Scientific Journal of Impact Factor (SJIF): 4.14
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 5, May -2016

Kalman Filter and SLAM Assisted Quadcoptor Navigation

[1]Aniket Datar, [2]Sagar Kadam, [3]Pankaj Deshmukh, [4]Vivek Kumar

[1,2,3,4]*Computer Engineering,D Y Patil College Of Engineering, Ambi, Talegoan,Pune, india*

Abstract —*Recently, there has been increased interest in the development of autonomous flying vehicles. Most of this research is going on in autonomous navigation of these vehicles. However, as most of the proposed approaches are suitable for outdoor operation as they use GPS as key technology for localization. Due to limitation of GPS these frameworks cannot be used for indoor or underwater environment. Only a few techniques have been designed for indoor environments. Also the fast dynamics of flying Quadcopter makes this task more difficult. We present a general navigation system that enables a small-sized quadrotor system to autonomously operate in indoor environments. System is based on SLAM algorithm with use of Kalman Filter. Since the Kalman Filter has some limitations on its use, we compare the variants of Kalman Filter for better computational efficient and reliable model. This system can be used in development of low cost autonomous indoor Quadcopter for area exploration navigation or similar task.*

Keywords*-SLAM, KALMAN FILTER, QUADCOPTER*

I. INTRODUCTION

Quadcopter is a multirotor class helicopter that is lifted and propelled by four rotors these four column of thrust are balanced dynamically. Quads can be equipped with On-Board computer running SLAM algorithm so that it can navigate autonomously. SLAM algorithm uses rangefinders or camera like sensors for analyzing environment this is where system becomes complex as the data form this sensor contain noise.

[7]SLAM algorithm can be divided in two parts first mapping the environment and second localizing the environment. The problem for mapping the environment first you need your location and to localize yourself you need map of environment as the both process are running simultaneously it becomes chicken egg problem. Various techniques are used to solve slam which has different hardware and software requirement one of the technique is use of Kalman filter.

[8]Kalman filter is an algorithm that uses a series of measurements observed over time. It is also known as linear quadratic estimation (LQE). Different versions of Kalman filter are developed like Extended Kalman Filter, Unscented Kalman filter etc.

Different versions of Kalman Filter have different computational requirement and accuracy. As SLAM is used on aerial vehicle the problem becomes three dimensional also the dynamics of Quadcopter are fast which limits computational complexity, hence the computational efficient Kalman filter becomes key parameter along with accuracy, this trade off can be adjusted according to application and cost (available hardware resources).

Further the paper is arranged in following sections
II Architecture
III Related Work
IV proposed system and implementation details
V Result
VI Conclusion

II. ARCHITECTURE

[9]Kalman Filter is based on Systems Dynamic model, Control input to system (known), multiple measurements from sensor to estimate the system varying quantities. A vector of real number is used to represent the state of system. New state is generated at each time discrete time increments by applying a linear operator.[8][9]The Kalman filter modelassumes the true state at time k is evolved from the state at (k − 1) according to

$$X_k = F_k X_{k-1} + B_k u_k + W_k$$

$where$

1. F_k $is\ the\ state\ transition\ model\ which\ is\ applied\ to\ the\ previous\ state\ X_{k-1}$
2. B_k $is\ the\ control-input\ model\ which\ is\ applied\ to\ the\ control\ vector\ u_k$
3. W_k $is\ the\ process\ noise\ which\ is\ assumed\ to\ be\ drawn\ from\ a\ zero\ mean\ multivariate\ normal\ distribution\ with$ $covariance\ Q_k$

$$W_k \sim \mathcal{N}(0, Q_k)$$

At time k an observation (or measurement) Z_k of the true state X_k is made according to

$$Z_k = H_k X_k + V_k$$

where H_k is the observation model which maps the true state space into the observed space and V_k is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_k

$$V_k \sim \mathcal{N}(0, R_k)$$

The state of the filter is represented by two variables:

- $\hat{X}_{k|k}$ *the a posteriori state estimate at time k given observations up to and including at time k;*
- $P_{k|k}$ *the a posteriori error covariance matrix (a measure of the estimated accuracy of the state estimate).*

