Use of Extended Kalman Filter in Estimation of Attitude of a Nano-Satellite

Sai Dheeraj Nadella
Manipal Institute of Technology/Electronics and Communication, Manipal, India
Email: saidheeraj@gmail.com

***Abstract*—State estimation theory is one of the best mathematical approaches to analyze the changes in the states of a system or a process. The state of the system is defined by a set of variables that provide a complete representation of the internal conditions of the system at any given time instant. There are two types of state models - linear state model and non-linear state model, which require different estimation techniques. Linear estimation of a system can be easily carried out by using Kalman Filter (KF), when the state space model is linear. But, most of the real life state models are nonlinear, thereby limiting the practical applications of the KF. The Extended Kalman Filter, Unscented Kalman filter and Particle filter are most commonly used for nonlinear estimation. EKF is the nonlinear version of the Kalman filters which revolves about the mean and covariance at the current time instant. The estimation can be linearized around the current estimate using the partial derivatives to compute estimates even in the nonlinear relations. This paper deals with estimation of various parameters of a nonlinear model with Extended Kalman filter (EKF).The paper analyses EKF method of estimating and then determining the attitude of the satellite depending upon the readings from the magnetometer.**

***Index Terms*—extended Kalman filter, estimation, attitude control, state space model, nano-satellite, magnetometers**

I. Introduction

Filtering and estimation are two very important tools of engineering. [1] Whenever the state needs to be estimated from noisy sensor information, the estimator produces the best estimate of the true system state. When the system dynamics and observation models are linear, the minimum mean squared error or the least mean square error estimate can be computed using the Kalman filter. We control the process modeling by obtaining from a priori-knowledge, certain observable parameters.

The success of the linear model in identification or in control has its cause in the good understanding of it. Since, most of the models are non-linear in reality; we generally deal with the non linear state space estimation. A common approach to overcome this problem is to linearize the system before using the Kalman filter, i.e. by using an Extended Kalman filter. This linearization does however have some problems, e.g. the error between the true value and the estimated values can vary beyond acceptance, after a long period of time. The development of better estimator algorithms for nonlinear Systems has therefore attracted a great deal of interest in the scientific community, because the improvements will undoubtedly have great impact in a wide range of engineering fields. This paper deals with how to estimate a nonlinear model with Extended Kalman filter (EKF). The approach in this paper is to analyze Extended Kalman filter where EKF provides better probability of state estimation for a satellite determination in the space, based upon the value of readings of Magnetometer and Sun Sensor.

II. Linear and Nonlinear Models

A. State Space Models

[2] A state space model is a mathematical model of a process, where state x of a process is represented by a numerical vector. State-space model actually consists of two sub models: The process model, which describes how the state propagates in time based on external influences, such as input and noise; and the measurement model, which describe how measurements z are taken from the process, typically simulating noisy and/or inaccurate measurements.

B. Linear State Space Model

A linear state-space model assumes the functions F and H are linear, in both state and input. The functions can then be expressed by using the matrices, B and H, reducing state propagation calculations to linear algebra. Overall this results in the following state-space model:

$$X(k) = F(k)X(k-1) + B(k)U(k-1) + w(k-1) \quad (1)$$

$$z(k) = H(k)x(k) + v(k) \quad (2)$$

where u is process input, w is state vector, v is measurement noise vector, k is the discrete time.

The above expressions (1) and (2) govern state propagation and measurements respectively. Linear model is easier both to calculate and analyze. Linear state models are either based on inherently linear processes, or simply linearized versions of a nonlinear process by means of a First order Taylor approximation.

C. Nonlinear State Space Model

The most general form of state-space models is the Non-linear model. This model does typically consist of two functions f and h.

Manuscript received February 2, 2014; revised April 11, 2014.

doi: 10.12720/ijeee.3.1.38-43

$$Xk = f(xk-1, uk-1, wk-1) \quad (3)$$

$$Zk = h\,(xk, vk) \quad (4)$$

III. EXTENDED KALMAN FILTER

A. Background-State Estimation

[3] State estimation concerns the problem of estimating the probability density function (pdf) for the state of a process which is not directly observable. This involves both predicting the next state (based on the current state) and applying updates (based on measurement model).

Estimator: Estimator is a tool that predicts the future behavior of a model from the available system information. The Estimator uses knowledge about the evaluation of the variable, the probabilistic characterization of the various random variables and the prior information. The generalized block diagram of space estimation is as shown in Fig. 1.

