\hat{\boldsymbol{n}}_i$. Using the facet representation, the reflection can be calculated for each of the individual facets and then summed to give the total reflection. Figure 1 illustrates the reflection geometry for a given facet.

In this figure, the vector $\bar{\boldsymbol{v}}_r$ is the satellite to observer vector, $\bar{\boldsymbol{s}}$ is the satellite to sun vector, $\bar{\boldsymbol{h}}$ is the bisector of these two vectors, and $\hat{\boldsymbol{n}}_i$ is the unit normal of the facet. With these parameters it is then possible to calculate photometric flux, I_Λ, in equation 9.

The term I_Λ is the calculated photometric flux received by the EO sensor from the satellite at a given time. The amount of flux that is being captured by a sensor is dependent upon the amount of light that is being reflected by the object. Under the assumption that the light reflects equally on all wavelengths over a particular wavelength band, the photometric flux can be found using equation 10 below.

$$I_\Lambda = \frac{1}{\bar{\boldsymbol{v}}_r^T\bar{\boldsymbol{v}}_r}\Im_{sun}(\bar{\boldsymbol{s}})\sum_{i=1}^{N_f} A_{i,vis}\rho_i\left(\hat{v}_r^b, \hat{s}^b, p_i,\right) \tag{10}$$

In the above equation, ρ_i is the total reflectance of the object which is dependent upon the reflection geometry and the physical reflection properties of the given facet represented by p_i. The terms properties of p_i include the specular/diffuse reflection weight, diffuse albedo term, and the surface roughness parameter of the facet. The vector $\bar{\boldsymbol{v}}_r^b$ is the satellite to observer vector and $\bar{\boldsymbol{s}}^b$ is the satellite to sun vector, both expressed in the satellite body frame.

The term $A_{i,vis}$ is the visible projected area of the facet. It is found using the following equations:

$$A_{i,vis} = A_i\langle\hat{\boldsymbol{n}}_i\cdot\hat{\boldsymbol{v}}_r\rangle \tag{11}$$

where $\langle\cdot\rangle$ is the nonnegative operator defined as

$$\langle x\rangle = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{12}$$

This term ensures that the light reflected from the spacecraft is visible to the observer.

Bidirectional reflectance distribution function

A bidirectional reflectance distribution function (BRDF) is a model that is used to describe the reflection from a facet surface given the direction of the impinging light and the direction of the observer viewing the reflected light. Here, the BRDF model introduced by Cook et al. for computer modeling is utilized [11]. The reflection is found for each individual facet (denoted by i) and then summed to give the total reflectance of the satellite, which is determined by a weighted combination of a specular and a diffuse reflection component as shown in the equation below.

$$\rho_i = \xi_i R_d + (1-\xi_i)R_s \tag{13}$$

The term ξ is the prescribed specular/diffuse weight of the facet. The diffuse reflection is modeled using a Lambertian reflection model where light reflects equally in all directions.

$$R_d = \frac{a_i(\hat{\boldsymbol{n}}^b\cdot\hat{\boldsymbol{s}}^b)}{\pi} \tag{14}$$

In the above equation, a_i is the diffuse albedo term for the ith facet.

The specular reflection is determined using the following equation.

$$R_s = \frac{F}{4}\frac{D}{(\hat{\boldsymbol{n}}^b\cdot\hat{\boldsymbol{s}}^b)}\frac{G}{(\hat{\boldsymbol{n}}^b\cdot\hat{\boldsymbol{v}}^b)} \tag{15}$$

Where F is the Fresnel equation, D is the facet slope distribution function, and G is the geometric attenuation factor. Each of these terms will be described below with the accompanying equations.

The Fresnel equation, F, is used to describe the reflectance of a surface.

$$F = \frac{1(g_i-c)^2}{2(g_i+c)^2}\left[1+\frac{(c(g_i+c)-1)^2}{(c(g_i+c)+1)^2}\right] \tag{16}$$

$$c = \hat{\boldsymbol{v}}^b\cdot\hat{\boldsymbol{h}}^b, \quad g_i^2 = n_i^2 + c^2 - 1 \tag{17}$$

$$n_i = \frac{1+\sqrt{F_{0,i}}}{1-\sqrt{F_{0,i}}} \tag{18}$$

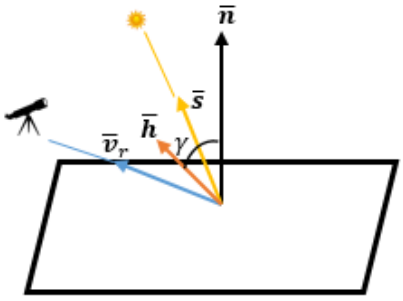


Fig. 1 Reflection geometry for a given facet of the spacecraft shape model

In the above equations, $F_{0,i}$ is the Fresnel equation at zero angle of incidence, and n_i is the index of refraction.

The facet slope distribution function, D, is a term that is used to determine the specular reflection due to a rough surface. According to Cook et al. in [11], it represents how much of the surface of the facet is oriented in the direction of the bisector $\hat{\boldsymbol{h}}_i$. To calculate this term, the Beckmann distribution function from [12] is used. For the observer to best view the specular reflection of a perfectly smooth surface, the facet unit normal has to be aligned with the bisector defined in Figure 1. The Beckmann distribution is a function of the angle γ between the bisector and the unit normal, and the slope of the roughness of the surface, m. This distribution term, D, is calculated as follows

$$D = \frac{1}{\pi m_i^2 cos^4\gamma_i} e^{-\left(\frac{tan^2\gamma_i}{m_i^2}\right)} \tag{19}$$

where

$$\gamma_i = cos^{-1}(\hat{\boldsymbol{n}}_i^b \cdot \hat{\boldsymbol{h}}^b) \tag{20}$$

The final term in Equation 15 is the geometric attenuation factor, G. This term accounts for the self-shadowing of the surface due to roughness.

$$G = min\left\{1, \frac{2cos\gamma_i(\hat{\boldsymbol{n}}^b \cdot \hat{\boldsymbol{v}}_r^b)}{(\hat{\boldsymbol{v}}_r^b \cdot \hat{\boldsymbol{h}}^b)}, \frac{2cos\gamma_i(\hat{\boldsymbol{n}}^b \cdot \hat{\boldsymbol{s}}^b)}{(\hat{\boldsymbol{v}}_r^b \cdot \hat{\boldsymbol{h}}^b)}\right\} \tag{21}$$

In implementing these equations in simulation, a simplifying assumption is made for the diffuse albedo term, $a_i = F_{0,i}$. For justification of this assumption refer to [1].

IV. Estimation Algorithms

The primary purpose of this paper is to compare the accuracy of three estimation algorithms in extracting attitude information through lightcurve inversion. The objective of this section is to present the information needed to use these algorithms for this objective. However, before going into the individual estimation methods, there are some definitions and explanations common to all of the algorithms that need to be presented. Thus, the following sections will first present these items, and then the specific algorithms.

A. Truth model dynamics

Using simulation to determine the effectiveness of an estimation algorithm requires definition of the truth model that will be used to gauge filter performance. The different estimation methods require that there be some models for the truth dynamics and also the uncertainty associated with the dynamics and measurement models.

In section II.B it was explained that the satellites under consideration are all equipped with an attitude control system (ACS). This means that in general, the satellite will follow a commanded trajectory. This is expressed in the differential equations

$$\dot{q}^*_{,I\to b} = \frac{1}{2}\begin{bmatrix}0\\ \bar{\omega}^b\end{bmatrix} \otimes q^*_{I\to b} \tag{22}$$

$$\dot{\bar{\omega}}^b = \mathbf{0} \tag{23}$$

Here the $*$ is used to signify the commanded attitude. However, no ACS operates perfectly. There are random perturbations that need to be rejected and biases needing to be corrected. This means that at any given time there is some range (determined by the ability of the ACS) in which the true attitude differs from the commanded attitude. To represent this in simulation, the commanded attitude is propagated forward in time, and the true attitude is computed at discrete intervals by perturbing the commanded attitude values by some small amount as shown in the equation below.

$$q_{I\to b} = \begin{bmatrix}1\\ \boldsymbol{w_d}/2\end{bmatrix} \otimes q^*_{I\to b} \tag{24}$$

Here $\boldsymbol{w_d}/2$ is a discrete noise term with strength given by $E[\boldsymbol{w_d}(i)\boldsymbol{w_d}(j)^T] = Q_{w_d}\delta(t_i - t_j)$ where δ is the Kronecker delta function.

