Hierarchical Extended Kalman Filter with Frequent Propagation for UAV Localization

2017 SURF Final Report
Author : Hyung Ju Suh
Mentor: Prof. Soon-jo Chung, Dr. Kyunam Kim, Karena Cai

***Abstract*— Kalman filters have been used extensively for platform localization, where the best working estimate of a platform's 3D position and orientation state must be calculated from recursive sensor inputs. While state-of-art localization methods such as GPS-inertial or visual-inertial have a good degree of accuracy of this estimation, they suffer from a low-sampling rate due to limitations in computation. We try to deal with this problem by fast propagation of inertial measurements, and a hierarchical structure of the Kalman Filter. The experimentation platform is a UAV equipped with two sensors: Qualcomm Snapdragon Flight, and the VectorNav Inertial Measurement Unit (IMU). Snapdragon Flight is a visual-inertial sensor platform that runs its internal Kalman Filter, while the VectorNav IMU is a 9 DOF inertial sensor that gives linear acceleration and angular velocity. We compare the results of our implemented Kalman Filter with path from Vicon Motion Capture-system, showing the validity of the filter as well as its ability to accurately portray state estimations within finer time intervals.**

I. INTRODUCTION

Kalman Filters[1] and its variants have been used widely for estimating the time-varying position and orientation of a platform, which is a process known as localization. The formulation of the filter usually involves fusing information from multiple sensors with different information, such as position and acceleration, orientation and angular velocity, and et cetra. In the field of UAV localization, the state-of-art methods are given by GPS-inertial, and visual-inertial. GPS-inertial method fuses position data from GPS and acceleration/angular velocity data from the inertial measurement unit (IMU), while visual-inertial methods use visual features to measure position to fuse with the IMU measurements.

While these methods are successful in achieving accurate localization, both methods suffer from the problem of low sampling rate. GPS-inertial localization has limitations in the sampling rate of GPS (1-10Hz), while visual-inertial localization suffers from limitations in computation time, usually not being able to achieve over 30Hz. Such shortcomings in sampling rate make both methods unsuitable for highly dynamic flight, which requires frequent feedback for control.

This research proposes a solution to this problem of low sampling rate by frequent propagation of IMU measurements, and a hierarchical structure to the Kalman Filter that helps in reducing computation time. This solution is tested on a UAV sensor platform using Snapdragon Flight, a visual-inertial localization sensor, and VectorNav IMU. The validity of the implemented Kalman Filter is shown in the results section.

II. PREVIOUS WORK

Real-time visual-inertial odometry in UAVs has not been utilized until recently due to hardware limitations. The sampling rate of sensor fusion is integral to a robust estimation algorithm, but on-board computers with small form factors were simply not efficient enough to carry out these computations in real-time.

The work by S. Weiss et al. [2] was one of the first to provide a real-time on board solution for visual-inertial geometry by using extended kalman filters on vision data and inertial sensor. From here some papers have used different techniques in order to create a more robust odometry out of vision and inertial sensors.

As mentioned before, the key factor in integrating the vision and inertial sensors together is undoubtedly the different sampling rates of two sensors - while IMU is able to typically publish data typically in the order of 800-1000Hz, most vision-based odometry requires a lot more computation time and still stays in the order of 10Hz. These differing rates inevitably affect the ability to estimate the states in real-time. Along with the aforementioned problem of sampling rates, such difference in frequency also leads to issues in synchronization between the two sensors.

The work by Leutenegger et al. [3] propose a solution by using nonlinear optimization on past keyframes instead of filtering methods and are able to achieve good results. In this paper the synchronization issue is handled by fitting a Gaussian distribution between the IMU samples to infer the IMU sample at the frame of the camera data. While such optimization-based methods have the ability to be more accurate, it introduces some complexities to the system both in terms of processing speed and memory. Currently, utilizing past state information seems to be a consensus between many papers, Forster et al. [4] introduces a filtering-based method that introduces one level of abstraction and preintegrating IMU samples before they are filtered with the vision features. Another work by M. Shelley [5] deals with this problem on the vision side by triangulating the in-between frames and calculating the residuals to fuse with the IMU. On the other hand, Yang et al. [6] uses the extended Kalman Filter with bias terms and incorporates the asynchronization issue within the error term and continuously filters the new measurements.