[8]Predict

Predicted (a priori) state estimate

$$\hat{X}_{k|k-1} = F_k \hat{X}_{k-1|k-1} + B_k u_k$$

Predicated (a priori) estimate covariance

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

[8]Update

Innovation or measurement residual

$$\tilde{y}_k = z_k - H_k \hat{X}_{k|k-1}$$

Innovation (or residual) covariance

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

Optimal Kalman gain

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

Updated (a posteriori) state estimate

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \tilde{y}_k$$

Updated (a posteriori) estimate covariance

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

RGBDSLAM Algorithm

1. *INPUT: Stream Of RGB-D image*
2. *Feature extraction and Matching (SRUF, SIFT, ORB etc.)*
3. *Pose Estimation (RANSAC)*
4. *Pose Refinement (ICP)*
5. *Pose Graph Optimization (HOGMAN)*
6. *OUTPUT: 3D model (Colored Point Cloud)*

III. RELATED WORK

In the last decade, flying platforms received an increasing attention from the research community. Grzonka et al. [1] used a distributed system of computational cost requirement also the system is based on laser range finder and use of two stage Kalman filter based SLAM algorithm. In this model part of the SLAM executes onboard and other part on computer which is connected via Wi-Fi link. In first stage Kalman filter is used to estimate altitude and vertical velocity. Following multi-level SLAM algorithm is used.

Input: beams deflect by mirror at time t:h_t
Input: previous multilevel map:Mˆ
Input: elapsed time:Δt
Input: current pose:x_t= 〖(x〗_(t,) y_(t)) output of SLAM module
Input: previous height state z_(t-1)=(z_(t-1),v_(z_(t-1)))
Input: previous height state uncertainty∑_(z_(t-1))
Input: z-accelaration and uncertainty: a_z,σ_z from IMU
Output: current height state: z_t,∑_(z_t)
Output: current multilevel map:M
1: function Multilevel-SLAM
2: // 1st stage: update height estimate

```
// KF is short for Kalman Filter
(zˆ_(t ),∑ˆ_(z_t ) )=KF(z_(t-1),∑_(z_(t-1) ) ).predictionStep(Δt,a_z,σ_z)
E=Mˆ.at(x_t±Δx).getExistingLevelMatching(h_t,zˆ_t)
if E ≠∅ then
(m˜_t,σ˜_m )=createVirtualHeightMeasurement(h_t,E)
(z_t,∑_(z_t ) )=KF(zˆ_t,∑ˆ_(z_t ) ).measurementUpdate(m˜,σ˜_m)
else
(z_t,∑_(z_t ) )=(zˆ_t,∑ˆ_(z_t ))
end if
//2nd stage: update map
L=estimateLevel(h_t,z_t)
M=Mˆ.addNewLevels(L,x_t)
M=M.updateExistingLevels(L,x_t)
M=M.extendedExistingLevels(L,x_t)
M=M.searchForLoopClouser(x_t)
return 〖z〗_t,∑_(z_t ),M
end function
```

Seongsoo Lee and Sukhan Lee [2] main focus is on low cost consumer based bots which are mainly grounded. Use of monocular camera and odometry for SLAM algorithm is based on feature extraction of 2d image and creating 3d points for navigation, the use of odometry gives robustness in real time. EKF is used for estimating camera orientation. Orthogonal lines of indoor environments were utilized to solve a localization problem within a given map under significant illumination changes. The Harris corner detector method is employed to extract corner like features as points of interest for feature matching.

The 3-D locations of features are simply initialized using a triangulation method, which is a popular method in computer vision.

Huang et al. [3] addressed two key limitations of UKF, first computational complexity, which is cubic in the size of the state vector. In the case of SLAM, where hundreds of landmarks are typically included in the state vector, this increased computational burden can preclude real-time operation. To address this problem, Holmes et al. proposed the square-root UKF (SRUKF) for monocular visual SLAM, which has computational complexity quadratic both in the propagation and in the update phases. This approach offers a significant improvement in terms of computational complexity, at the cost of a considerably more complicated implementation. Additionally the algorithm is an order of magnitude slower than the standard EKF, due to the need to carry out expensive numerical computations.