This approach is used to ensure that the covariance matrix of the estimation algorithms doesn't grow due to process noise, but remains within a set value like what would be expected from an operational ACS.

The apparent magnitude measurements, $\tilde{y}$ used in the filters are all taken from this true state.

$$\tilde{\boldsymbol{y}} = h(\boldsymbol{x}_k, k) + \nu_k = M_\Lambda(q_{I\to b,k}, k) + \nu_k \tag{25}$$

The strength of the measurement noise, R, is given as $E[\nu_j \nu_k^T] = R\delta(t_j - t_k)$, where again δ is the Kronecker delta function.

B. Error vector representations

As stated previously in section II.B, the attitude representation used is the quaternion as it proves an efficient means of representing the attitude while avoiding singularities associated with other representations. However, a defining feature of the quaternion attitude representation is that it requires a unit norm. As will be shown in the following sections, calculating or applying the state updates is done in an additive manner. This can result in the quaternion unit norm constraint being violated. To prevent this from occurring, a "local" small angle error vector attitude representation is used inside the filters while maintaining a "global" quaternion attitude representation as the state estimate.

The filter attitude error states are represented using the generalized Rodrigues parameters (GRPs). Linares et al. in [13], express that this representation yields a minimum parameter representation for the attitude state, and the error covariance is the same to the first order using this representation or quaternions. A quaternion can be expressed as a GRP using the following equation:

$$\bar{\boldsymbol{\theta}} = f\frac{\bar{\boldsymbol{q}}}{a + q_0} \tag{26}$$

where a is a parameter from 0 to 1, and f is a scale factor commonly set to $f = (2(1 + a))$ [13]. Computing quaternions from GRPs can be done using the following equations.

$$q_0 = \frac{-a\|\bar{\boldsymbol{\theta}}\|^2 + f\sqrt{f^2 + (1 - a^2)\|\bar{\boldsymbol{\theta}}\|^2}}{f^2 + \|\bar{\boldsymbol{\theta}}\|^2} \tag{27}$$

$$\bar{\boldsymbol{q}} = f^-1\,[a + q_0]\,\bar{\boldsymbol{\theta}} \tag{28}$$

$$q = \begin{bmatrix} q_0 \\ \bar{\boldsymbol{q}} \end{bmatrix} \tag{29}$$

A second representation is used to express the error between the truth model and the estimated state for the results. This other representation is a rotation vector. The rotation vector can be computed from a quaternion using the following equations.

$$\bar{\boldsymbol{e}} = \frac{\bar{\boldsymbol{q}}}{\|\bar{\boldsymbol{q}}\|} \tag{30}$$

$$\psi = 2\arctan\left(\frac{\|\bar{\boldsymbol{q}}\|}{q_0}\right) \tag{31}$$

$$\bar{\boldsymbol{\psi}} = \psi\bar{\boldsymbol{e}} \tag{32}$$

The conversion to a quaternion from the rotation vector is found using the following equations.

$$\psi = \|\bar{\boldsymbol{\psi}}\| \tag{33}$$

$$\bar{\boldsymbol{e}} = \frac{\bar{\boldsymbol{\psi}}}{\psi} \tag{34}$$

$$q = \begin{bmatrix} \cos(\psi/2) \\ \sin(\psi/2)\bar{\boldsymbol{e}} \end{bmatrix} \tag{35}$$

C. Bootstrap particle filter

A particle filter is an estimation method that is used to recursively estimate the state variables by creating a distribution of particles that are weighted with the measurements to represent the actual state that is being estimated. The Bootstrap particle filter (BPF) was developed by Gordon et. al. [14]. It is also known as the sample importance resample filter. This filter first samples the given distribution to place the particles such that they represent the current knowledge of the true probability density function. Then the particles are propagated forward in time through the full nonlinear differential equations, and when a measurement is available the filter assigns weights and determines the particles of highest likelihood. Then after this, it uses what is called a resampling algorithm. This involves the reassigning of particle values based on those particles that have the largest weights so that the same number of particles is kept at each update phase of the filter, but they are grouped around the area of greatest likelihood.

The BPF is especially suited to nonlinear state estimation problems. In [1], Holzinger et. al. suggest that the BPF is very well suited for the lightcurve inversion problem as the measurements and dynamics are exceedingly nonlinear and non-Gaussian in nature.

This following paragraphs outline the formulation of the BPF for use in the lightcurve inversion problem. The majority of the algorithm is taken as presented in [15] with the resampling algorithm implemented as shown in [14].

The state that is being estimated consists of the attitude quaternion and the angular velocity terms.

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{\boldsymbol{q}}_{I\to\hat{b}} \\ \hat{\boldsymbol{\omega}}^b \end{bmatrix} \tag{36}$$

The measurement model for the filter is shown in equation 25

1. Propagation

The first step in implementing the particle filter is to create a distribution of particles about the initial estimate of the state, $\hat{\boldsymbol{x}}_0$, using the initial covariance, $\boldsymbol{P}_0$.

The state estimate is the expected value of the commanded trajectory and so the differential equations for the particle propagation are shown in the following equations.

$$\begin{bmatrix} \dot{\hat{\boldsymbol{q}}}_{I\to\hat{b}} \\ \dot{\hat{\boldsymbol{\omega}}}^b \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\begin{bmatrix} 0 \\ \bar{\boldsymbol{\omega}}^b \end{bmatrix} \otimes \hat{\boldsymbol{q}}_{I\to\hat{b}} \\ 0 \end{bmatrix} \tag{37}$$

When no measurements are available, the state estimate is just the propagated value of the state estimate at the previous time. However, the covariance matrix for any given time is calculated using the following approach. First the error between the particle values and the state estimate is calculated. Note that the frame indices are dropped for ease in writing.

$$\delta\boldsymbol{x}_q = \begin{bmatrix} \delta q \\ \delta\omega \end{bmatrix} = \begin{bmatrix} q^{(i)} \otimes \hat{q}^{-1} \\ \omega^{(i)} - \hat{\omega} \end{bmatrix} \tag{38}$$

Here the i refers to the ith particle and $\hat{q}^{-1}$ is the quaternion conjugate of the attitude estimate. The general equation to find the conjugate of a quaternion is shown below.

$$q^{-1} = \begin{bmatrix} q_0 \\ -\bar{\boldsymbol{q}} \end{bmatrix} \tag{39}$$

Then all of the δq terms are converted to error GRPs using equation 26 which gives $\delta\boldsymbol{x}_\theta$. The covariance matrix is then calculated using the standard statistical approach.

