TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology

Institute of Informatics

IDK40LT

Annaliisa Romanenkov 120942IAPB

A STUDY ON THE EFFICIENCY OF THE KALMAN FILTER FOR ATTITUDE DETERMINATION AND CONTROL SYSTEM OF A SATELLITE

Bachelor's thesis

Supervisor: Martin Rebane

MSc

Lecturer

Tallinn 2016

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatikainstituut

IDK40LT

Annaliisa Romanenkov 120942IAPB

KALMANI FILTRI EFEKTIIVSUSE UURIMINE SATELLIIDI ASENDI MÄÄRAMISE JA JUHTIMISE JUURES

Bakalaureusetöö

Juhendaja: Martin Rebane

MSc

Lektor

Tallinn 2016

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Annaliisa Romanenkov

23.05.2016

Abstract

The aim of this thesis is to analyse the Kalman filter and its implementation for the most effective attitude determination and control system of a satellite.

The fulfilment of set aims and determining the suitability of the results is achieved by comparing the results of various versions of the algorithm and by varying fixed constants.

The result of this thesis is a more effective attitude determination and control system of the satellite, which will be achieved mostly by using the MATLAB software and a UKF packet that has been created for it.

This thesis is written in English and is 31 pages long, including 5 chapters and 10 figures.

Annotatsioon

Kalmani filtri efektiivsuse uurimine satelliidi asendi määramise ja juhtimise juures

Töö eesmärgiks on uurida Kalmani filtrit ning selle rakendamist satelliidi võimalikult efektiivseks asendi määramiseks ja juhtimiseks.

Eesmärkide täitmine ja tulemuste sobivuse kontrollimine on saavutatav läbi erinevate algoritmi versioonide tulemuste võrdlemise ning fikseeritud konstantide varieerimise.

Töö tulemuseks on efektiivsem satelliidi asendi määramine ja juhtimine, mis saavutatakse põhiliselt kasutades MATLAB tarkara ning selle jaoks juba loodud UKF paketti.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 31 leheküljel, 5 peatükki ja 10 joonist.

List of Abbreviations and Terms

ADCS – Attitude Determination and Control System

$E_{estimate}$ – Error in estimate

E_{EST_t} – Error in the current estimate

$E_{EST_{t-1}}$ – Error in the previous estimate

$E_{measurement}$ – Error in measurement

EKF – Extended Kalman Filter

EST_t – Current estimate

EST_{t-1} – Previous estimate

I – Identity matrix

KG – Kalman gain

KF – Kalman Filter

MEA – Measurement

P – Process covariance matrix

Q – Process noise covariance matrix

R – Sensor noise covariance matrix

u – Control variable matrix

UKF – Unscented Kalman Filter

UT – Unscented transformation

w – Predicted state noise matrix

X – State matrix

Y – Measurement of the state

Z – Noise of the measurement

Table of Contents

List of Figures

1 Introduction

In March 2014 Mektory Space Centre was established. According to the official website [1], one of the most important initiatives that the Space Centre has been brought into life for has been the launch of Tallinn University of Technology Nanosatellite Programme. The Nanosatellite programme is an international, interdisciplinary, university wide programme, carried out by students and professors from different nations, in association with national and international partners from various industries and academia. The aim of the Mektory Nanosatellite programme is to ensure the development of new technologies and provide high quality work force both to Estonia and international high-tech companies.

The goals of the satellite programme [1] are to design, build and launch a reliable and powerful Nanosatellite to the orbit, to create and operate the satellite through an earth station, to provide the students with theoretical and practical experience to prepare them for working as highly rated technology and space industry experts and to encourage the creation of new high value added occupations and new technology companies to the market.

The building and testing of this satellite is planned to the first half of 2016 and the launch of the satellite will likely take place in the first half of 2018.

1.1 Problem of Interest

The main problem, which we will examine in this paper, is whether the unscented Kalman filter helps us to determine the attitude and to control the satellite most effectively.

This problem is important because we need to choose the best algorithm for controlling the attitude of the satellite and to specify the inputs and outputs of the module. We also need to evaluate whether the unscented Kalman filter is the most suited option for this work or should we rather choose some other algorithm. Another reason, why it is important to study this problem, is that fully understanding the UKF can be quite difficult and it is extremely important for the programmers in the satellite project to completely understand every aspect of Kalman filtering to be able to implement it in code.

1.2 Overview of the Paper

In the first chapter we take a look at the satellite programme in general, explain the problem we will be studying and its importance and also give the overview of the paper.

The second chapter first explains what the ADCS is and then goes more in depth about both the attitude determination and attitude control, describing the processes more closely.

In the third chapter we start learning about the Kalman filter, first looking at how the filter works and then explaining the simple system model and the multi-dimensional system model with examples.

The fourth chapter provides us with an implementation of the Kalman filter in MATLAB. We take a look at a UKF toolbox, explain the different parts of it and then test the algorithm with example values.

In the last chapter we sum up the whole paper, look at the results and conclusions of this study.

2 Attitude Determination and Control System

The aim of Attitude Determination and Control System (ADCS) [2] is to make sure, what is the position of the satellite on the orbit, how fast is the satellite gyrating and to change the attitude and speed of the satellite if need be.

2.1. Attitude Determination

We know that the satellite is equipped with sun sensors, magnetometers and gyroscopes. By combining the results of all these sensors it is possible to determine the position of the satellite in relation to the Earth and its magnet field and to determine the speed at which the satellite is gyrating at.

The sun sensors [3] consist of outer housing that has two thin slits in it and behind the slits are the sensors with all the needed electronics. When sunlight goes through a slit it will create a stripe of light on the sensors and the attitude of the light can be determined by the electronics. With the attitude on sensors we can calculate the angle of the current sensor in relation to the Sun. The satellite has a total of six sun sensors (a sensor on every side) and each of them can determine the direction of the Sun on its own. As a result, there is always at least one sensor that can see the sun and therefore we can always determine the attitude of the satellite.