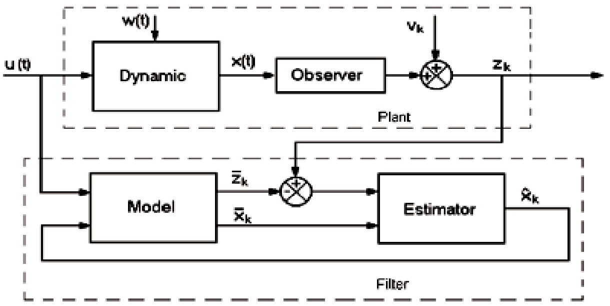

Figure 1. Mathematical view of state estimation

Different estimators

- Recursive Bayesian Estimation
- Kalman Filter (KF)
- Extended KF (EKF)
- Unscented KF (UKF) and
- Particle filter (PF)

B. Recursive Bayesian Estimation (RBE)

[4] The most general form of state estimation is known as Recursive Bayesian Estimation. This is the optimal way of predicting a state pdf for any process, given a system and a measurement model. RBE works by simulating the process, while at the same time adjusting it to account for new measurements z, taken from the real process. The calculations are performed recursively in a two-step procedure. First, the next state is predicted by extrapolating the current state onto next time step, using state propagation belief $p(x_k | x_{k-1})$ obtained from function f. Secondly, this prediction is corrected using measurement likelihood $p(z_k | x_k)$ obtained from function h, taking new measurements into account. Unfortunately, this method does not scale very well in practice, mainly vectors. Calculating the prior probability of each point in this state space involves a multidimensional integral, which quickly becomes intractable as the state space grows. Computers are also limited to calculation of the pdf in discrete point in state space, requiring a discretization of the state space. This technique is therefore mainly considered as a theoretic foundation for state estimation in general. Bayesian estimation by means of computers is only possible if either the state space can be discretized, or if certain limitations apply for the model.

C. Kalman Filter

[5] The problem of state estimation can be made tractable if we put certain constrains on the process model, by requiring both ‘f’ and ‘h’ to be linear functions, and the Gaussian and white noise terms ‘w’ and ‘v’ to be uncorrelated, with zero mean. Put in mathematical notation, we then have the following constraints (5) and (6). As the model is linear and input is Gaussian, we know that the state and output will also be Gaussian. The state and output pdf will therefore always be normally distributed, where mean and covariance are sufficient statistics. This implies that it is not necessary to calculate a full state pdf anymore, a mean vector x cap and covariance matrix P for the state will suffice. The basic Kalman filter loop is as shown in the Fig. 2.

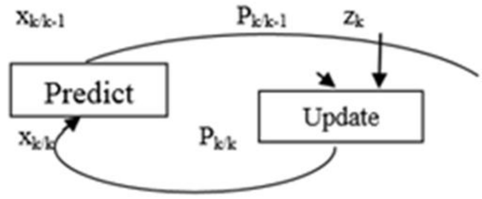

Figure 2. Kalman filter loop

The recursive Bayesian estimation technique is then reduced to the Kalman filter, where f and h is replaced by the matrices F, B and H. The Kalman filter is, just as the Bayesian estimator, decomposed into two steps: predict and update.

The Kalman filter is quite easy to calculate, due to the fact that it is mostly linear, except for a matrix inversion. It can also be proved that the Kalman filter is an optimal estimator of process state, given a quadratic error metric. Most processes in real life are not linear, and therefore need to be linearized before they can be estimated by means of a Kalman filter. So the practical applications of the KF are limited and so modified KF, aka EKF is generally used. Different from KF, EKF deals with nonlinear process model and nonlinear observation model. In the extended Kalman filter, the state transition and observation models need not be linear functions of the state, but may be differentiable functions. The nonlinear process model (from time k-1 to time k) is described as

$$Xk = f(xk-1, uk-1) + wk-1 \quad (5)$$

$$Z_k = h\,(x_k) + w_k \quad (6)$$

where *xk*-1, *xk* are the system state (vector) at time *k*-1; *k*, *f* is the system transition function, *uk* is the control, *wk* is the zero - mean Gaussian process noise $wk \sim N(0; Q)$; h is the observation function and *vk*+1 is the zero - mean Gaussian observation noise $vk+1 \sim N(0; R)$.