2. Update and resampling

In order to account for the deviations of the truth model from the commanded trajectory at a given time, whenever a measurement is available, the attitude state of the individual particles is perturbed by a process noise sample drawn from $N(0, Q_{w_d})$. This process noise is added in the same manner as equation 24. Then to process the measurement, the residuals are first found by subtracting the computed measurements of the individual particles from the available measurement.

$$w_k^i = \tilde{y}_k - h_k\left(\hat{\boldsymbol{x}}_k^i, t_k\right) \tag{40}$$

It is assumed that the measurement residual is Gaussian in nature, and so a Gaussian distribution is used to assign weights to the particles. Note that the terms scaling the exponential are dropped as the whole thing will be normalized later.

$$\tilde{w}_k^i = e^{-\frac{1}{2}\left(w_k^{i\,T} R_k^{-1} w_k^i\right)} \tag{41}$$

where R_k is the variance of the measurement. The weights $\tilde{w_k}^i$ are then normalized as shown below:

$$W_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i} \tag{42}$$

Where N is the number of particles. After the weights have been assigned, the updated estimate of the mean and the covariance are calculated before resampling the particles. As this, by definition, changes the distribution of the particles. To do this, the first step is to find the error quaternions of all of the particles compared to the current state estimate.

$$\boldsymbol{x}_{\delta q} = \begin{bmatrix} \delta q \\ \omega \end{bmatrix} = \begin{bmatrix} q^{(i)} \otimes (\hat{q}^-)-1 \\ \omega^{(i)} \end{bmatrix} \tag{43}$$

The error quaternions are then converted to error GRPs using equation 26. This gives $\boldsymbol{x}_{\delta\theta}$. Then the mean and covariance can be calculated using the following equations taken from [16]. A new estimate value is calculated using the weights and the particles.

$$\hat{\boldsymbol{x}}_{\delta\theta,k} = \begin{bmatrix} \delta\hat{\boldsymbol{\theta}} \\ \hat{\omega}^+ \end{bmatrix} = \sum_{i=1}^N W_k^{(i)} \boldsymbol{x}_{\delta\theta,k}^{(i)} \tag{44}$$

The difference of each of the particles from the mean is determined and used to calculate the updated covariance.

$$\tilde{\boldsymbol{x}}_{\delta\theta,k}^{(i)} = \boldsymbol{x}_{\delta\theta,k}^{(i)} - \hat{\boldsymbol{x}}_{\delta\theta,k} \tag{45}$$

$$P_k \approx \sum_{i=1}^N W_k^{(i)} \tilde{\boldsymbol{x}}_{\delta\theta,k}^{(i)} \left(\tilde{\boldsymbol{x}}_{\delta\theta,k}^{(i)}\right)^T \tag{46}$$

The mean estimate is then converted back into a quaternion by first converting $\hat{\boldsymbol{x}}_{\delta\theta,k}$ back into an error quaternion, $\delta\hat{q}_k$, using equations 27–29. Then the new quaternion estimate is found using

$$\hat{q}^+ = \delta\hat{q}_k \otimes \hat{q}^- \tag{47}$$

and the *a posteriori* state estimate is

$$\hat{\boldsymbol{x}}_k^+ = \begin{bmatrix} \hat{q}^+ \\ \hat{\omega}^+ \end{bmatrix}. \tag{48}$$

After the updated mean estimate and covariance are calculated, the particles are resampled using the method presented in Gordon et al. [14].

3. Particle roughening

A key feature of the BPF is that it is driven by process noise. In cases where there is little to no process noise, the filter does not perform well. Additionally, the BPF is prone to what is known as the impoverishment problem. The impoverishment problem is what occurs when the weights are not very well distributed and so the filter narrows in on a solution space of only one or two particles. This results in obtaining an erroneous answer that is output with very high certainty. To avoid this issue, an additional step called particle roughening may be used. Particle roughening jitters the particles with another set of noise, which results in changing the distribution so that they aren't all focused on a single particle. The method used for roughening the particles is taken from [16].

To implement this roughening, an independent jitter is drawn from a Gaussian distribution, $N(0, J_k)$. The matrix J_k is a diagonal matrix where the diagonal elements comprise $\sigma_1^2 ... \sigma_n^2$. The lth standard deviation is given by

$$\sigma_l = GE_l N^{-1/n} \tag{49}$$

where E_l is the length of the interval between the maximum and minimum values of this component (before roughening), n is the dimension of the state space, and N is the number of particles. G is a tuning parameter that is used to determine how much noise is needed.

When computing these values of the jitters, it is important to note that the quaternion components of the particles need to be converted to error GRPs using equation 26 and the *a posteriori* estimate of the quaternion. Then the above equations can be used to calculate the jitters. Finally, the jitters are applied by converting the GRP components of the jitters to error quaternions and applying the noise in a similar manner to the update in equation 47. The ω components of the jitters are just added to the ω components of the different particles.

D. Extended Kalman filter

The extended Kalman filter (EKF) is an estimation technique that is widely used for nonlinear state estimation. The EKF operates by linearizing the state dynamics and measurements about the current estimate. Then using measurements that are weighted based off of their accuracy, a gain is calculated that updates the state and covariance to better match the truth. The EKF is less expensive computationally than the particle filter, but the filter may cease to function when the linear dynamics and measurements approximations don't closely match the nonlinear dynamics and measurements.

The multiplicative extended Kalman filter (MEKF) is a form of the Kalman filter that has been modified to account for the quaternion as one of the states being estimated. As explained in Markley and Crassidis [10], the standard update equation for the extended Kalman filter (EKF) is additive. However, a quaternion is defined as having a magnitude of 1. Adding the standard update can change the magnitude of the quaternion, so the MEKF is an EKF that has been adapted to update the quaternion in a multiplicative manner. This is done by assuming that the current estimate of $\hat{\boldsymbol{q}}$ is related to the true $\boldsymbol{q}$ by a small angle rotation vector $\boldsymbol{\delta\theta}$. The manner in which this is done is outlined in the following sections.

1. Propagation

The state for the propagation portion of the MEKF is the full quaternion and angular velocity vector. These are propagated forward using Equation 37. However, the state that goes into the update portion of the filter is:

$$\boldsymbol{\delta x} = \begin{bmatrix} \boldsymbol{\delta\theta} \\ \boldsymbol{\delta\omega} \end{bmatrix} \tag{50}$$

The state error covariance is propagated using equation 51.

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_d \tag{51}$$

Where Q_d is a very small discrete random noise that is applied only to keep the filter from "becoming overconfident" and rejecting measurements. As stated previously, with the controlled satellite, it is not desirable that there is process noise that greatly increases the covariance of the estimate. It should just grow based on the initial uncertainties in the dynamics. The term Φ is the state transition matrix found from the linearized dynamics of the system. It is calculated as:

$$\Phi = e^{F\Delta T} \tag{52}$$

where the Jacobian $\boldsymbol{F}$ for the controlled satellite is

$$F = \begin{bmatrix} [-\bar{\omega}\times] & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}\Bigg|_{\hat{\boldsymbol{x}}}. \tag{53}$$

In the previous equation, the brackets $[\times]$ are used to signify the cross product matrix of a vector.

2. Update

The state and state error covariance is updated as follows: First the Kalman gain is calculated.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + H_k Q_{w_d,k} H_k^T + R_k)^{-1} \tag{54}$$

The term R is the variance of the measurement error, the $H_k Q_{w_d,k} H_k^T$ is used to map the noise that perturbs the commanded quaternion trajectory to the truth model into the measurement space, and the measurement partial, H, is given by

$$\boldsymbol{H} = \left(\frac{\partial h}{\partial \boldsymbol{x}}\right)_{\boldsymbol{x}_k^-} = \begin{bmatrix} \frac{\partial M_\Lambda}{\partial \boldsymbol{\delta\theta}}_{1\times 3} & 0_{1\times 3} \end{bmatrix}. \tag{55}$$

As the measurement model is very complex, these partial derivatives are computed numerically.