III. Kalman Filter Formulation

A. System Dynamics

First, it is necessary to describe the system dynamics in order to achieve the propagation step of the Kalman Filter. Due to the nature of attitude dynamics, the system is non-linear and we deal with a Extended Kalman Filter (EKF), which is a variant of the Kalman Filter to deal with non-linear systems. The state of the system is defined as

$$\boldsymbol{x} = [\boldsymbol{p}_B^{NED}, \boldsymbol{v}^B, \boldsymbol{b}_a^B, \boldsymbol{q}_B^{NED}, \boldsymbol{b}_g^B]^T$$

where $\boldsymbol{p}_B^{NED}$ and $\boldsymbol{q}_B^{NED}$ are respectively position and quaternion-orientation of the body seen from NED (North-East Down inertial frame), $\boldsymbol{v}^B$ is the velocity of the body seen from the body frame, and $\boldsymbol{b}_a^B$ and $\boldsymbol{b}_g^B$ are respectively the accelerometer and gyroscope biases seen from the body.

Along with this state definition, we define the control vector as inputs from the IMU, such that

$$\boldsymbol{u} = [\boldsymbol{a}^B, \boldsymbol{\omega}^B]^T$$

With these definitions, the system dynamic is formulated as a non-linear function of the state and control vectors.

$$\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, \boldsymbol{u}) = \frac{d}{dt}\begin{bmatrix} \boldsymbol{p}_B^{NED} \\ \boldsymbol{v}^B \\ \boldsymbol{b}_a^B \\ \boldsymbol{q}_B^{NED} \\ \boldsymbol{b}_g^B \end{bmatrix}$$

$$= \begin{bmatrix} R(\boldsymbol{q}_B^{NED})\boldsymbol{v}^B \\ (\boldsymbol{a}^B - \boldsymbol{b}_a^B) - [\boldsymbol{\omega}^B - \boldsymbol{b}_g^B]_\times \boldsymbol{v}^B + R^T(\boldsymbol{q}_B^{NED})\boldsymbol{g}^{NED} \\ \mathbf{0^{3\times 1}} \\ \frac{1}{2}\Omega(\boldsymbol{\omega}^B - \boldsymbol{b}_g^B)\boldsymbol{q}_B^{NED} \\ \mathbf{0^{3\times 1}} \end{bmatrix}$$

Here $R(\boldsymbol{q})$ is the rotation matrix derived from the quaternion (A.1), $[]_\times$ is the skew-symmetric matrix representation of the cross product (A.2), and Ω is the matrix representation of the Hamiltonian product between the quaternion and the expanded angular velocity vector (A.3). We then propagate the system dynamics recursively using Euler integration, where the propagation matrix is defined by the Jacobian of $f(\boldsymbol{x}, \boldsymbol{u})$ in regards to the state.

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k \qquad \boldsymbol{F} = \boldsymbol{I} + \frac{\partial f}{\partial \boldsymbol{x}} \cdot \Delta t$$

B. Attitude-Position Hierarchy

Observing the system dynamics, it can be seen that the dynamics of the first three position states $\boldsymbol{x}_p = [\boldsymbol{p}_B^{NED}.\boldsymbol{v}^B, \boldsymbol{b}_a^B]^T$ and the latter two attitude states $\boldsymbol{x}_q = [\boldsymbol{q}_B^{NED}, \boldsymbol{b}_g^B]^T$ can be decoupled. The first three states require attitude and gyroscope bias information to achieve propagation, but the latter two attitude states can be obtained independent of the position values. Thus it is possible to achieve a hierarchical structure where attitude information is first estimated, then fed into the position variables. Thus we describe a hierarchical decoupled structure using two filters respectively describing position, and attitude.