Amitava Chatterjee and FumitoshiMatsuno [4] Most of the works reported in the area of adaptive Kalman filters have so far concentrated on utilizing new statistical information from innovation sequence to correct the estimation of the states, for adapting the EKF is based on the innovation adaptive estimation (IAE) approach, which was originally proposed and later utilized in combination with fuzzy logic. The neuro-fuzzy model is trained to determine the suitable free parameters of the system i.e., the parameters of the MFs, the output consequence singletons and the output gain.

Paz et al. [5] simulated experiments and the Victoria Park data sets are used to provide evidence of the advantages of the Divide and Conquer SLAM algorithm. Authors show that all processes associated with the move-sense-update cycle of Extended Kalman Filter (EKF) Simultaneous Localization and Mapping (SLAM) can be carried out in time linecomputational complexity per step is reduced from O(n2) to O(n), and the total cost of SLAM from O(n3) to O(n2). Unlike many current large-scale EKF SLAM techniques, this algorithm computes a solution without relying on approximations or simplifications to reduce computational complexity. Estimates and covariances are also available when needed by data association without any further computation. It can be seen that the resulting vehicle and map estimates using 'Divide and Conquer SLAM algorithm' are more precise than those obtained with standard EKF SLAM, and errors with respect to the true value are smaller.

Chang et al. [6] proposed P-SLAM algorithm based on the environmental-structure prediction. With P-SLAM, a mobile robot can predict an unexplored region for a look-ahead mapping allowing a mobile robot to reduce the exploration time and to speed up the SLAM process. In addition, Bayesian formulation of P-SLAM, which merges the predicted maps into the traditional SLAM Bayesian formulation has been derived. The Bayesian formulation generalizes how to use the predicted information in the SLAM problem.

Computer simulations have shown that P-SLAM is very effective in an indoor environment. Finally, implementation of P-SLAM on a Pioneer 3-DX mobile robot in real time showed that P-SLAM improved the mapping results.ar with the number of map features. Authors also describe Divide and Conquer SLAM, an EKF SLAM algorithm in which the

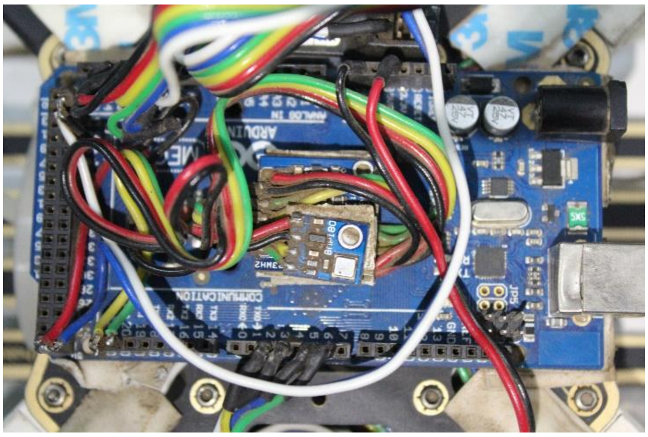
IV. PROPOSED SYSTEM AND IMPLEMENTATIONOTES