The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed. At each time step the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

D. Predict and Update Equations

Predicted state estimate

$$Xcapk|k-1 = f(xcap\ k-1|k-1, uk-1) \quad (7)$$

Predicted covariance matrix

$$Pk\big|k-1 = F_{k-1} + P_{k-1|k-1}\big| + F_{k-1}^{T} + \mathrm{Qk}-1 \quad (8)$$

Update Equations

Innovation or measurement residual

$$y^{\sim}{}_{\mathrm{k}} = z_{\mathrm{k}} - h(xcap_{\mathrm{k|k\text{-}1}}) \quad (9)$$

Innovation (or residual) covariance

$$Sk = Hk + Pk|k-1 + HkT + Rk \quad (10)$$

Near-optimal Kalman gain

$$Kk = Pk|k-1 + HkT + Sk-1 \quad (11)$$

Updated state estimate

$$Xcapk|k-1 = xcapk|k-1 + Kkyk\sim \quad (12)$$

Updated covariance estimate

$$Pk|k = (1 - KkHk)Pk|k-1 \quad (13)$$

where the state transition and observation matrices are defined to be the following Jacobians

$$Fk-1 = \frac{\partial f}{\partial x} \mid xcapk-1|k-1, uk-1 \quad (14)$$

$$Hk = \frac{\partial h}{\partial x} \mid xcapk|k-1 \quad (15)$$

E. Continuous-Time Extended Kalman Filter

Model

$$x(t) = f(x(t), u(t)) + w(t), w(t) \sim N(0, Q(t)) \quad (16)$$

$$z(t) = h(x(t)) + v(t), v(t) \sim N(0, R(t)) \quad (17)$$

Initialize

$$xcap(t0) = E[x(t0), P(t0) = Var[x(t0)] \quad (18)$$

Predict Update

$$xcap(t0) = f(xcap(t0), u(t) + K(t)z(t) - h\big(xcap(t0)\big) \quad (19)$$

$$(t) = F(t)P(t) + P(t)F(t)^{\mathrm{T}} - K(t)H(t)P(t) + \mathrm{Q(t)} \quad (20)$$

$$F(t) = \frac{\partial f}{\partial x} \mid xcap(t), u(t) \quad (21)$$

$$K(t) = P(t)H(t)^{\mathrm{T}}R(t)^{-1} \quad (22)$$

$$H(t) = \frac{\partial h}{\partial x} \mid xcap(t) \quad (23)$$

Unlike discrete-time extended Kalman filter, the prediction and update steps are coupled in continuous-time extended Kalman filter.

F. Continuous-Discrete Extended Kalman

[6] Most physical systems are represented as continuous-time models while discrete-time measurements are frequently taken for state estimation via a digital processor. Therefore, the system model and measurement model are given by

$$x(t) = f\big(x(t), u(t)\big) + w(t)$$

$$w(t) \sim N(0, Q(t)) \quad (24)$$

$$z(t) = h(xk) + v_{\mathrm{k}} \quad (25)$$

$$vk \sim N(0, Rk)$$

where, $x_k = x(t_k)$

Initialize

$$xcap(t0) = E[x(t0)], P(t0) = Var\,[x(t0)\,] \quad (26)$$

Predict

$$Xcap'(t) = f(xcap'(t), u(t))$$

$$P'(t) = F(t)P(t) + P(t)F\,(t)T + Q(t)$$

$$Xcapk|k-1 = xcap(tk)$$

$$Pk|k-1 = P(tk) \quad (27)$$

with

$$F(t) = \frac{\partial f}{\partial x} \mid xcap(t), u(t) \quad (28)$$

The update equations are identical to those of discrete-time extended Kalman filter.

$$Kk = Pk|k-1Hk\,T(HkPk|k-1Hk + R \quad (29)$$

$$xcapk|k = xcapk|k-1 + Kk\big(z - h(xcapk|k-1)\big) \quad (30)$$

$$Pk\big|k = \big(I - KkHk\big)Pk|k-1 \quad (31)$$

where $H_k = \partial h/\partial x\,|xcapk|k\text{-}1$

G. Modeling Example of Attitude Estimation and Control of a Nano-Satellite

Functional Flow

[7] The Nano-satellite extended Kalman filter uses the MBMV from the satellite emulator and the MIRV from the orbit propagator to estimate the spacecraft's attitude at each time step in the time vector. The EKF also uses the estimated quaternions to generate a magnetic body estimate vector (MBEV) as a check to make sure the updated quaternion estimates transform the MIRV into a vector that is similar to the MBRV. As another check, the EKF takes the SIRV from the orbit propagator and transforms it into the solar body estimated vector (SBEV) using the quaternion estimates. The SBEV is then used to find the solar panel estimated power (SPEP). The SPEP is then compared to the SPRP, and the solar panel characterization used to find the SPEP is different than how the solar panels actually generate their power, it is assumed that large differences between the SPEP and SPRP will be able to be seen and attributed to filter

divergence. In the verification mode the filter outputs plots of the error between actual and estimated quaternion and rates, as well as the error between the MBRV and the MBEV, and the SPRP and the SPEP, all of which will be shown in detail later on in this chapter.