The attitude filter can be first described with the system dynamics of

$$\frac{d\boldsymbol{x}_q}{dt} = f_q(\boldsymbol{x}_q, \boldsymbol{\omega}^B) = \begin{bmatrix} \frac{1}{2}\Omega(\boldsymbol{\omega}^B - \boldsymbol{b}_g^B)\boldsymbol{q}_B^{NED} \\ \mathbf{0}^{3\times 1} \end{bmatrix}$$

and the position filter is propagated by the dynamics of

$$\frac{d\boldsymbol{x}_p}{dt} = f_p(\boldsymbol{x}_p, \boldsymbol{x}_q, \boldsymbol{a}^B, \boldsymbol{\omega}^B)$$

$$= \begin{bmatrix} R(\boldsymbol{q}_B^{NED})\boldsymbol{v}^B \\ (\boldsymbol{a}^B - \boldsymbol{b}_a^B) - [\boldsymbol{\omega}^B - \boldsymbol{b}_g^B]_\times \boldsymbol{v}^B + R^T(\boldsymbol{q}_B^{NED})\boldsymbol{g}^{NED} \\ \mathbf{0^{3\times 1}} \end{bmatrix}$$

Along with the dynamics propagation from IMU inputs, the estimation step is done by measurement inputs from Snapdragon Flight. The visual-inertial odometry device is capable of measuring both position and orientation. Thus we define the measurement vector and matrices by

$$\boldsymbol{z}^q = \boldsymbol{C}^q \boldsymbol{x}_q \qquad \boldsymbol{z}^p = \boldsymbol{C}^p \boldsymbol{x}_p$$

where the measurement matrices are defined as

$$\boldsymbol{C}^q = [\boldsymbol{I}^{4\times 4}, \mathbf{0}^{4\times 3}] \qquad \boldsymbol{C}^p = [\boldsymbol{I}^{3\times 3}, \mathbf{0}^{3\times 6}]$$

The rest of the process follows the EKF construction, where $\boldsymbol{Q}$ is the covariance matrix of propagation (related to the IMU), $\boldsymbol{R}$ the covariance of the estimation step (related to Snapdragon Flight). We describe the tuning procedures of these parameters in the later section. The rest of the estimated process is done by computing the Kalman Gain (A.4)

We prove that this hierarchical structure is more computationally efficient compared to propagating the whole state due to the non-linear nature of this Kalman Gain computation, which is computed using matrix inversion. Denoting the Kalman gain as $\boldsymbol{K}$, we see from appendix A.4 that

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T(\boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \boldsymbol{R}_k)^{-1}$$

This process of obtaining inversion can be done by Gaussian-Jordan elimination, but matrix factorization (decomposition) is usually utilized to speed up the process. Although the computational complexity of this factorization process is dependent upon matrix types, the fastest complexity is around $O(n^{2.376})$ using the Coppersmith-Winograd (CW) algorithm [7]. The CW algorithm is also the fastest for matrix multiplication. Then denoting the number of position states as n_p and the number of orientation states as n_q, we see that in non-asymptotic terms, most computations are much faster by

$$(n_p + n_q)^{2.376} \geq n_p^{2.376} + n_q^{2.376}$$

In our configuration of $n_p = 9$ and $n_q = 7$, we can calculate a 60.49% decrease in computation time. For cubic complexity ($O(n^3)$) the reduction reaches 73.83%.

C. Frequent Propagation

To deal with the aforementioned problem of low sampling rate, we propagate as soon as the IMU data becomes available, and estimate as soon as Snapdragon Flight's data is available. The IMU is capable of publishing data at 800Hz, while Snapdragon Flight publishes at 30Hz. This results in multiple propagation steps in between the estimation steps. Thus denoting $\hat{x}_{k+i|k}$ as the $(k+i)^{th}$ propagation step after the k^{th} estimation step, we change the propagation step by

$$\hat{\boldsymbol{x}}_{k-i|k-n} = f(\hat{\boldsymbol{x}}_{k-i-1|k-n}, \hat{\boldsymbol{u}}_{k-i-1})$$

where n is the number of propagation steps in between estimations, and $n \geq i$. Figure 1 Summarizes the overall construction of the filter using hierarchical structure and frequent propagation.