The state correction is computed using the Kalman gain and the measurement residual,

$$\boldsymbol{\delta x}_k = K_k \left(\tilde{y}_k - \hat{y}_k\right), \tag{56}$$

and the quaternion and angular velocity are updated using the following equations.

$$\hat{\boldsymbol{q}}_k^+ = \delta q(\hat{\delta\boldsymbol{\theta}}_k) \otimes \hat{q}_k^- \tag{57}$$

$$\hat{\boldsymbol{\omega}}_k^+ = \hat{\boldsymbol{\omega}}_k^- + \hat{\delta\boldsymbol{\omega}}_k \tag{58}$$

The δq term in equation 57 is found by converting the $\hat{\delta\boldsymbol{\theta}}$ to a quaternion using equations 27–29. The error covariance is updated as shown below.

$$P^+ = \left(I_{6\times 6} - K_k H_k\right) P_k^- \left(I_{6\times 6} - K_k H_k\right)^T + K_k R_k K_k^T \tag{59}$$

E. Unscented Kalman Filter

The unscented Kalman filter is another type of Kalman filter that is designed to deal with nonlinear estimation problems. It is typically a method of choice when there are complex dynamics and measurement models that make it difficult to compute the Jacobians for the EKF. Thus it should be well-suited to the problem examined here. The driving mechanism of UKF is the unscented transform (UT) of the system. The UT uses the mean and the covariance matrix of the state to determine the location for the sigma points. These points are then weighted and used to propagate and update the state estimate utilizing the full nonlinear dynamics and measurement models. However, the standard UT can drive the covariance matrix to be non-positive semidefinite and so most often the scaled unscented transform (SUT) is used to maintain the positive semidefinite nature of the covariance matrix. The SUT replaces the original set of sigma points with a transformed set scaled by a parameter α while maintaining the same mean and covariance of the estimate. The following equations demonstrate how to set up an UKF for attitude determination using the SUT. The development of these equations is taken mostly from [9].

Starting with an estimate for the state and the covariance, the first step is augmenting the covariance to include the measurement noise and process noise if it is being used. Here a superscript a will be used to stand for the augmented terms.

$$P_0^{(a)} = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix} \tag{60}$$

As with the EKF, a very small amount amount of process noise is used to "open" up the filter to prevent measurement rejection. The mean is also augmented to include the means of the process and measurement noise. This forms the initial sigma point, χ_0.

$$\chi_0^{(a)} = \begin{bmatrix} \chi_0^{(x)} \\ \chi_0^{(Q)} \\ \chi_0^{(R)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}} \\ 0 \\ 0 \end{bmatrix} \tag{61}$$

The next step involves assigning the locations of the other sigma points and their respective weights using the scaled unscented transform. There are three tuning parameters which go into this. The first, α, is a term between 0 and 1 that is used to control the spread of the sigma points. The smaller the value of α, the smaller the effect of nonlocal terms on the estimate. The second term, β, is a nonnegative weighting term that is used to incorporate knowledge of higher order terms like kurtosis. $\beta = 2$ is the choice for a Gaussian prior. A choice of the third term, $\kappa \geq 0$, is used to ensure that the covariance matrix is positive semidefinite. These three parameters are used in conjunction with the square root of the covariance matrix to determine the location of the sigma points as shown below.

$$\chi_0 = \hat{x}, \qquad i = 0 \tag{62}$$

$$\chi_i = \hat{x} + \sqrt{(L+\lambda)P_x}, \qquad i = 1, ..., L \tag{63}$$

$$\chi_i = \hat{x} - \sqrt{(L+\lambda)P_x}, \qquad i = L+1, ..., 2L \tag{64}$$

Here L is length of the augmented state, and λ is given by

$$\lambda = \alpha^2(L+\kappa) - L. \tag{65}$$

The sigma point with index $i = 0$ contains the original mean value. The square root of the covariance matrix is done using the Cholesky decomposition. The weights assigned to the different sigma points are given:

$$w_0^{(m)} = \frac{\lambda}{L+\lambda} \qquad i = 0 \tag{66}$$

$$w_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \qquad i = 0 \tag{67}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \qquad i = 1, ..., 2L \tag{68}$$

Here the superscripts of m refer to the weights associated with the specific sigma points for calculating the mean, and the c for the weights of the specific sigma points when calculating the covariance matrix.

Now in the context of the attitude with the quaternion representation. The attitude sigma points cannot be assigned in the additive manner above. To overcome this, the error GRPs are used to define the sigma points. The approach for dealing with the sigma points in this manner comes from [17]. With the error GRPs the 0th sigma point that corresponds to the mean is always 0 as it should align with the mean and thus, have no error. The other sigma points are then just taken directly from the covariance. This is done by taking the first sigma point and adding or subtracting from it the columns of the scaled square root covariance matrix. So when initializing the sigma points, they are

$$\chi_{\theta}^{(i)} = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \boldsymbol{\omega}0 \\ 0 \end{bmatrix}, \quad i = 0...2L \tag{69}$$

1. Propagation and time update equations

The general form of propagating the sigma points through the differential equation is

$$\chi_{k|k-1}^{x} = f\left(\chi_{k-1}^{x}, \chi_{k-1}^{v}\right) \tag{70}$$

where only the state components of the sigma points change, but the process noise components of the respective sigma points are still applied. To propagate the different sigma points through the nonlinear dynamics of equation 37 requires that the state vectors be in terms of the quaternion. To do this, the GRPs of the sigma points are converted to error quaternions using equations 27–29. Then the errors are applied to the current state estimate.

$$\chi_{q}^{(i)} = \delta q^{(i)} \otimes \hat{q} \tag{71}$$

With the sigma points expressed in terms of the quaternion, the state sigma points can be propagated through the dynamics of equation 37.

After propagation the process noise is applied in a similar manner to equation eq: disperturb. Note that for this controlled case, the value of the process noise is very small. After propagation, the sigma points need to be changed back to the error GRPs for calculating the updated mean estimate and covariance matrix. This is done by first computing the error quaternion using the following equation.

$$X_{\delta q,K}^{(i)} = \delta q_{k}^{(i)} \otimes (q_{k}^{(0)})^{-1} \tag{72}$$

Note that the 0th sigma point error quaternion is the unit quaternion. Then using equation 26 the quaternion components of the sigma points are changed to error GRPs. Again, the 0th sigma point should have 0 error when expressed as a GRP. The following equations are then applied to compute the new mean and the new covariance matrix.

$$\hat{x}_{k}^{-} = \sum_{i=0}^{2L} w_{i}^{(m)} \chi_{i,k|k-1}^{x} \tag{73}$$

$$P_{x_k}^{-} = \sum_{i=0}^{2L} w_{i}^{(c)} \left(\chi_{k|k-1}^{x} - \hat{x}_{k}^{-}\right)\left(\chi_{k|k-1}^{x} - \hat{x}_{k}^{-}\right)^{T} \tag{74}$$

The mean value calculated in equation 73 is then converted back to the error quaternion using equations 27–29 and used to update the previous attitude state estimate (after propagation). This update is done in a similar manner to equation 57.

2. Measurement Update equations

At the times when a measurement is available, a small amount of process noise of strength Q_{w_d} is applied to the quaternion sigma points to account for the deviations of the truth model from the commanded trajectory. This is the same approach as was used for the BPF. However, instead of the noise being drawn from the normal distribution, it is found from the scaled square root of the covariance of the process noise strength. With that extra noise applied the sigma points can then be used to update the state estimate based on the available measurements.

To begin, the expected measurements for each of the sigma points are computed. Note that in this case, the quaternions just after propagation, are used to calculate the expected measurements. Measurement noise is applied to the computed measurements by using corresponding noise components of the sigma points.

$$Y_{k|k-1} = h\left(\chi_{k-1}^{x}, \chi_{k-1}^{R}\right) \tag{75}$$

Then, using the computed distribution of measurements from the sigma points, a mean measurement is calculated.

$$\hat{y}_{k}^{-} = \sum_{i=0}^{2L} w_{i}^{(m)} Y_{i,k|k-1} \tag{76}$$

Using this mean measurement value and the mean of the sigma points from 73, the measurement covariance and the state/measurement cross-covariance can be calculated.

$$P_{\tilde{y}_k} = \sum_{i=0}^{2L} w_{i}^{(c)} \left(Y_{k|k-1} - \hat{y}_{k}^{-}\right)\left(Y_{k|k-1} - \hat{y}_{k}^{-}\right)^{T} \tag{77}$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} w_{i}^{(c)} \left(\chi_{k|k-1}^{x} - \hat{x}_{k}^{-}\right)\left(Y_{k|k-1} - \hat{y}_{k}^{-}\right)^{T} \tag{78}$$

These two matrices are then used to calculate the Kalman gain.

$$K_k = P_{x_k y_k} P_{\tilde{y}_k}^{-1} \tag{79}$$

With the Kalman gain the state update can be computed using the measurement and the mean calculated measurement.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - \hat{y}_k^-) \tag{80}$$

The state updates can then be applied like what was done with the EKF and the covariance of the state variables is updated using the following equation.

$$P_{x_k}^+ = P_{x_k}^- - K_k P_{\tilde{y}_k} K_k^T \tag{81}$$

V. Results

The purpose of this section is to explain the different test cases, present the results obtained, and provide some discussion as to the significance of these results.