Magnetometer [4] is a device to determine the direction of the magnet field. It allows us to identify the direction of the magnet field at the current position of the satellite and to direct the satellite with the comparison of the on-board model of the Earth's magnet field.

On-board gyroscopes [5] measure the speed at which the satellite is rotating at around all three axles.

Given the last measured position and the inputs from the on-board sensors, it is possible to calculate how much the attitude of the satellite differs from the desired position and how the strength of the satellite's electromagnets should be adjusted to achieve a desired attitude and speed of gyrating. The info that we get from the sensors of the satellite allows

us to determine the direction of the magnet field and the attitude of the satellite at any given moment in time.

The satellite attitude estimation can generally be viewed as a process consisting of two stages [6]. The first stage is to estimate the attitude of the satellite using models of the given system and the second stage is to improve the estimate using sensor measurements and reference observations. Attitude estimation can be performed in a lot of different ways, which can generally be separated into two main groups - deterministic point by point solutions and filters. Deterministic point by point solutions estimate the attitude from at least two or more reference vectors observed at a single point in time. The point estimation algorithms are fast and relatively simple, but the requirement of at least two reference vectors constrains the use of these algorithms on satellites with an inadequate number of sensors. The recursive filters can estimate the attitude more accurately as they also estimate the attitude by combining dynamic and/or kinematic models with measurements obtained by several sensors. Because the recursive filters include dynamic and/or kinematic models they also have the ability to propagate the attitude during eclipse and do not necessarily require two or more reference observations like the deterministic methods.

To get a better overview where exactly does the attitude determination of a satellite take place, we will look at a drawing [7] showing all the components of ADCS in normal working mode.

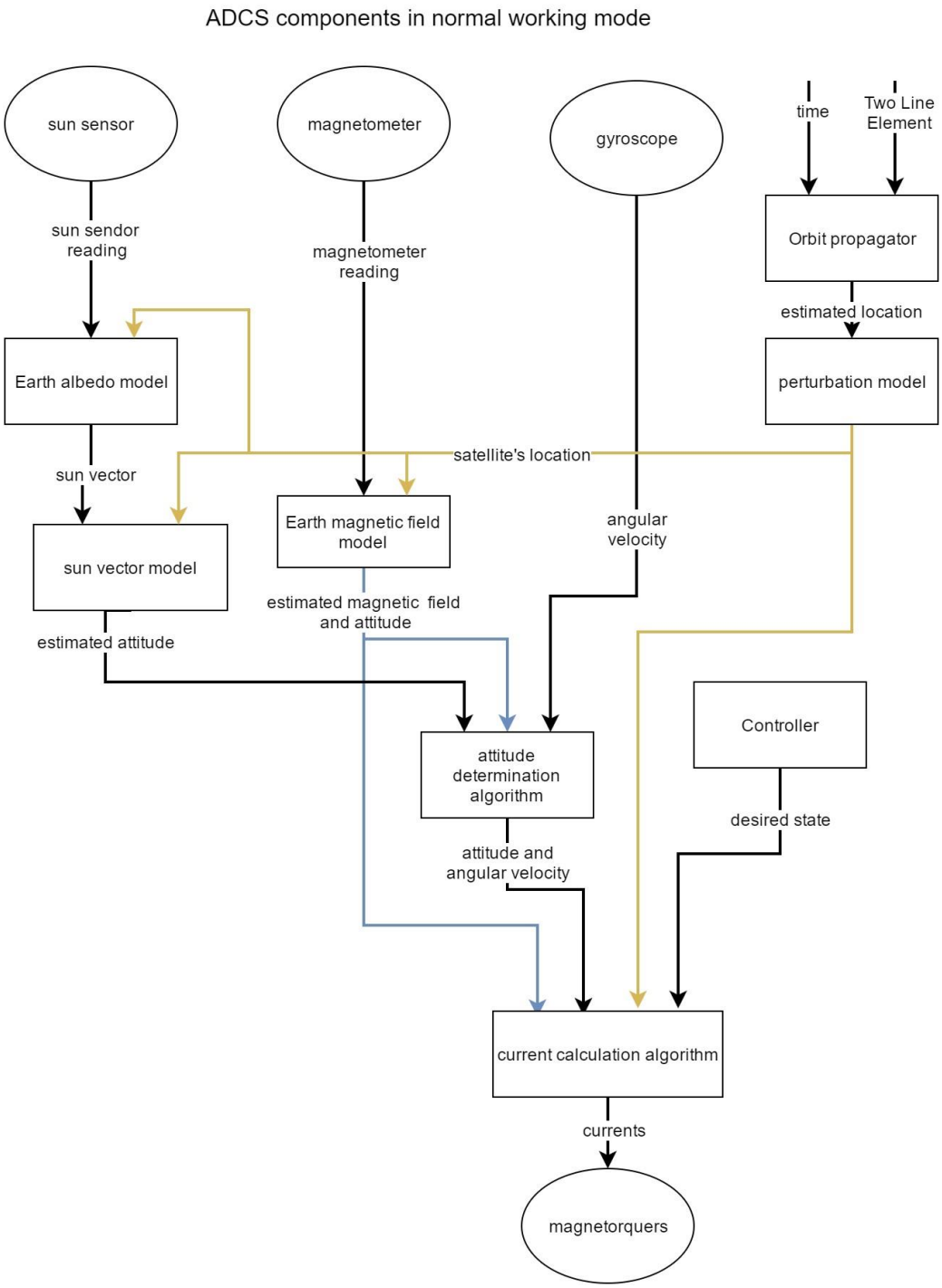

Figure 1. ADCS components in normal working mode [7].

In this drawing, the attitude determination algorithm box is the part where the unscented Kalman filtering, that we will later look at, will take place. As we can see, the inputs for this algorithm come from sun sensors, magnetometers and gyroscopes and the output goes into current calculation algorithm.

2.2 Attitude Control

Attitude control [8] is controlling the orientation of an object with respect to an inertial frame of reference or another entity. Controlling a satellite's attitude requires sensors for measuring the satellite's orientation, actuators for applying the torques needed to re-orient the satellite to a desired attitude, and algorithms for commanding the actuators based on sensor measurements of the current attitude and specification of a desired attitude.