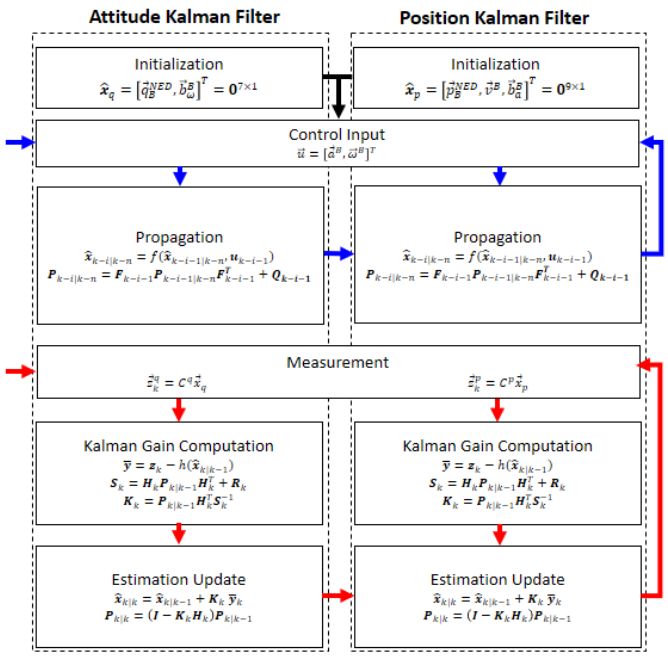

Fig. 1. Summary of the Extended Kalman Filter with Hierarchical structure and Frequent Propagation

D. Observability Analysis

We finally note that the system is observable since there are direct position and orientation measurements, from which velocity and both bias terms are observable. To see that this is true, we use the former definitions of the discrete propagation matrix $\boldsymbol{F}$ and the measurement matrix $\boldsymbol{C}$, that are defined such that

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k \qquad \boldsymbol{z}_k = \boldsymbol{C}\boldsymbol{x}_k$$

We note that the discrete propagation matrices and the measurement matrix each differ for the attitude and position filters ($\boldsymbol{F}^q$,$\boldsymbol{C}^q$ and $\boldsymbol{F}^p$,$\boldsymbol{C}^p$). For both filters we construct the observability matrix $\boldsymbol{O}^q$ and $\boldsymbol{O}^p$ such that

$$\boldsymbol{O}^q = \begin{bmatrix} \boldsymbol{C}^q \\ \boldsymbol{C}^q\boldsymbol{F}^q \\ \boldsymbol{C}^q(\boldsymbol{F}^q)^2 \\ \dots \\ \boldsymbol{C}^q(\boldsymbol{F}^q)^6 \end{bmatrix} \qquad \boldsymbol{O}^p = \begin{bmatrix} \boldsymbol{C}^p \\ \boldsymbol{C}^p\boldsymbol{F}^p \\ \boldsymbol{C}^p(\boldsymbol{F}^p)^2 \\ \dots \\ \boldsymbol{C}^p(\boldsymbol{F}^p)^8 \end{bmatrix}$$

Here we use the fact that the number of states for each filters are $n_q = 7$ and $n_p = 9$. By determining the rank of the matrix we confirm that

$$\text{rank}(\boldsymbol{O}^q) = n_q = 7 \qquad \text{rank}(\boldsymbol{O}^p) = n_p = 9$$

from which it can be concluded that both states in the filters are always observable.

IV. Platform Specification

We test our Kalman Filter on a UAV platform custom-built by Skylift Global. The platform is capable of high-velocity flight (over 60mph) and has computational power needed for various tasks such as Simultaneous Localization and Mapping (SLAM) and Fast Motion Planning (FMP).

Qualcomm Snapdragon Flight was later added to the the UAV as a part of SURF, during which many additional hardware preparations were done. Snapdragon Flight is one of the first commercialized visual-inertial solution to localization. Since the product is still early in release, we came across some hardware and firmware issues such as board's vibration damping. Since the sensor is relatively new in the market, this research is one of the few to utilize this sensor as a part of the localization stack.