Equation and Matrix Derivations

State Vector

The full state vector to be used for the EKF will be:

$$Xbar = \begin{bmatrix} qbar \\ \omega bar \end{bmatrix} \quad (32)$$

where

$$Qbar = \begin{bmatrix} qbar \\ q4 \end{bmatrix} \quad (33)$$

The system model equations

$$Q.bar = ½\, \Omega(\omega bar) qbar$$

and

$$\omega.1 = K1\omega2\omega3$$

$$\omega.2 = K2\omega1\omega3$$

$$\omega.3 = K3\omega1\omega2 \quad (34)$$

These equations are used to propagate the state vector from one time step to the next in the EKF. The propagation is done using a numerical differential equation solver in Matlab, this is acceptable for a ground-based routine; however, this propagation method will have to be changed for this routine to be implemented on-board a satellite.

Body-Fixed (Reduced) State Vector

A body-fixed state vector is also used in the EKF in order to remove the built-in quaternion redundancy that comes from representing a frame transformation with four variables. The removal of this redundancy eliminates the need to deal with quaternion normalization and associated issues with the covariance matrix P in the EKF. The body-fixed, or reduced, state vector is defined as:

$$x^{\sim} = \begin{bmatrix} \delta qbar \\ \omega bar \end{bmatrix} \quad (35)$$

The δ qv vector is the vector component of the error quaternion that transforms the estimated full quaternion to the actual quaternion, shown below.

$$q = \delta q \otimes q\hat{} \quad (36)$$

Assuming the filter is converging, the error quaternion will start to represent a smaller and smaller rotation; and as such, the magnitude of the vector component will be less than one and will be approaching zero. Thus, the fourth component of the error quaternion can be computed as:

$$\delta q_4 = \sqrt{1 - ||\delta qbar||}^2 \quad (37)$$

However, during initial convergence the magnitude of the vector component could be greater than one, leading to an imaginary fourth quaternion component. During this situation the error quaternion is calculated by the method developed by Humphreys of Utah State University, shown below.

$$\delta qbar = \frac{1}{\sqrt{1+||\delta qbar||}^2} \begin{bmatrix} \delta qbar \\ 1 \end{bmatrix} \quad (38)$$

A relationship between the reduced state vector and the full state vector can be derived.

First the quaternion product is expressed as matrix multiplication

$$\text{qbar} = \delta qbar \otimes q_{\text{bar}}{}^{\wedge} = [\textstyle\sum(q_{\text{bar}}{}^\wedge) | \, qbar^\wedge]\delta qbar \quad (39)$$

$$\textstyle\sum(q_{\text{bar}}{}^\wedge) = [q4 - q3 \;\; q2$$
$$q3 \;\; q4 \; -q1$$
$$-q2 \;\; q1 \;\; q4$$
$$-q1 \; -q2 \; -q3]$$

Knowing that quaternions are normalized, the following can be shown.

$$\delta qv = \Xi T\, (q\hat{}\,) q \quad (40)$$

Now the relationship can be shown mathematically as:

$$\begin{bmatrix} \delta qbar \\ \omega bar \end{bmatrix} = \begin{bmatrix} \Xi|T\ (qbar^\wedge) & 0\,3x3 \\ 03x4 & I3x3 \end{bmatrix} \begin{bmatrix} qbar \\ \omega bar \end{bmatrix} \quad (41)$$

Using properties of quaternion multiplication, the reduced state dynamic equation is found to be:

$$\delta qbar = ½\, \Omega(\omega\text{bar})\, \delta qbar - ½\, \delta qbar \otimes \omega\text{bar}^{\wedge} \quad (42)$$