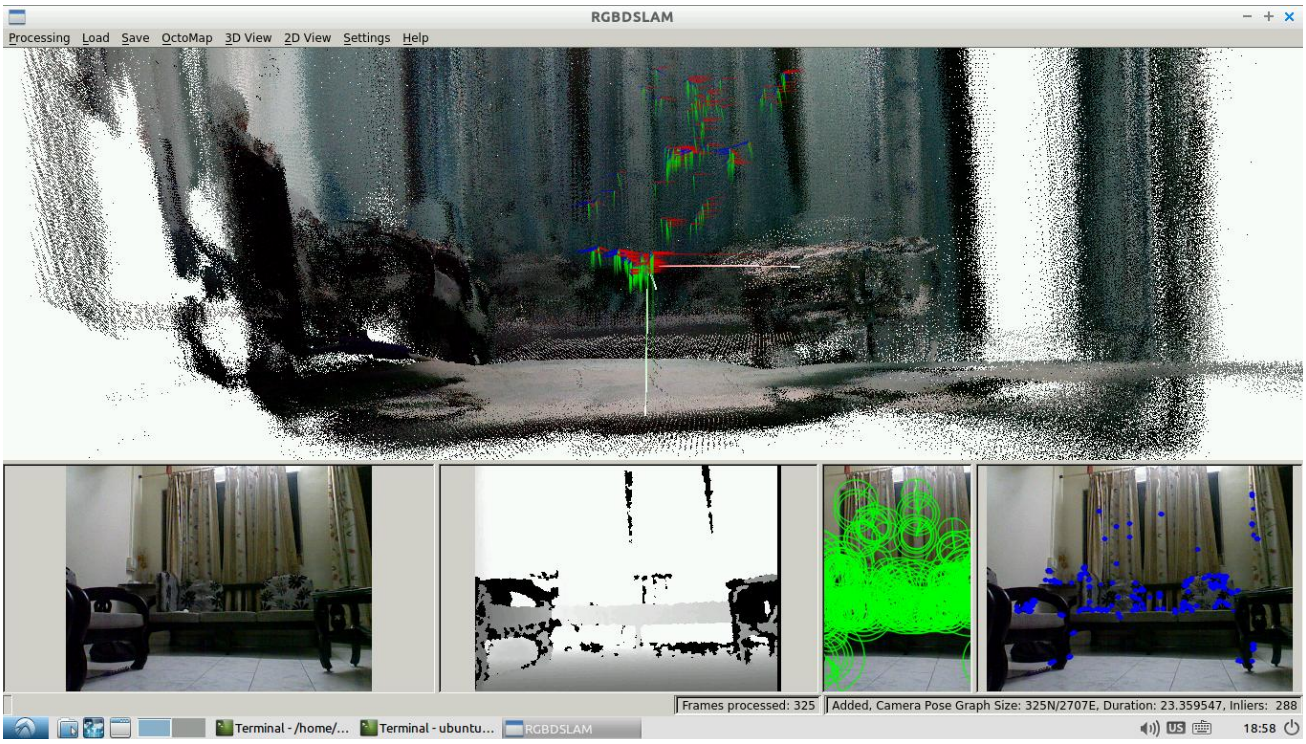
Microsoft Kinect along with Raspberry PI can be used to implement a RGBDSLAM algorithm which is a SLAM algorithm based on RGB+DEPTH camera and feature extraction from the image and maping and generating point cloud based on depth information. System uses raspberry pi with Ubuntu 14.4 lts as OS. ROS indigo is installed on Ubuntu and application for RGBDSLAM run inside ROS. OpenNI library is used as a driver for Kinect which provide rgb and depth image. Maping is done at qvga resolution considering the computing power of raspberry pi and minimum acceptable speed for state update.

Flight controller for quadcopter uses Arduino mega and Multi-Wii firmware's modified version in which complementary filter is replaced by better Kalman filter. IMU consist of a MPU6050 which is both 3 axis accelerometer and gyro, HMC5883Lwhich is a 3 axis magnetometer, and BMP180 which is a pressure sensor.