To stabilize a satellite [9], it must have a system that keeps it moving evenly through its orbit. Satellites often use a spinning or gyroscopic motion to keep them stable. If it is not stabilized, a satellite's measurements and pictures will be inaccurate and its orbit is more likely to slowly change course either toward the Earth or out into space. In stabilizing a satellite, the direction that the satellite's instruments and solar panels are facing is also important. It is easier and cheaper to power a satellite that has solar panels that are constantly exposed to the sunlight. This is necessary for satellites with unusually high energy requirements. However, this is not possible if the satellite is spinning. There are several ways to stabilize a satellite, for example spin stabilizing, spinning/despinning and three-axis stabilizing.

Magnetic torqueing [6] is also a very common attitude control method for nanosatellites. It works simply by running a current through a coil of wire. This creates a magnetic field which interacts with Earth's magnetic field and generates a torque T that brings the fields into alignment. The equation for the torque is

$$T = NIS\ x\ B \tag{1}$$

where N is the number of turns of wire, I is the current, S is the area of the wire loop and B is the local magnetic field vector. By reversing the current, thee direction of the torque is also reversed. Magnetic rods work the same way, but they have a ferromagnetic core to amplify the magnetic field created. The torque is typically small at reasonable currents and rod sizes. Proper control also requires knowledge about the local magnetic field direction and strength. In equatorial orbit, 3-axis stabilization with only magnetic torque is not achievable, because the magnetic field lines point constantly to one direction. On the other hand, in polar orbits the magnetic field line directions are not always predictable near the poles which decreases the accuracy of

this method. Nevertheless, magnetic torques are widely used due to the simplicity and for being an external control method to achieve momentum dumping.

3 Kalman Filter

3.1 History and Introduction

One of the options for implementing ADCS for a nanosatellite is the modified Kalman filter. The Kalman filter got its name after engineer Rudolph E. Kalman published an article about state estimation in 1960 even though a similar algorithm was developed earlier than that. It has been said [10] that some were slow to accept Kalman's work in the beginning, but the first implementation of the filter, developed by Stanley F. Schmidt after Kalman's visit to the NASA Ames Research Centre made him see the applicability of Kalman's ideas to the nonlinear problem of the trajectory estimation for the Apollo program, helped to gain acceptance.

By definition [11], the Kalman filter is a linear algorithm that predicts the next state of a system and its efficiency is expressed by real time, optimal and fast calculation of a new state based on the previous one. For that reason, this filter is ideal for a continuously changing system. Kalman filter calculates the new state prediction based on the Kalman gain, measured value and the previous prediction. The implementation of the filter [12] is made up of two main steps: first the new state and its uncertainty is predicted and then it is corrected by the next measurement.

The Kalman filter is optimal in cases where the model perfectly matches the real system, the entering noise is white and Gaussian and the covariance of the noise are exactly known. After the covariance are estimated, it is useful to evaluate the performance of the filter, i.e. whether it is possible to improve the state estimation quality. If the Kalman filter works optimally, the output prediction error is a white noise, therefore the whiteness property of the innovations measures filter performance. Several different methods can be used for this purpose.

The basic Kalman filter [13] is meant for linear systems, but the changing of a satellite's states is nonlinear so it is necessary to implement a special version of the Kalman filter. In that case, the options are either to use the extended Kalman Filter (EKF) [14] or the unscented Kalman Filter (UKF) [15]. The main qualities of both of these filters play the key role in choosing the more suitable option. EKF is very efficient and it is not linear. It can also diverge if nonlinearities are large. Still, EKF works surprisingly well even when

all assumptions are violated. On the other hand, the UKF also happens to be a very efficient (same complexity as the EKF) filter with an even better linearization than the EKF. UKF is also free of derivatives, but it is still not very optimal. In conclusion, the UKF describes a nonlinear system better than the EKF and it is also faster when getting the results.

3.2 Simple System Model

Based on the previous definition, we can make up a figure that shows how the filter works.

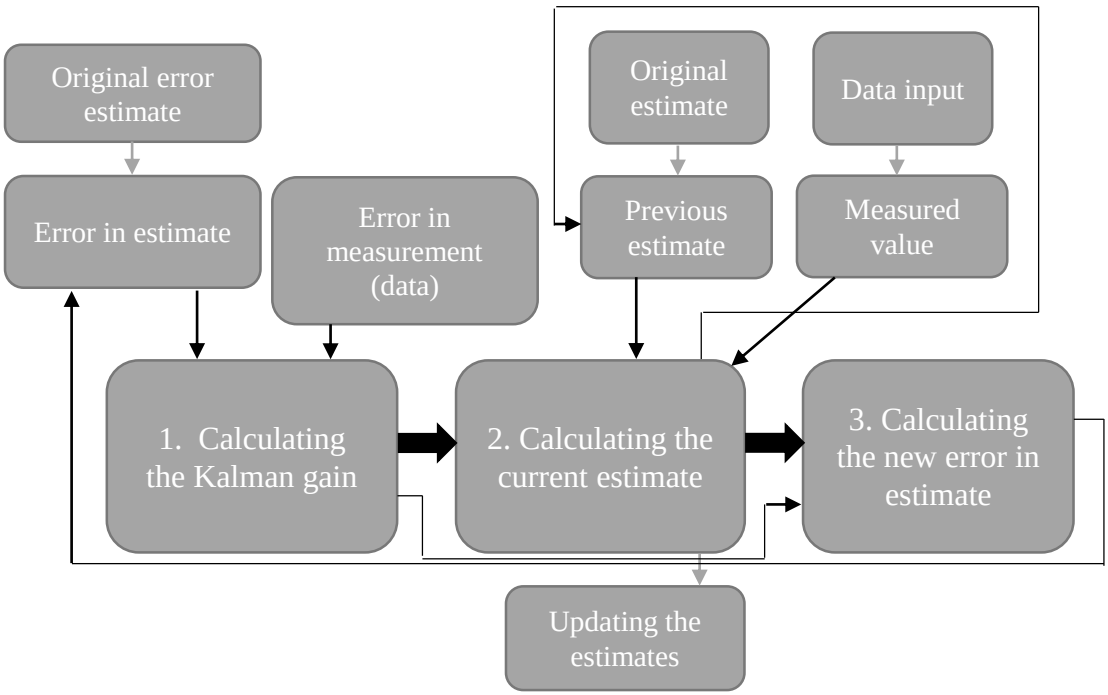

Figure 2. A simple model for the Kalman filter.