State Error Vector

The state error vector $\Delta x^{\sim}$ is defined as:

$$\Delta x^{\sim} \equiv x^{\sim} - x^{\sim\wedge} \quad (43)$$

Using the fact

$$\Xi^{T}(qbar)^\wedge qbar^\wedge = 0 \quad (44)$$

The state error vector becomes

$$\Delta x^{\sim} = \begin{bmatrix} \delta qbar \\ \Delta\omega\text{bar} \end{bmatrix} \quad (45)$$

State Transition Matrix

The state transition matrix is approximated as:

$$\Phi k \approx I + FkTs \quad (46)$$

The linear system dynamics matrix Fk can be found be linearizing the dynamics equations about the current state vector estimate. The linearization results in the following matrix.

$$F_k = \begin{bmatrix} -[\omega^\wedge \times] & 1/2 I_{3\times3} \\ 0_{3\times3} & \Theta(\omega^\wedge) \end{bmatrix}$$

where

$$\Theta(\omega^\wedge) = \begin{matrix} 0 & K_1\omega_3 & K_1\omega_2 \\ K_2\omega_3 & 0 & K_2\omega_1 \\ K_3\omega_2 & K_3\omega_1 & 0 \end{matrix} \quad (47)$$

Measurement Matrix

The measurement matrix is derived from the following equation.

$$b\ barBody = \text{A}(\delta qbar) A(qbar^\wedge) MIRV \quad (48)$$

This equation can be rewritten as:

$$Bbarkbody = A(\delta qbar_k * \mathrm{bbar}_k^{body\,\wedge})$$

where

$$\mathrm{Bbar}_k^{body\,\wedge} = A(qbar_k^{\wedge})\ \mathrm{MIRV}_k \qquad (49)$$

The transformation matrix A(q) can be written as:

$$A(qbar) = (q42 - ||q||2)I3 \times 3 + 2(qbar)(qbar)T - 2q4[qbar\ \times] \qquad (50)$$

The rotation created from δq is small, and reduces the previous equation to:

$$A\ (\delta qbar) = I_{3\times 3} - 2[\delta qbar\ \ \times] \qquad (51)$$

Now, using the cross product commutative relationship, h (~x) can be written as:

$$h(x\sim) = I_{3\times 3} + 2[(b\ bar)^{\wedge}\ \times]\ \delta qbar \qquad (52)$$

The discrete measurement matrix easily follows from the above equation.

$$Hk = [2[bbark^{\wedge}\ Body \times]\ 03 \times 3] \qquad (53)$$

The Noise Matrices

The measurement noise matrix is defined as:

$$R_k = \sigma^2_{\mathrm{magnetometer}} I_{3\times 3} \qquad (54)$$

The constant in front of the identity matrix is the square of the standard deviation of the Magnetometers. The process noise matrix is found from the equation in Section 6.2.2.2 using the previously defined state transition matrix. The non-discrete process noise matrix gets factored into an identity matrix and scalar as shown below.

$$[Q] = QI6 \times 6 \qquad (55)$$

Physically the scalar Q is the square of the error of the first derivatives of the state vector. This parameter is set during the filter tuning process.

Loop Initialization Equations

The initial quaternion estimate $-\hat{\bar{q}}_0$ is just taken to be aligned with the ECI frame. The initial rate estimate $-\hat{\bar{\omega}}_0$ is found be using the first two magnetic field measurements, which will be shown below. The change in the magnetic field measured by the satellite is given by the following equation.

$$bbar^{\cdot Body}_{Measured} = b\ bar^{\cdot Body}_{IGRF} + \omega bar \times bbar^{\cdot Body}_{Measured} \qquad (56)$$

For short sampling times the inertial change of the magnetic field is negligible and the measured change can be approximated by the difference between two consecutive measurements divided by the sampling time.

$$(bbar_2^{Body} - bbar1Body)/T = \omega\mathrm{bar} \times bbar_1^{\mathrm{Body}} \qquad (57)$$

A pseudo-inverse cross product is then used to yield the initial rate estimate, as shown below.

$$\omega_0 bar^{\wedge} = \frac{bbar_1^{\mathrm{Body}} \times \frac{bbar_2^{\mathrm{Body}} - bbar_1^{\mathrm{Body}}}{T_S}}{\left\|bbar_1^{\mathrm{Body}}\right\|^2} \qquad (58)$$

The initial covariance estimate Pbar_0 is defined as the 6x6 identity matrix multiplied by constant.

$$Pbar0 = PI6 \times 6 \qquad (59)$$

The scalar parameter *P* is set during the filter tuning process along with *Q*.