V. RESULT

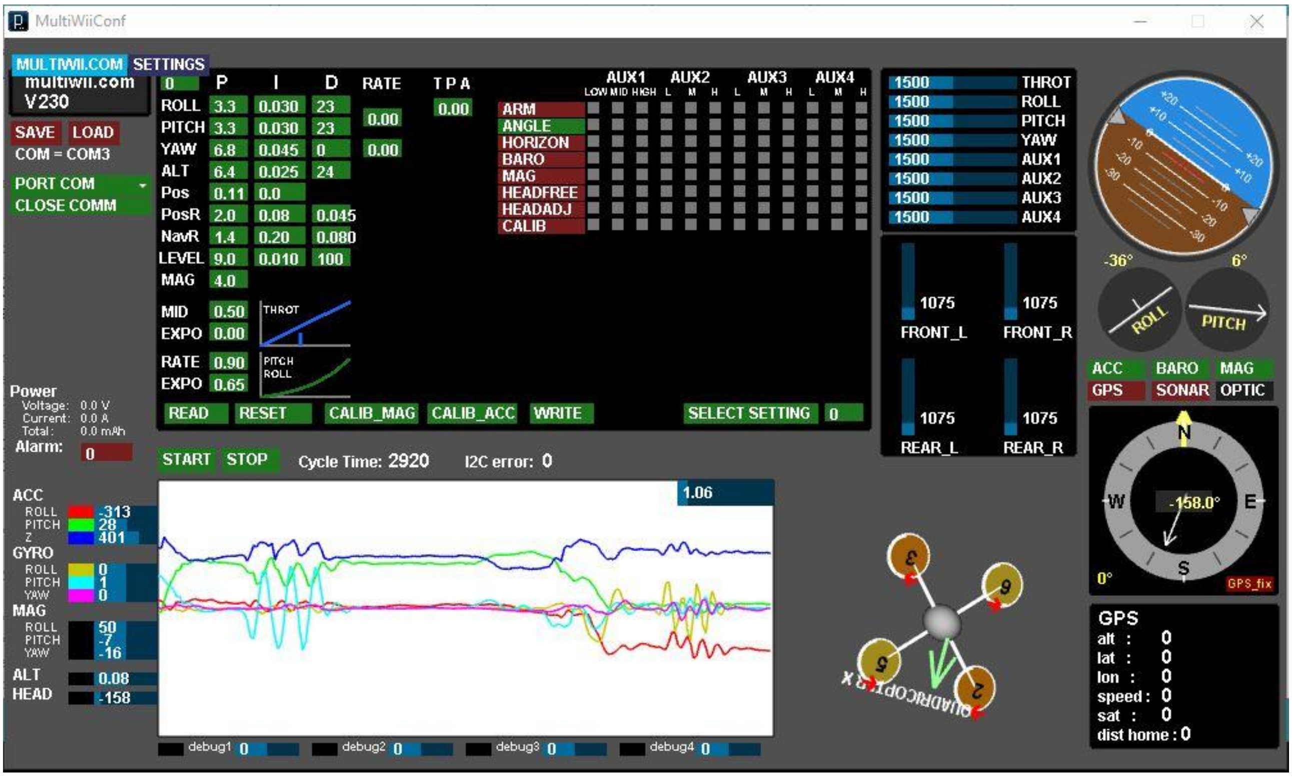
Map generated by SLAM takes nearly 1 second per frame