There are some parameters common to all of the different test cases considered. To be able to determine if the filters were actually working well, the shape model was chosen so that each of the different facets had a distinct value for the apparent magnitude when at conditions for perfectly specular reflectance to the ground observer. This corresponds to the facet normal vector aligning with the bisector $\bar{\boldsymbol{h}}$ depicted in figure 1. The shape and reflection parameters of the satellite are presented in table 1.

Table 1 Shape and reflection model parameters of the spacecraft

Face	A(m^2)	ξ	a	m
$+Z$	0.01	0.1	0.7	0.4
$+Y$	0.01	0.1	0.5	0.5
$+X$	0.01	0.1	0.95	0.04
$-X$	0.01	0.5	0.95	0.08
$-Y$	0.01	0.5	0.80	0.5
$-Z$	0.01	1	0.95	0.3

The satellite being observed is in a medium-Earth orbit (MEO) with initial Keplerian orbital elements of $a = 20,000$ km, $e = 1.38 \times 10^{-8}$, $i = 0$ deg, $\Omega = 0$ deg, $\omega = 0$ deg, and $M = 0$ deg. The observation periods for all simulations share a common epoch of December 17, 2009, at 4:47:15 universal time. The simulated observatory is located at 30 (deg) south latitude, 111 (deg) west and an elevation of 3.059 km. All the results are based off of observations simulated for 600 seconds with measurements taken every 5 seconds. The 1σ measurement noise value for the apparent magnitude measurements is set at 0.3 apparent magnitude.

In all of the test cases the initial attitude of the satellite was set so that the +x (spacecraft body frame) facet normal was aligned with the sun/observer bisector. The initial inertial to body quaternion is $q_{I\to b}(t = 0) = \begin{bmatrix} 0.2077 & 0.3209 & -0.3641 & -0.8493 \end{bmatrix}^T$. In each case the filter models are initialized by adding a random perturbation to both the attitude and angular velocity of the satellite. This perturbation is determined by the initial levels of uncertainty in these parameters. The initial conditions of the filter estimates were all initialized to the same values so that the results could be easily compared. The initial 1σ uncertainty in the angular velocity was set to $0.2\ \frac{deg}{s}$ for all of the different test cases. In each case the particle filter used 10,000 particles and the different filters were each tuned in order to have good performance.

As the attitude of the satellite is being controlled, the simulated motion of the spacecraft was limited to two modes of operation. The first mode being the spacecraft maintaining an inertially fixed attitude, and the second mode is the spacecraft spinning at a constant rate. For each of these different modes of operation the different test cases are set up so that there is only a difference in the initial conditions of the knowledge of the attitude. However, only a single test case is presented for the inertially fixed mode of operation. This first test case is when the satellite is maintaining and inertial hold with five degrees of initial uncertainty in the attitude estimate. The other four test cases are for a spinning spacecraft. For these tests the truth model angular velocity is $\omega = \begin{bmatrix} 2.5 & 0 & 3.6 \end{bmatrix} \frac{deg}{s}$. The initial uncertainty levels of the spinning test cases are 5, 10, 30 , and 60 degrees 1σ. Please note that as the measurements from a single one of the spinning satellite test cases is representative of the measurements for the other rotating spacecraft test cases, only a single measurement curve for the spinning cases is depicted in figure 3.

As the goal is to compare the abilities of the different filters to estimate the attitude of the satellite, the presented results contain plots designed to show filter performance. These plots comprise the error between the true spacecraft state, and the 3σ bounds taken from the covariance matrix of the different filters. The error is calculated by first finding the error quaternion using the equation shown below, and then transforming this error quaternion into an error rotation vector using equations 30–32.

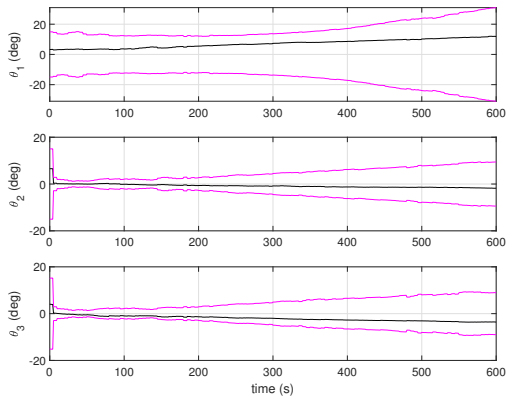
$$\delta q_\epsilon = q_{I\to b} \otimes \hat{q}_{I\to\hat{b}} \tag{82}$$

The 3σ values for each component of the state are found by taking the square root of the respective diagonal elements of the covariance matrix and multiplying these values by 3. Please note that for brevity, the results only

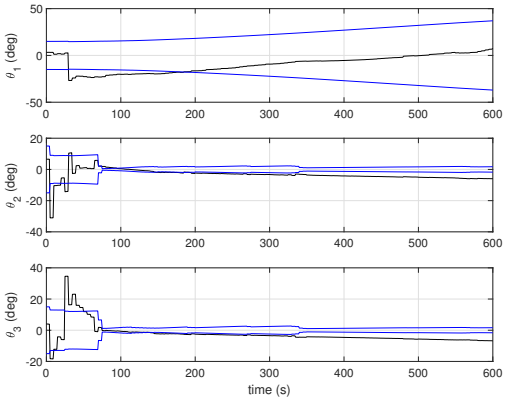
show the estimates for the attitude. As the attitude is a function of the angular velocity, an accurate attitude estimate implies an accurate angular velocity estimate, and so only the attitude results are presented.

A. Test case 1: satellite maintaining inertial attitude

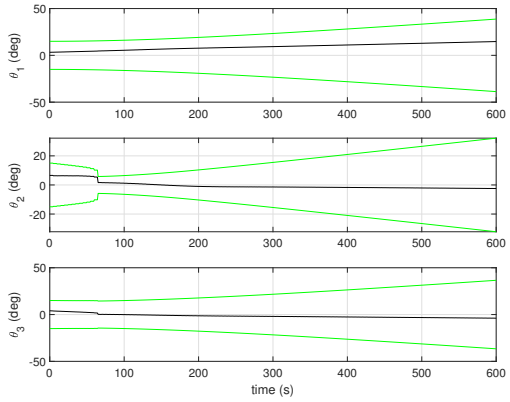
The first test case involves the satellite maintaining an inertial orientation. In this case the angular velocity components are all set to 0 while maintaining the same initial uncertainty in the angular velocity of 0.2 $\frac{deg}{s}$. This is the same value used for all of the other test cases. The initial uncertainty in the attitude was set to five degrees. The results for this test case are presented in figure 2.