Processor
- NVidia Jetson TX1 Microprocessor
- Qualcomm Snapdragon Flight Board

Flight Controller
- PixHawk PX4 Flight Controller

Visual / Laser Sensors
- ZED Stereo Vision Camera
- Sony IMX135 4K High-Res Camera
- Omnivision OV7251 Optical Flow Camera
- Lidar-Lite 3 Laser Rangefinder

Inertial Sensors
- VectorNAv VN-100 9DOF IMU
- InvenSense MPU-9250 9DOF IMU

Fig. 2. Experimentation Platform Specification

The platform's overall hardware specification is given in Figure 2, and Figure 3 illustrates a photo of the platform. Figure 4 describes additional hardware work, and Snapdragon Flight.

V. Filter Tuning & Initialization

Another important perspective of implementing a Kalman Filter lies in tuning the filter. Although many variables are given as output parameters of the sensor, some are characteristics of the sensor that has to be tuned or measured in order to achieve a reliable filtering process.

Fig. 3. UAV Sensor Platform with TX1 Processor

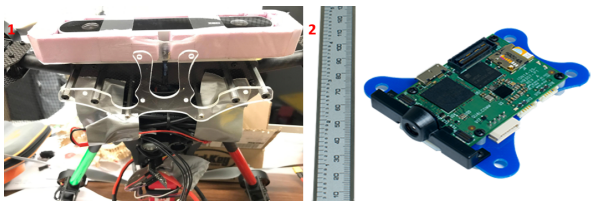

Fig. 4. 1. Hardware mount for Snapdragon 2. Qualcomm Snapdragon Flight

A. Bias Initialization

Although the biases are observable in the system (i.e. the filter will automatically estimate the bias terms), it is important to have a priori knowledge of the biases to achieve a good propagation step during the first few seconds of the filter. In a well-designed filter the bias terms can be simply calculated by the filter output, and may be initialized with that value. The attitude filter's bias can be directly observed by propagating when the gyroscope bias term is zero.

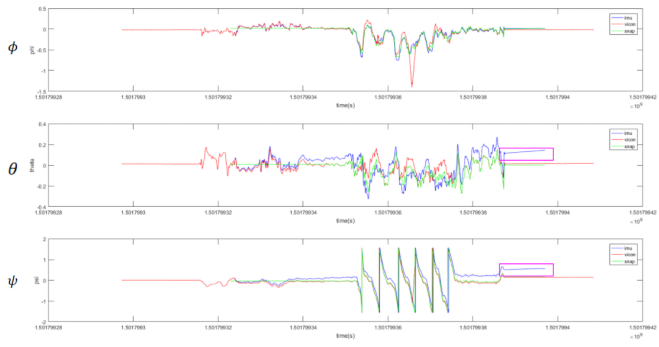

Fig. 5. Attitude Propagation in ZYX-Euler angles without the bias term. Magenta highlights the linear offset caused by a gyroscope bias term.

Figure 5 describes propagation of angular velocity measurements with zero bias term. Calculating the slope of this linearly increasing offset gives us a good initialization process for the gyroscope bias term. With these values, propagation alone achieves the result of Figure 6.

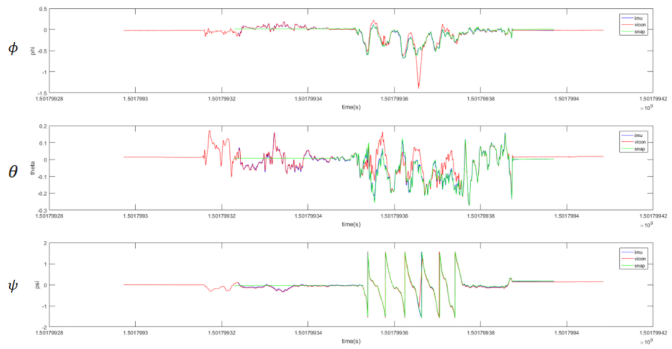

Fig. 6. Attitude Propagation in ZYX-Euler angles with a properly initialized bias terms.