From the model above, we can derive the steps of Kalman filtering. The three main processes are calculating the Kalman gain, calculating the current estimate and calculating the new error in estimate.

First, we need to calculate the Kalman gain. For that we need to know the values of error in estimate and error in measurement. We get the Kalman gain by dividing the error in estimate by the sum of both the error in estimate and error in measurement. The value of

Kalman gain is between 0 and 1, so the bigger the error in measurement is, the smaller Kalman gain will be and *vice versa*. Therefore, the formula for calculating the Kalman gain is as follows.

$$KG = \frac{E_{estimate}}{E_{estimate}+E_{measurement}} \quad (2)$$

Secondly, we have to recalculate the current estimate. As we can see, we now need the Kalman gain we just looked at, the previous estimate and the measured value. The get the value of the current estimate, we need to subtract the previous estimate from the measurement, multiply the subtraction by the Kalman gain and then add the previous estimate to it. When we put all of this together we get the following formula:

$$EST_t = EST_{t-1} + KG(MEA - EST_{t-1}) \quad (3)$$

From this, we also get that if the value of the Kalman gain is either 1 or close to it, then the measurements are more accurate, but the estimates are unstable and if the value of Kalman gain is either 0 or close to it, then the measurements are inaccurate, but the estimates are stable (meaning the error is smaller). Typically, over time the Kalman gain gets smaller and we get closer and closer to the true value.

Lastly, we need to recalculate the new error in the current estimate. In order to do that, we have to know the values of the error in measurement and error in the previous estimate. Then we can multiply the error in measurement by the error in the previous estimate and divide that by the sum of these two as shown below:

$$E_{EST_t} = \frac{E_{measurement}*E_{EST_{t-1}}}{E_{measurement}+E_{EST_{t-1}}} \quad (4)$$

This formula can also be shown in a slightly different format using the Kalman gain:

$$E_{EST_t} = (1 - KG)E_{EST_{t-1}} \quad (5)$$

3.2.1 An example

For our first example, we will use the Kalman filter to estimate temperature. We already know that the first estimated value of temperature is 15°C and that the initial error of this estimate is 2°C. We also have the first measurement of the temperature, which is 19°C and the error in this measurement is 3°C.

We start off by calculating the Kalman gain using the aforementioned formula. Therefore

$$KG = \frac{2°C}{2°C+3°C} = \frac{2}{5} = 0{,}4 \quad (6)$$

Now that we know that the first value of Kalman gain is 0.4, we can calculate both the new estimate and the new error in the estimate:

$$EST_t = 15°C + 0{,}4 * (19°C - 15°C) = 15 + 1{,}6 = 16{,}6°C \quad (7)$$

$$E_{EST_t} = (1 - 0{,}4) * 2°C = 0{,}6 * 2 = 1{,}2°C \quad (8)$$

With the values of the current estimate and the error in the current estimate, we can go through all three formulas again with these new inputs. We now have a new measurement of temperature and it is 18°C.

$$KG = \frac{1{,}2°C}{1{,}2°C+3°C} = \frac{1{,}2}{4{,}2} = 0{,}29 \quad (9)$$

$$EST_t = 16{,}6°C + 0{,}29 * (18°C - 16{,}6°C) = 16{,}6 + 0{,}41 = 17{,}01°C \quad (10)$$

$$E_{EST_t} = (1 - 0{,}29) * 1{,}2°C = 0{,}71 * 1{,}2 = 0{,}85°C \quad (11)$$

Out next measurement is 20°C.

$$KG = \frac{0{,}85°C}{0{,}85°C+3°C} = \frac{0{,}85}{3{,}85} = 0{,}22 \quad (12)$$

$$EST_t = 17{,}01°C + 0{,}22 * (20°C - 17{,}01°C) = 17{,}01 + 0{,}66 = 17{,}67°C \quad (13)$$

$$E_{EST_t} = (1 - 0{,}22) * 0{,}85°C = 0{,}78 * 0{,}85 = 0{,}66°C \quad (14)$$

The new measurement we have is 21°C.

$$KG = \frac{0{,}22°C}{0{,}22°C+3°C} = \frac{0{,}22}{3{,}22} = 0{,}01 \quad (15)$$

$$EST_t = 17{,}67°C + 0{,}01 * (21°C - 17{,}67°C) = 17{,}67 + 0{,}03 = 17{,}77°C \quad (16)$$

$$E_{EST_t} = (1 - 0{,}01) * 0{,}66°C = 0{,}99 * 0{,}66 = 0{,}65°C \quad (17)$$

Now if we analyse these calculations, we see that with each iteration both the Kalman gain and the error of the estimate got smaller. As the Kalman gain gets smaller, we get closer to the true value of the temperature. Each time, the change in the estimated value got smaller and therefore, based on these calculations we can say that the real value of the temperature is around 18°C.

3.3 Multi-Dimensional System Model

Previously, we looked at a simple system model where the Kalman filter could be useful, but as a satellite is a multi-dimensional system and the inputs are matrices then we need to use an appropriate model. First, let us take a look at a model [16] explaining what the Kalman filter process for a multi-dimensional system looks like.