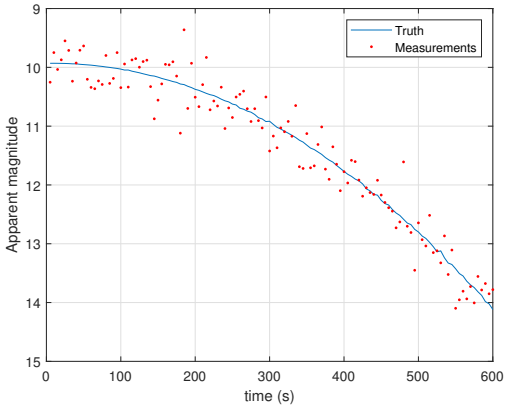
(a) BPF attitude estimate error and covariance bounds



(b) MEKF attitude estimate error and covariance bounds



(c) UKF attitude estimate error and covariance bounds



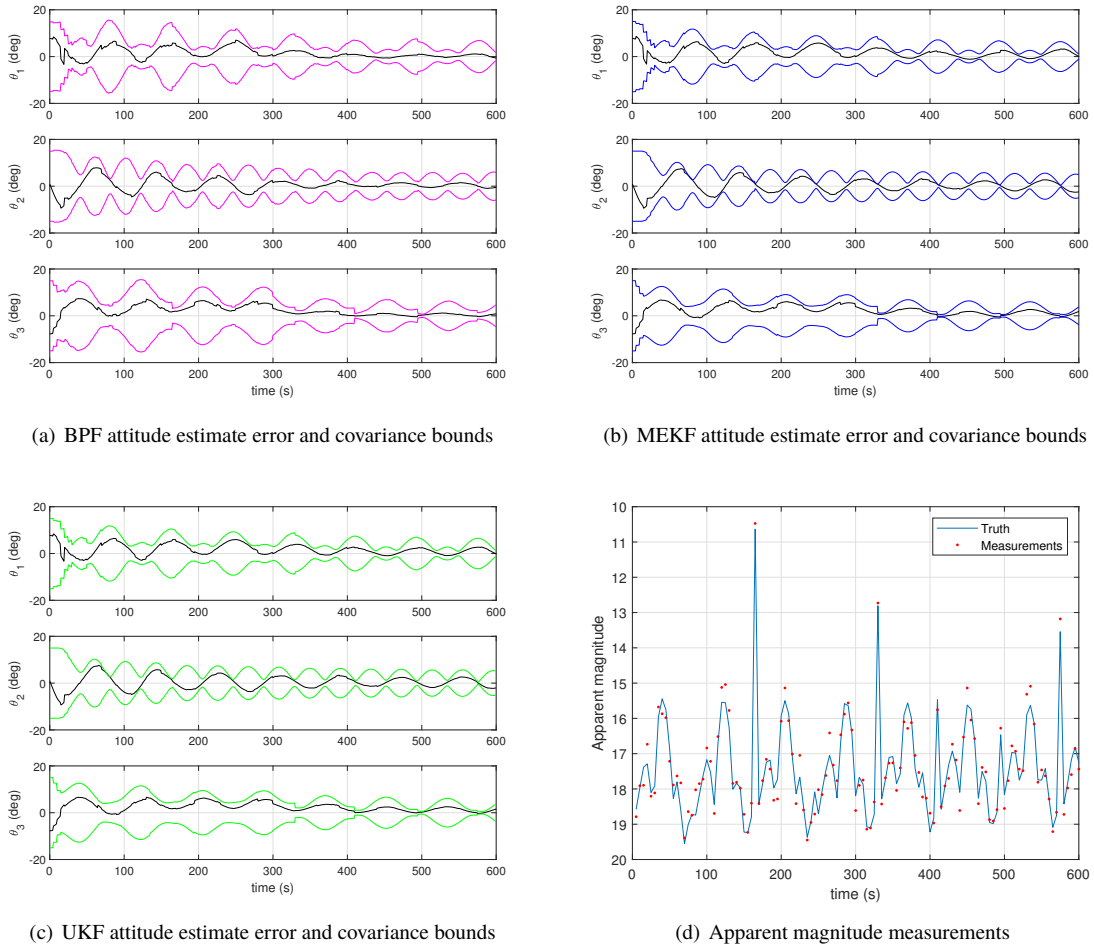
(d) Apparent magnitude measurements

Fig. 2 Results for test case 1, satelite maintaining an inertially fixed attitude with 5 degrees of initial uncertainty

From these results it is clear that none of the filters was able to accurately estimate the spacecraft attitude. With the BPF and UKF, the error only increased over time. This means that the uncertainty in the state estimate increased using the filter. With the EKF, the filter diverged, as the error left the covariance bounds. From looking at the measurements in the figure, there are no peaks or valleys in the measurement curve. The measurements just tend to trend slowly downward. It is likely that there was not enough information in the measurement for the filters to determine whether this decrease in magnitude was due to some rotation, or due to the actual case of the apparent magnitude changing strictly because of the satellite orbital motion. It is possible that the inclusion of a second observatory could improve these results. As the filters all failed for this low initial uncertainty case, only the one test case with the inertially fixed attitude is presented.

B. Test case 2: Spinning satellite with 5 deg initial attitude uncertainty

The second test case involves the truth model satellite spinning with an angular velocity of $\omega = \begin{bmatrix} 2.5 & 0 & 3.6 \end{bmatrix} \frac{deg}{s}$. The initial 1σ uncertainty in the attitude estimate is 5 degrees. The simulation results for this case are presented in figure 3



(a) BPF attitude estimate error and covariance bounds

(b) MEKF attitude estimate error and covariance bounds

(c) UKF attitude estimate error and covariance bounds

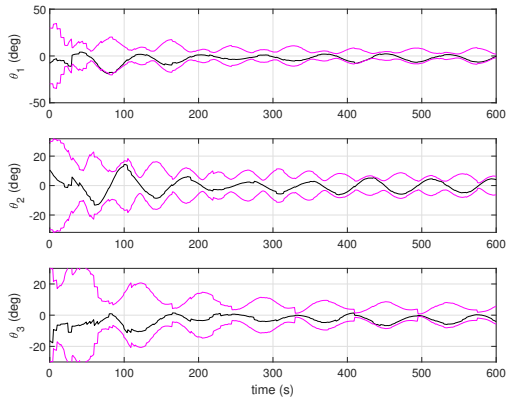
(d) Apparent magnitude measurements

Fig. 3 Results for test case 2, the spinning satellite with 5 degrees of initial attitude uncertainty

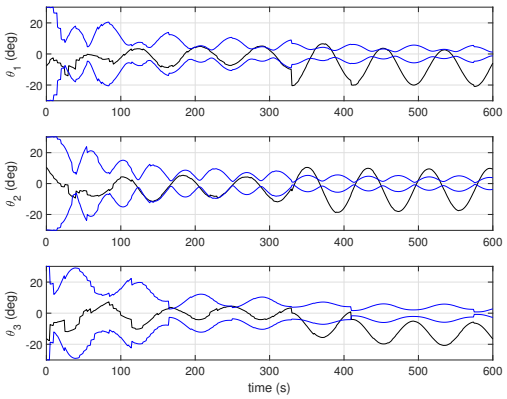
In the figure, the error between the truth state and the filter state for each of the three angle components is depicted in black with the 3σ error bounds depicted in the bright colors. As can be seen from the figure, the overall trend is that the covariance bounds are decreasing. This shows that the filters are converging upon the state estimate. All three filters are able to get the error down to the same level thus they would each be equally suitable for attitude determination under these conditions. .

C. Test case 3: Spinning satellite with 10 deg initial attitude uncertainty

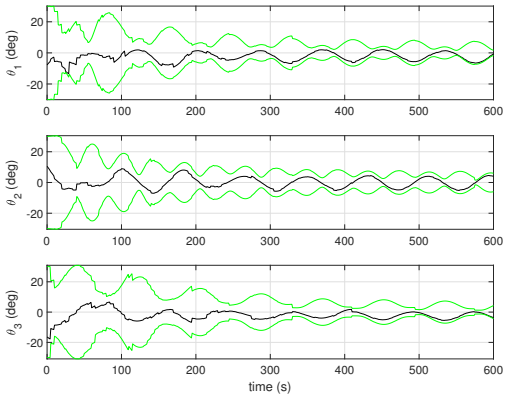
In figure 4 the results are presented for the case where the spacecraft is spinning, but the initial attitude uncertainty has been increased to 10 degrees.