Filter Tuning and Performance

[8] In the actual implementation of the filter, the measurement noise covariance is usually measured prior to operation of the filter. Measuring the measurement error covariance is generally practical (possible) because we need to be able to measure the process anyway (while operating the filter) so we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement noise. The determination of the process noise covariance is generally more difficult as we typically do not have the ability to directly observe the process we are estimating. Sometimes a relatively simple (poor) process model can produce acceptable results if one “injects” enough uncertainty into the process via the selection. Certainly in this case one would hope that the process measurements are reliable. In either case, whether or not we have a rational basis for choosing the parameters, often times superior filter performance (statistically speaking) can be obtained by tuning the filter parameters Q and R. The tuning is usually performed off-line, frequently with the help of another (distinct)Kalman filter in a process generally referred to as system identification. Fig. 3 clearly illustrates the tuning of the Kalman filter.

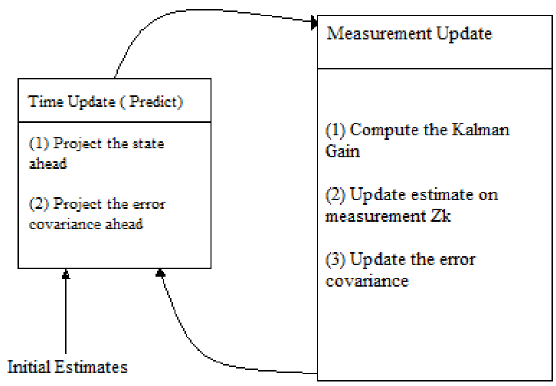

Figure 3. Filter tuning

In closing we note that under conditions where states are in fact constant, both the estimation error covariance and the Kalman gain will stabilize quickly and then remain constant.

If this is the case, these parameters can be pre-computed by either running the filter off-line, or for example by determining the steady-state value.

[9] It is frequently the case however that the measurement error (in particular) does not remain constant. For example, when sighting beacons in our optoelectronic tracker ceiling panels, the noise in measurements of nearby beacons will be smaller than that in far-away beacons. Also, the process noise is sometimes changed dynamically during filter operation—becoming

—in order to adjust to different dynamics. For example, in the case of tracking the head of a user of a 3D virtual environment we might reduce the magnitude of if the user seems to be moving slowly, and increase the magnitude if the dynamics start changing rapidly. In such cases the choice would be to account for both uncertainties about the user's intentions and uncertainty in the model.

REFERENCES

[1] Simon Julier and Jeffrey Uhlmann. "A new extension of the kalman filter to nonlinear systems," in Proc. *Int. Symposium. Aerospace/Defence Sensing, Simulation and Controls*, Orlando, FL, 1997.

[2] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. "A novel approach to nonlinear/non-gaussian bayesian state stimulation," *IEEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107-113, 1993.

[3] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 2nd Ed. Wiley-Interscience, Jan. 2001.

[4] Cox, H., "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Transactions on Automatic Controls*, vol. AC-9, pp. 5-12, 1964.

[5] F. E. Daum, "New exact nonlinear filters," *Bayesian Analysis of Time Series and Dynamic Models*, vol. 94, pp. 199-226, 1988.

[6] M. Athans, R. P. Wishner, and A. Bertolini. "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 504-514, 1968.

[7] M. L. Psiaki, F. Martel, and P. K. Pal, "Three-axis attitude determination via kalman filtering of magnetometer data," *Journal of Guidance, Control, and Dynamics*, vol. 13, pp. 506-514, 1990.

[8] E. J. Lefferts, *et al*., "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance and Control*, vol. 5, pp. 417-429, Sep. 1982.

[9] Extended Kalman Filter, *ARC Centre of Excellence for Autonomous Systems (CAS)*, Faculty of Engineering and Information Technology, University of Technology Sydney, 2010.

Sai Dheeraj Nadella was born in Andhra Pradesh, India, on 22nd May, 1992. He is currently doing his Bachelor of Engineering in Electronics and Communication branch at Manipal Institute of Technology, Manipal, Karnataka, India.
He has been a part of Parikshit Student Satellite team, as a part of Attitude Determination and Control Subsystem from the past 2 years. He has helped design the estimation algorithms for the satellite. Further, he has also worked on the Global Positioning System, and the actuators of the Nano-satellite. His areas of interest include Signal processing, Communication theory and adaptive controls.