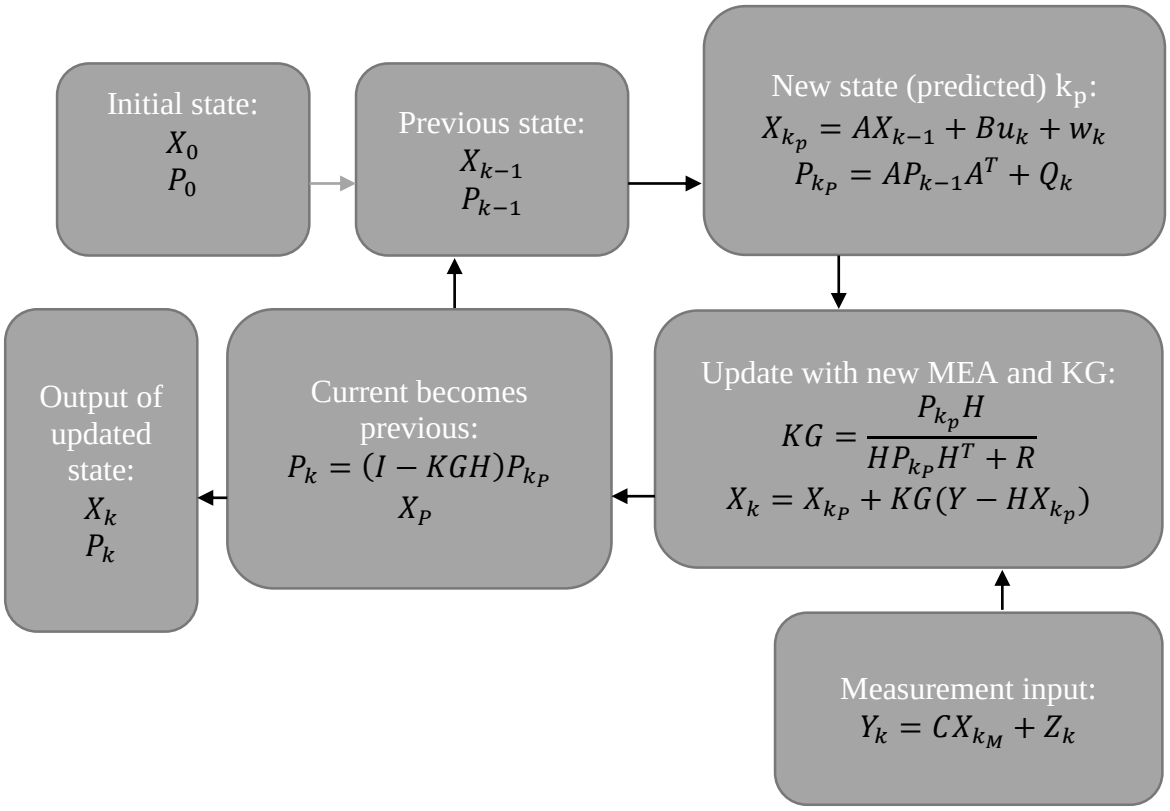

Figure 3. The multi-dimensional model of the Kalman filter.

First, we have the initial state that consists of the initial state matrix and the process covariance matrix, which represents error in the process (estimate). Then we move on to the previous state and here we also have both the state matrix and the process covariance matrix.

At this point, we can predict the new state X. In order to do that, we need to know the previous state so we can update the new state with a prediction. For that, we use the control variable matrix u that helps us control the object we are interested in while the gravity also has effect on its controls and the predicted state noise matrix w. The A and B matrices are simply adaptation matrices that are used to convert the input to the new

state matrix. We also need to update the process covariance matrix P. Here we use the matrix A and also the transpose of the matrix A to put the input into a correct format and add process noise covariance matrix to the multiplication. After we have theoretically predicted the new state, we can add an actual measured input to the prediction.

In the measurement input part of the model we calculate the measurement of the current state. In order to do that we multiply the state matrix of the measurement by a matrix C that is another adaption matrix that helps us get the measurement into a correct format and if there is some noise on uncertainty in our system then we also need to add a measurement noise to the multiplication.

After we have the updated measurement we can update the state. For that, we need to calculate two different things. First, we have to come up with the Kalman gain and it will decide how much of both the measurement and the prediction we will use to update the new state. Therefore the second calculation, updating the current state, will include both the theoretical prediction and the actual measurement. In calculating the Kalman gain we will divide the process covariance matrix by the sum of itself and the sensor noise covariance matrix. Secondly, we can calculate the new state by subtracting the predicted state from the measurement, multiplying it by the Kalman gain and then also adding the previous state to it.

As the next step, we need to update the process covariance matrix. There we also output the new state and the new process covariance matrix (the new predicted error in the process). That output shows us the position where we think the object we are monitoring is at. Now we can give the output to the previous state and therefore all the current values will become previous state values and we can go through the cycle of calculations again with the new values.

3.3.1 An example

As we are learning the efficiency of the Kalman filter for tracking and controlling a satellite, we will look at an example where we are dealing with a three dimensional

system. For our example, we will look at how to calculate the current state matrix X_k. As explained above, to calculate the state matrix we have the following formula:

$$X_k = AX_{k-1} + Bu_k + w_k \tag{18}$$

In a 3-dimensional system the matrix $X_k = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$ and the control variable $u_k = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$, which represents the acceleration in all three directions. As we remember, the matrices A and B are adaption matrices that help us get the values into the correct format. Therefore we now have the following equation:

$$X_k = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \\ \Delta T & 0 & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}\Delta T \\ y_{k-1} + \dot{y}\Delta T \\ z_{k-1} + \dot{z}\Delta T \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} +$$

$$\begin{bmatrix} a_x \frac{1}{2}\Delta T^2 \\ a_y \frac{1}{2}\Delta T^2 \\ a_z \frac{1}{2}\Delta T^2 \\ a_x \Delta T \\ a_y \Delta T \\ a_z \Delta T \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta T + a_x \frac{1}{2}\Delta T^2 \\ y_{k-1} + \dot{y}_{k-1}\Delta T + a_y \frac{1}{2}\Delta T^2 \\ z_{k-1} + \dot{z}_{k-1}\Delta T + a_z \frac{1}{2}\Delta T^2 \\ \dot{x} + a_x \Delta T \\ \dot{y} + a_y \Delta T \\ \dot{z} + a_z \Delta T \end{bmatrix} \tag{19}$$

As a result we get the new current state of the object we are tracking.