(a) BPF attitude estimate error and covariance bounds



(b) MEKF attitude estimate error and covariance bounds



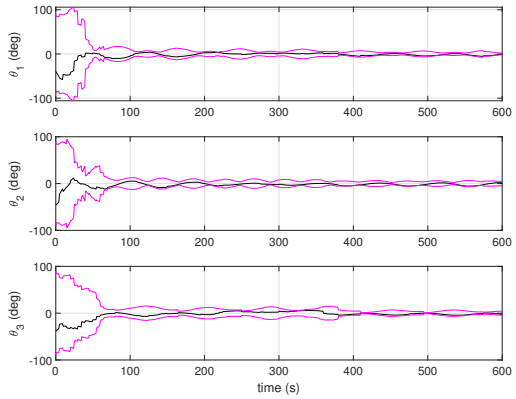
(c) UKF attitude estimate error and covariance bounds

Fig. 4 Results for test case 3, the spinning satellite with 10 degrees of initial attitude uncertainty

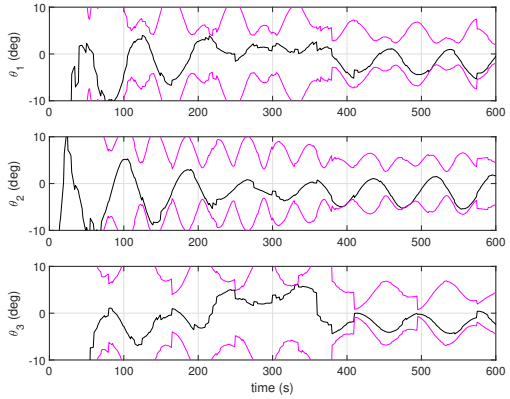
In this run, the solution of the extended Kalman filter has diverged. This can be seen from how the error is no longer contained within the 3σ bounds. However, despite the EKF failing, the unscented Kalman filter is still able to accurately estimate the attitude of the spacecraft. This implies that the initial uncertainty is large enough that the underlying assumptions of the extended Kalman filter begin to break down. The BPF is also to accurately estimate the attitude of the spacecraft.

D. Test case 4: Spinning satellite with 30 deg initial attitude uncertainty

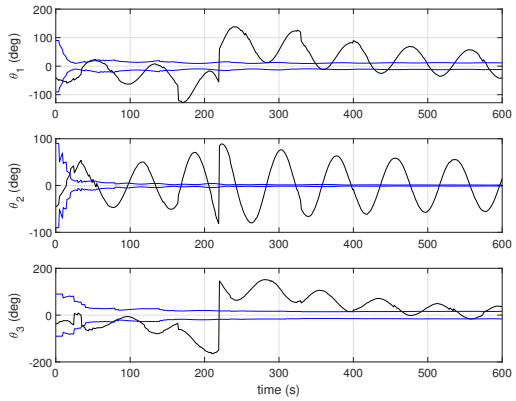
For this test case, the initial attitude uncertainty of the spinning spacecraft is increased to 30 degrees. The filter results are presented in figure 5.



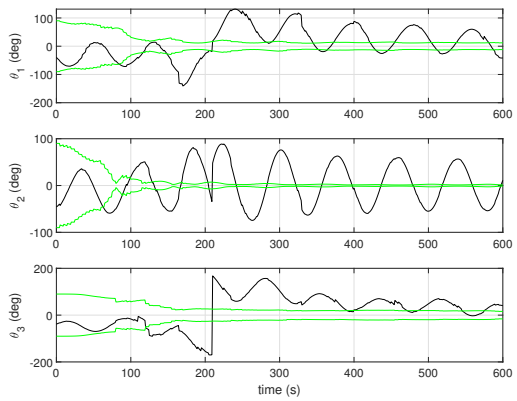
(a) BPF attitude estimate error and covariance bounds



(b) A zoomed in look at the results of the BPF in estimating the attitude of the spacecraft



(c) EKF attitude estimate error and covariance bounds



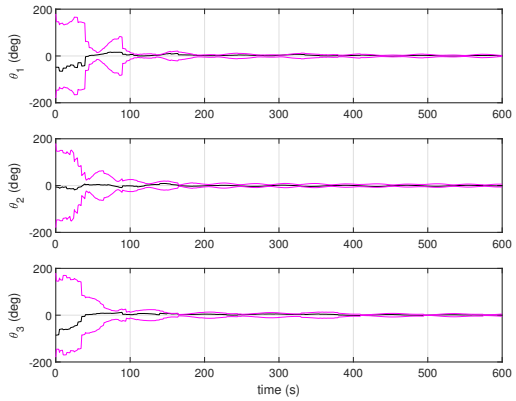
(d) UKF attitude estimate error and covariance bounds

Fig. 5 Results for test case 4, the spinning satellite with 30 degrees of initial attitude uncertainty

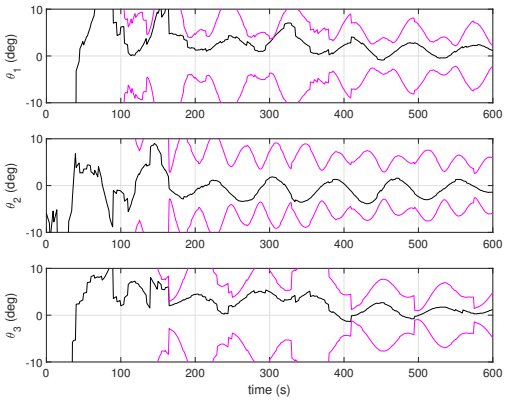
It can clearly be seen that both the UKF and the EKF have diverged at this level of initial uncertainty in the spacecraft attitude, and they are unable to accurately estimate the state. However, the top parts of the figure show that the BPF is still able to obtain an accurate estimate. The final 3σ covariance bounds are less than 10 degrees, and the error is between the covariance bounds. This means that at this level of uncertainty in the attitude state, the best choice for attitude estimation would be the bootstrap particle filter.

E. Test case 5: Spinning satellite with 60 deg initial attitude uncertainty

In this case, the initial uncertainty has been increased to 60 degrees. The 3σ value is 180 degrees which corresponds to the uncertainty limit for spacecraft attitude. As the previous results showed that the other filters diverged. The only results presented for this test case are for the BPF. The filter results are presented in figure 6



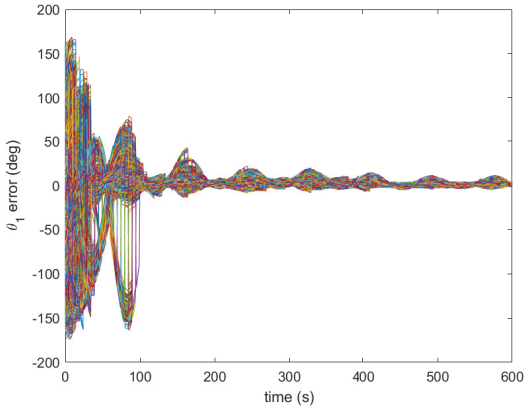
(a) BPF attitude estimate error and covariance bounds



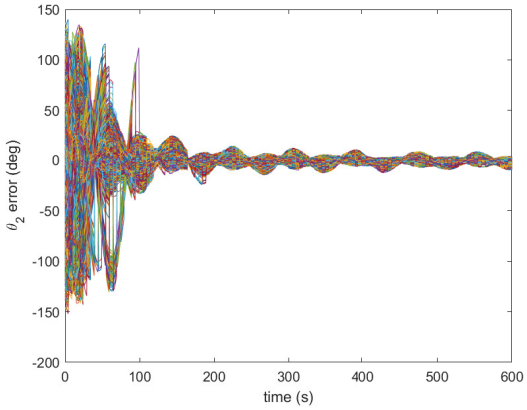
(b) A zoomed in look at the results of the BPF in estimating the attitude of the spacecraft