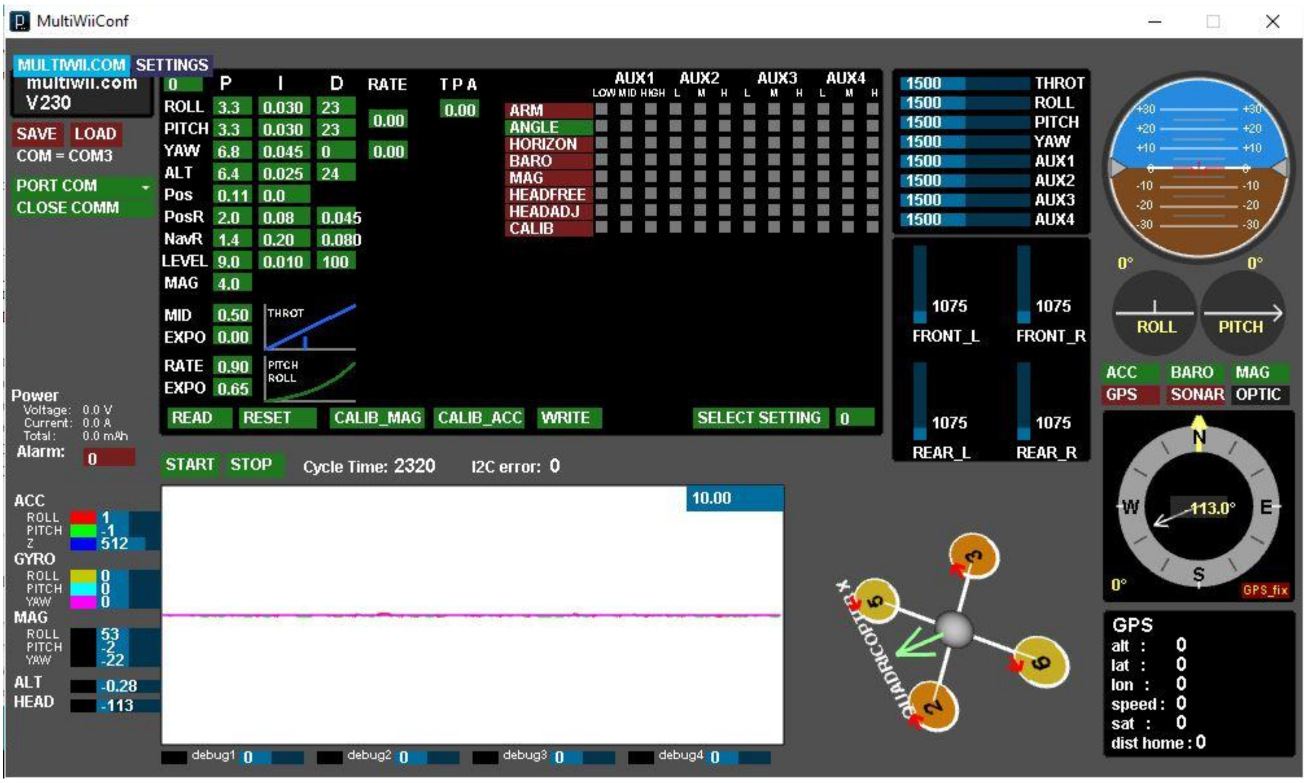




With Kalman filter loot time is about 2.4 miliseconds max which is more than enough for very high dynamic behavior

Normal movement IMU data graph





Steady state nosie level 10x scalling

VI. CONCLUSION

In this paper we studied and compared various variants of filtering techniques used which are based on Kalman filter, by comparison we found that Kalman filter is optimal filter as it takes less computational cost and also provides optimal accuracy which makes it most useful. SLAM algorithm, which is designed to work on low cost systems having limited amount of physical hardware can be implemented by RGBDSLAM method which provide acceptable results.

REFERENCES

[1] SlawomirGrzonka, Giorgio Grisetti, and Wolfram Burgard “A Fully Autonomous Indoor Quadrotor”.
[2] Seongsoo Lee and Sukhan Lee “Embedded Visual SLAM”
[3] Guoquan P. Huang, Anastasios I. Mourikis, Member, IEEE, and Stergios I. Roumeliotis, Senior Member, IEEE “A Quadratic-Complexity Observability-Constrained Unscented Kalman Filter for SLAM”
[4] Amitava Chatterjee and FumitoshiMatsuno, Member, IEEE “A Neuro-Fuzzy Assisted Extended Kalman Filter-Based Approach for Simultaneous Localization and Mapping (SLAM) Problems”
[5] Lina M. Paz, Member, IEEE, Juan D. Tard´os, Member, IEEE, and Jos ´ e Neira, Member, IEEE “Divide and Conquer: EKF SLAM in O(n)”Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.
[6] H. Jacky Chang, Student Member, IEEE, C. S. George Lee, Fellow, IEEE, Yung-Hsiang Lu, Member, IEEE, and Y. Charlie Hu, Member, IEEE “P-SLAM: Simultaneous Localization and Mapping With Environmental-Structure Prediction”
[7] Fernández-Madrigal, J.A. and Blanco, J.L. “ Simultaneous Localization and Mapping for Mobile Robots Introduction and Methods” 2012 ISBN: 978-1466621046
[8] Maria Isabel Ribeiro “Kalman and Extended Kalman Filters: Concept, Derivation and Properties” Available [online] Http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf
[9] “An Introduction to the Extended Kalman Filter” Available [online] <http://www.goddardconsulting.ca/extended-kalman-filter.html>