3.4 Extended Kalman Filter

The extended Kalman filter (EKF) [14] is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance. In the extended Kalman filter, the state transition and observation models can be differentiable functions instead of linear functions of the state.

$$x_k = f(x_{k-1}, u_k) + w_k \tag{20}$$

$$z_k = h(x_k) + v_k \tag{21}$$

In these equations w_k and v_k are the noises (process and observation), which are both assumed to be zero mean multivariate Gaussian noises with covariance Q_k and R_k respectively and u_k is the control vector.

The function f can be used to calculate the predicted state from the previous estimate. Similarly, the function h can be used to compute the predicted measurement from the predicted state. However, a matrix of partial Jacobian derivatives is computed because f and h cannot be applied to the covariance directly. The Jacobian is evaluated with current predicted states at each step in time and these matrixes can be used in the Kalman filter equations. Thanks to this process, we can essentially linearize the non-linear function around the current estimate.

3.5 Unscented Kalman Filter

The unscented Kalman filter (UKF) [10] uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points, which are called sigma points, around the mean. These sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered. The result is a filter which more accurately captures the true mean and covariance. In addition, this technique removes the requirement to explicitly calculate Jacobians, which for complex functions can be a difficult task in itself. Similar to the EKF, the UKF prediction can also be used in combination with a linear update, independently from the UKF update or *vice versa*. The UKF also addresses the approximation issues that we encounter with the EKF.

The way the unscented Kalman filter works is that first we initialize the starting points, then we calculate Sigma points, after that we can update the time and measurements. For the UKF algorithm we have the following nonlinear system described by the difference equation and the observation model with additive noise:

$$x_k = f(x_{k-1}) + w_{k-1} \tag{22}$$

$$z_k = h(x_k) + v_k \tag{23}$$

where x_k is the current state of our system, described by the function of the previous state and the previous state noise matrix.

In the implementation and testing part of this paper, we will focus on the unscented Kalman filter instead of the extended Kalman filter, as for a nonlinear system like a satellite, using the UKF is more optimal.

4 Implementation in MATLAB

Implementing the Kalman Filter in MATLAB is the main way of testing the filter in order to give a better understanding of the algorithm and to ensure that the filter works as intended. MATLAB (the short form of Matrix Laboratory) is a multi-paradigm numerical computing environment and programming language [17]. Among other options, MATLAB also allows implementation of algorithms, which enables us to implement the Kalman filter in MATLAB.

In our study we will use the EKF/UKF Toolbox for MATLAB V1.3 [18] and an example code for unscented Kalman filter [19]. We will look at all three functions separately with explanations.

First, we will study the main function, `ukf(f,x,P,h,z,Q,R)`, which returns state estimate x and state covariance P for nonlinear dynamic system.

Inputs for this function are: f - function handle for $f(x)$, x - *a priori* state estimate, P - *a priori* estimated state covariance, h - function handle for $h(x)$, z - current measurement, Q - process noise covariance and R - measurement noise covariance and outputs for this function are: x - *a posteriori* state estimate and P - *a posteriori* state covariance.

```
function [x,P]=ukf(f,x,P,h,z,Q,R)
L=numel(x);
m=numel(z);
alpha=1e-3;
ki=0;
beta=2;
lambda=alpha^2*(L+ki)-L;
c=L+lambda;
Wm=[lambda/c 0.5/c+zeros(1,2*L)];
Wc=Wm;
Wc(1)=Wc(1)+(1-alpha^2+beta);
c=sqrt(c);
X=sigmas(x,P,c);
[x1,X1,P1,X2]=ut(f,X,Wm,Wc,L,Q);
[z1,Z1,P2,Z2]=ut(h,X1,Wm,Wc,m,R);
P12=X2*diag(Wc)*Z2';
K=P12*inv(P2);
x=x1+K*(z-z1);
P=P1-K*P12';
```

Figure 4. Function for the UKF in MATLAB.

This function calculates the number of states L based on the state estimate given as an input and similarly the number of measurements m based on the given current measurement. We also get the weights for means W_m and weights for covariance W_c. After that we need to calculate the Sigma points and both UT of process and measurements. In the end, we update the values of state and covariance.

Secondly, we will look at the function for unscented transformation. Here we will give the function the values of nonlinear map f, Sigma points X, weights for mean W_m, weights for covariance W_c, number of outputs of nonlinear map n, additive covariance R nad as a result we will get transformed mean y, transformed sampling points Y, transformed covariance P and transformed deviation Y_1 for output.

```
function [y,Y,P,Y1]=ut(f,X,Wm,Wc,n,R)
L=size(X,2);
y=zeros(n,1);
Y=zeros(n,L);
for k=1:L
    Y(:,k)=f(X(:,k));
    y=y+Wm(k)*Y(:,k);
end
Y1=Y-y(:,ones(1,L));
P=Y1*diag(Wc)*Y1'+R;
```

Figure 5. Function for the UT in MATLAB.

And lastly, we have the function that is used to calculate the Sigma points around reference point. For this function, the inputs are reference point x, covariance P, coefficient c and the output is X, which will give us the Sigma points.

```
function X=sigmas(x,P,c)
A = c*chol(P)';
Y = x(:,ones(1,numel(x)));
X = [x Y+A Y-A];
```

Figure 6. Function for the Sigmas in MATLAB.

4.1 Validation of UKF results

As a test we run the UKF function with the following example inputs:

```
n=3;
q=0.1;
r=0.1;
Q=q^2*eye(n);
R=r^2;
f=@(x)[x(2);x(3);0.05*x(1)*(x(2)+x(3))];
h=@(x)x(1);
s=[0;0;1];
x=s+q*randn(3,1);
P = eye(n);
N=20;
```

Figure 7. Test inputs for the UKF toolbox in MATLAB.