Fig. 6 Results for test case 4, the spinning satellite with 60 degrees of initial uncertainty

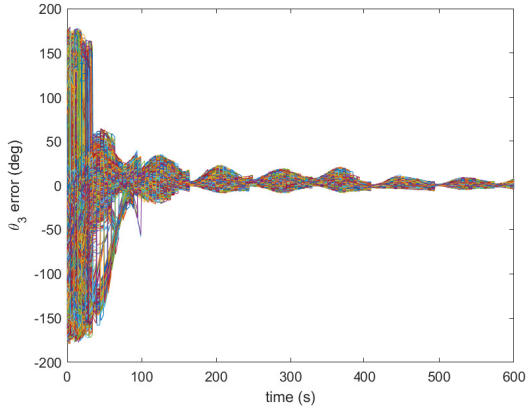
In the figure, the plot on the left shows the high level view of the filter's performance, and the plot on the right depicts a zoomed in look at these results. The figure illustrates that even under some of the worst conditions, the bootstrap particle filter is still able to determine the attitude of the satellite. Figure 7 provides some insight into why this is the case despite the failing of the other two filters.



(a) All of the particles' angle errors vs time for θ_1



(b) All of the particles' angle errors vs time for θ_2



(c) All of the particles' angle errors vs time for θ_3

Fig. 7 The error angles components over time for all of the particles used in the BPF

The time history of the error rotation vector components for all of the individual particles is plotted. From examining the figures, it becomes apparent that in the beginning of the estimation process, there were instances in which the particles were clustered about multiple locations. It is likely that these multiple clusters formed due to the distribution being multi-modal at these times. Over time all of the particles came together to converged to the true solution. Further work needs to be done to verify that the system is indeed multi-model. However, it is clear that for this scenario, the BPF is the best option for determining the attitude of the satellite.

VI. Conclusion

The choice of which filter to use for lightcurve inversion is heavily dependent upon the knowledge of the spacecraft's state and the actual rotation of the spacecraft. For the setup of this paper, none of the filters were well suited for estimating the spacecraft attitude while it maintained an inertially fixed attitude. However, in increasing the rotation rate, and exposing more of the shape's facets to the observer, the filters were able to accurately estimate the spacecraft attitude.

The EKF, while much faster computationally than the other two approaches, is only able to determine the attitude of the satellite when there is already a lot of information about the spacecraft state. When the initial attitude uncertainty was five degrees (for the rotating spacecraft), the EKF performed as well as the BPF or the UKF. However when that error doubled, the EKF diverged and filter performance decreased.

The UKF was able to perform better than the EKF in some instances. However, as the initial uncertainty increased, the UKF was also unable to estimate the attitude of the spacecraft. However, there may be many instances in which the UKF could serve as a starting point for other estimation techniques.

In all of the test cases involving a rotating spacecraft, the BPF was able to estimate the satellite attitude with

a high level of accuracy. However, the BPF took much longer to run than either the EKF or the UKF. Furthermore, in realistic situations, it may be difficult to properly tune the BPF. However, if properly tuned, the BPF would be the best suited for determining the attitude of a spacecraft when there is little to no *a priori* information about the attitude of the spacecraft.

Please note that these results are only for very specific cases. The facet parameters were configured to yield different apparent magnitude measurements when placed under ideal conditions for specular reflection. In a realistic situation, it is likely that the different components of the spacecraft may reflect light in a similar manner. Furthermore, the rotation rates chosen for the spacecraft are rather large. The spacecraft is spinning very quickly. In practice, the spacecraft might spin at a much slower rate and it might take many more observations to be able to determine the spacecraft attitude to the accuracy shown in the figures. However, this is intended to be a starting point to use as a reference when selecting the filters for use in practical applications or analysis.

Another point to consider is that although the BPF was shown to be the best choice when little to no information about the spacecraft attitude is known, it is possible that a bank of filters, either UKFs or EKFs could be used with different sets of initial conditions to determine the attitude of the spacecraft. Further work will need to be done to determine whether the banks of filters would yield more accurate results in a more timely manner than using the BPF.

Future work involves further testing to determine whether the BPF is the best filter for every occasion. Additionally, work could be done to determine whether the addition of a second sensor could improve the results obtained in this work. Other work could include determining whether there are the "best" values for tuning the filters such that they will work for every satellite attitude determination using lightcurve scenario. Furthermore, one of the key assumptions of this work is that the shape and physical reflection properties of the satellite are perfectly known. This is likely not the case for real-world applications. Thus, future work includes adding these parameters to the estimation process. Finally, future work also includes testing these different estimation techniques on real-world satellite data.

VII. Acknowledgments

The authors would like to acknowledge and thank the Utah NASA Space Grant Consortium for funding this work.

References

[1] Holzinger, M. J., Alfriend, K. T., Wetterer, C. J., Luu, K. K., Sabol, C., and Hamada, K., "Photometric attitude estimation for agile space objects with shape uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 921–932.

[2] Piergentili, F., Santoni, F., and Seitzer, P., "Attitude Determination of Orbiting Objects from Lightcurve Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No. 1, 2017, pp. 81–90.

[3] Du, X., Wang, Y., Hu, H., Gou, R., and Liu, H., "The attitude inversion method of geostationary satellites based on unscented particle filter," *Advances in Space Research*, 2018, pp. –. doi:https://doi.org/10.1016/j.asr.2018.01.018, URL `https://www.sciencedirect.com/science/article/pii/S0273117718300449`.

[4] Kaasalainen, M., Torppa, J., and Muinonen, K., "Optimization methods for asteroid lightcurve inversion: II. The complete inverse problem," *Icarus*, Vol. 153, No. 1, 2001, pp. 37–51.

[5] Jah, M., and Madler, R. A., "Satellite characterization: angles and light curve data fusion for spacecraft state and parameter estimation," *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 49, 2007.

[6] Henderson, L. S., "Modeling, estimation, and analysis of unresolved space object tracking and identification," Ph.D. thesis, The University of Texas at Arlington, 2014.

[7] Linares, R., Crassidis, J. L., and Jah, M. K., "Particle filtering light curve based attitude estimation for non-resolved space objects," *AAS/A1AA Space Flight Mechanics Meeting*, Vol. 14, 2014, p. 210.

[8] Bernard, A., and Geller, D., "A Comparison of the Boostrap Particle Filter and the Extended Kalman Filter and the Effects of Non-linear Measurements in Spacecraft Attitude Determination Using Light Curves," *Utah NASA Spacegrant Consortium*, 2018.

[9] Van Der Merwe, R., "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. thesis, Oregon Health Science University, 2004.

[10] Markley, F. L., and Crassidis, J. L., *Fundamentals of spacecraft attitude determination and control*, Vol. 33, Springer, 2014.

[11] Cook, R. L., and Torrance, K. E., "A reflectance model for computer graphics," *ACM Transactions on Graphics (TOG)*, Vol. 1, No. 1, 1982, pp. 7–24.

[12] Beckmann, P., and Spizzichino, A., "The scattering of electromagnetic waves from rough surfaces," *Norwood, MA, Artech House, Inc., 1987, 511 p.*, 1987.

[13] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., "Space object shape characterization and tracking using light curve and angles data," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 13–25.

[14] Gordon, N. J., Salmond, D. J., and Smith, A. F., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F (Radar and Signal Processing)*, Vol. 140, IET, 1993, pp. 107–113.

[15] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman filter: Particle filters for tracking applications*, Artech house, 2003.

[16] Crassidis, J. L., and Junkins, J. L., *Optimal estimation of dynamic systems*, CRC press, 2011.

[17] Crassidis, J. L., and Markley, F. L., "Unscented filtering for spacecraft attitude estimation," *Journal of guidance, control, and dynamics*, Vol. 26, No. 4, 2003, pp. 536–542.