Here we give the algorithm the number of states n, std of process q, std of measurement r, covariance of process Q, covariance of measurement R, the nonlinear state equation f, the measurement equation h, initial state s, initial state with noise x, initial state covariance P and the total of dynamic steps N.

When we run this algorithm, we get the result [0.1094; 0.0552; 0.0019].

As we run the function for the UKF we also get the charts showing how the actual state (blue line) and estimated state (red line) vary throughout the dynamic steps.

Below are shown the charts of three different tests, each showing three subplots.

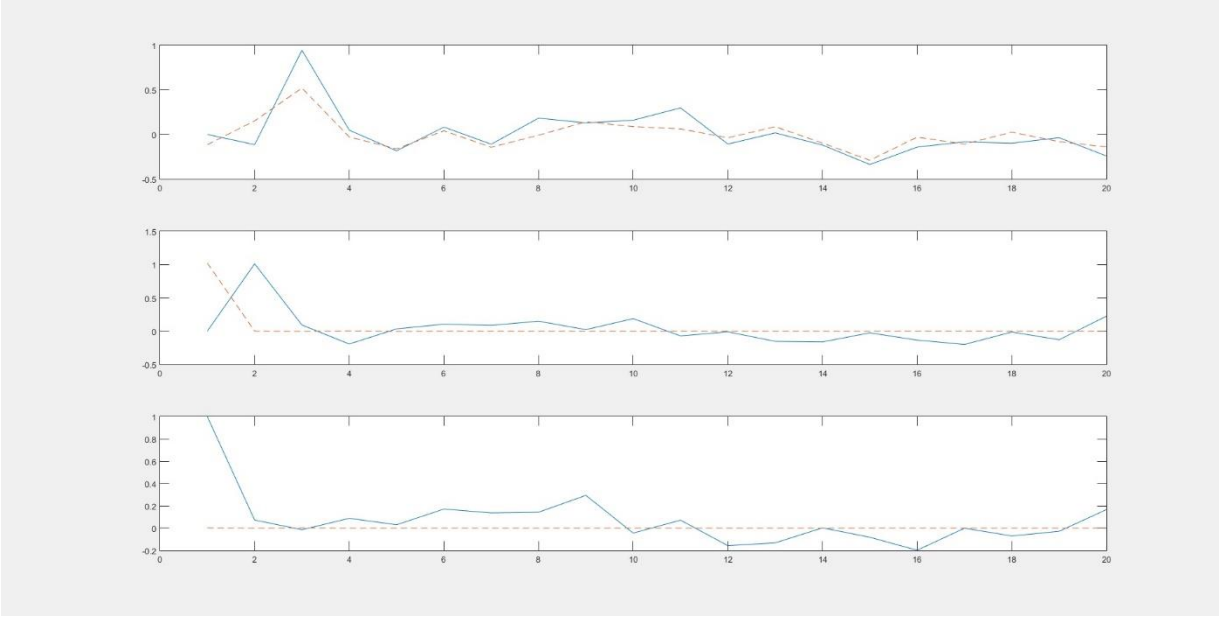

Figure 8. First test of the UKF in MATLAB.

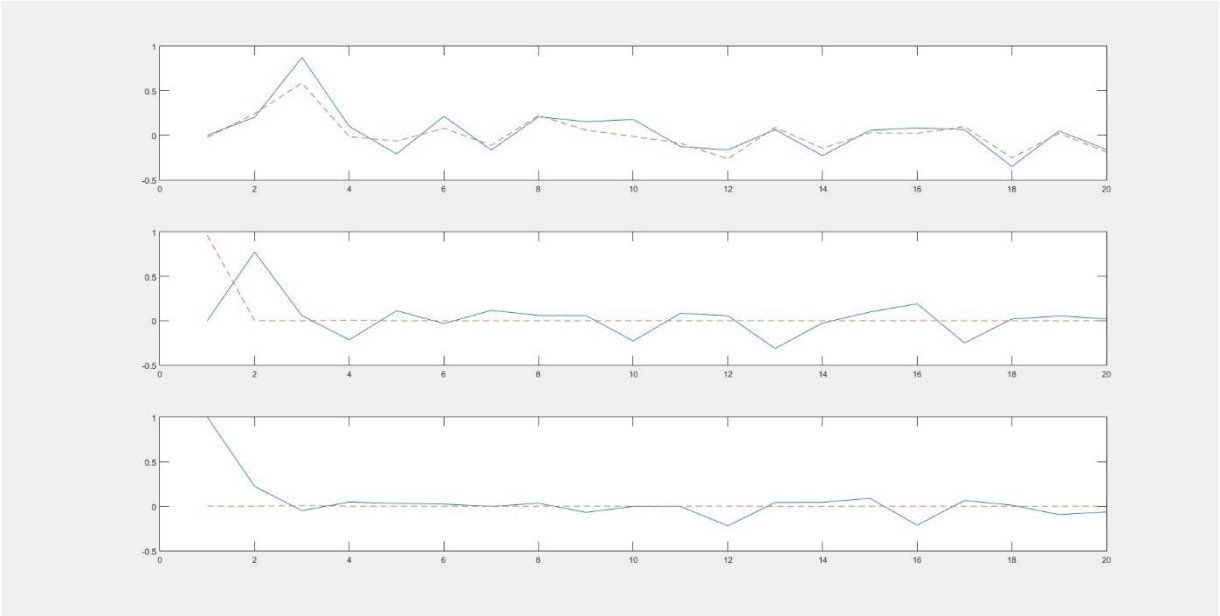

Figure 9. Second test of the UKF in MATLAB.

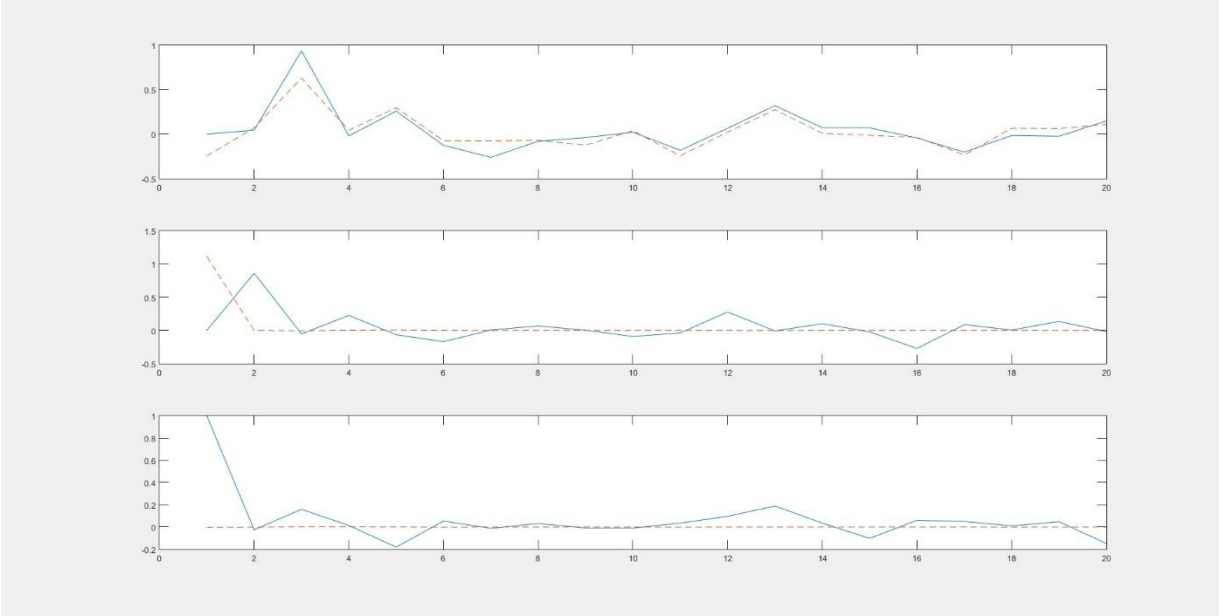

Figure 10. Third test of the UKF in MATLAB.

As shown on the above figures, quite often the estimated state ends up being very near the actual state. It can be concluded from these results that the estimations made by the unscented Kalman filter are quite accurate and could be used in the attitude determination and control system of a satellite.

5 Summary

The main goal of this study was to find out whether the unscented Kalman filter helps us to determine the attitude and to control the satellite most effectively compared to other similar algorithms. Finding the answer to this is important because we need to choose the best algorithm for controlling the attitude of the satellite and to specify the inputs and outputs of the module. To method chosen to achieve our goal was the UKF testing.

As a result of this work, we can say that the study was successful. The results of testing the unscented Kalman filter in MATLAB prove that the UKF helps us to accurately estimate the state of the satellite.

This paper can be used by the programmers in the satellite project to fully understand every aspect of the Kalman filter and the UKF in order to move forward with the project and successfully implement it in code.

References

[1] "Satellite programme < Satellite Programme < MEKTORY < Tallinn University of Technology." [Online]. Available: https://www.ttu.ee/projects/mektory-eng/satellite-programme-3/satellite-programme/. [Accessed: 16-May-2016].

[2] "ADCS." [Online]. Available: http://www.estcube.eu/satelliit/adcs. [Accessed: 16-May-2016].

[3] "Sun sensor," *Wikipedia, the free encyclopedia*. 15-Apr-2016.

[4] "Magnetometer," *Wikipedia, the free encyclopedia*. 17-May-2016.

[5] "Gyroscope," *Wikipedia, the free encyclopedia*. 09-May-2016.

[6] K. F. Jensen and K. Vinther, "Attitude Determination and Control System for AAUSAT3," *Google Docs*. [Online]. Available: https://drive.google.com/file/d/0BweUan7bPqFEWUxKM3YxMVFIQ3c/view?usp=embed_facebook. [Accessed: 03-May-2016].

[7] P. Org, "Software Architecture and Development Tools API for Attitude Control System of TUT Nanosatellite, BSc thesis, TUT, 2016."

[8] "Attitude control," *Wikipedia, the free encyclopedia*. 09-Feb-2016.

[9] "Satellites - Attitude Control: Overview." [Online]. Available: http://satellites.spacesim.org/english/anatomy/attitude/index.html. [Accessed: 17-May-2016].

[10] "Kalman filter," *Wikipedia, the free encyclopedia*. 13-Apr-2016.

[11] L. Kleeman, "Understanding and Applying Kalman Filtering." [Online]. Available: http://biorobotics.ri.cmu.edu/papers/sbp_papers/integrated3/kleeman_kalman_basics.pdf. [Accessed: 13-Mar-2016].

[12] "Kalmani filter," *Vikipeedia, vaba entsüklopeedia*. 16-Aug-2015.

[13] P. Abbeel, "EKF, UKF." [Online]. Available: http://www.cs.berkeley.edu/~pabbeel/cs287-fa11/slides/EKF_UKF--v2.pdf. [Accessed: 13-Apr-2016].

[14] "Extended Kalman filter," *Wikipedia, the free encyclopedia*. 27-Apr-2016.

[15] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation." [Online]. Available: https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf. [Accessed: 13-Mar-2016].

[16] “Special Topics - The Kalman Filter (7 of 55) The Multi-Dimension Model 1 - YouTube.” [Online]. Available: https://www.youtube.com/watch?v=-cD7WkbAIL0. [Accessed: 22-May-2016].

[17] “MATLAB,” *Wikipedia, the free encyclopedia*. 14-Apr-2016.

[18] “BECS / Research / Bayesian Statistical Methods / Downloads / EKF/UKF Toolbox for Matlab.” [Online]. Available: http://becs.aalto.fi/en/research/bayes/ekfukf/. [Accessed: 13-Mar-2016].

[19] “Learning the Unscented Kalman Filter - File Exchange - MATLAB Central.” [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/18217-learning-the-unscented-kalman-filter. [Accessed: 13-Mar-2016].