In the position filter, the constant accelerometer bias is integrated twice to give a quadratic function. The process is, however, less evident to observe compared to the attitude filter because the centripetal term also builds up this bias. Thus while it is possible to initialize the bias based on a quadratic fit, we chose to initialize it by choosing what the filter values converge to.

B. Covariance Tuning

Determining the value of the covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ is integral to the performance of the filter, as they provide stochastic representations of how much to trust the sensors. This process is usually done empirically depending on the accuracy of the sensors, and we extracted data from experiments for values of $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$. For the discrete formulation of these covariance matrices, we first define constant matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ and update the covariances using the time-dependent form:

$$\boldsymbol{Q}_k = \boldsymbol{Q} \cdot \Delta t \qquad \boldsymbol{R}_k = \boldsymbol{R}/\Delta t$$

In order to obtain values of $\boldsymbol{Q}$ and $\boldsymbol{R}$, we can obtain the values of the stochastic variables w_k and v_k. w_k, the propagation noise, is obtained by subtracting the value of propagation from ground-truth obtained using Vicon. v_k, the measurement noise, is obtained by subtracting the sensor measurement from ground-truth obtained using Vicon. The graph of these noises are described in Figure 7.

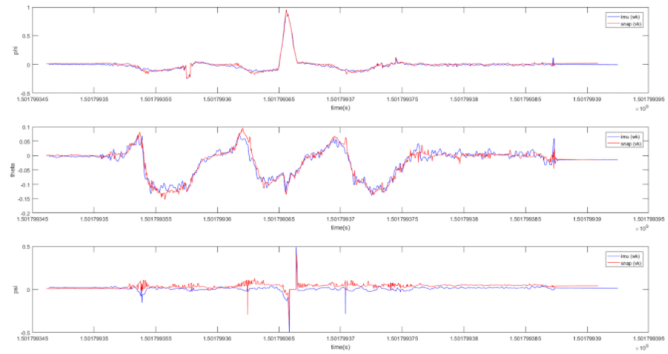

Fig. 7. Graph of measured stochastic variables $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ in the attitude filter

After obtaining these values, covariance is calculated from the time samples and accordingly tuned to the values of $\boldsymbol{Q}$ and $\boldsymbol{R}$.

VI. Results

For experimentation, we collect data from Snapdragon Flight, VectorNav IMU, and the Vicon Motion Capture System to compare the results of each sensor against the implemented Kalman Filter.

A. Results of the attitude filter

The result of the attitude filter is shown in ZYX-Euler angles in Figure 8.

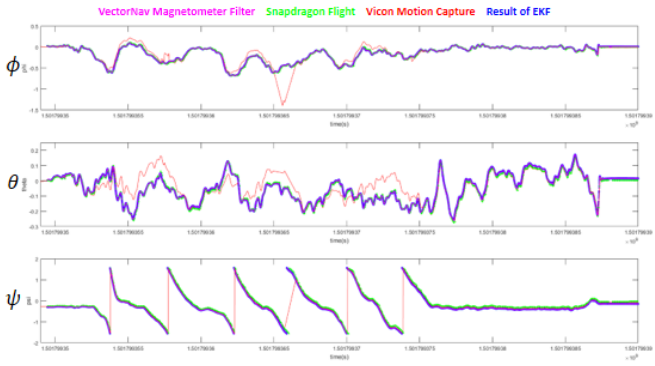

Fig. 8. Attitude filter result in ZYX-Euler Angles

We can observe that the implemented Kalman Filter agrees well with the results of VectorNav's internal filter, even though the filter was run on IMU's uncompensated (unfiltered) data. Unfortunately at this stage Vicon suffers from experimental conditions (marker obstruction) but the overall filter works well to describe orientation.

B. Results on Position

While orientation goes through a single integration step for propagation (angular velocity to quaternions), position must go through two integration steps before position can be recovered from acceleration. For this reason, The result of the position filter is shown in Figure 9.

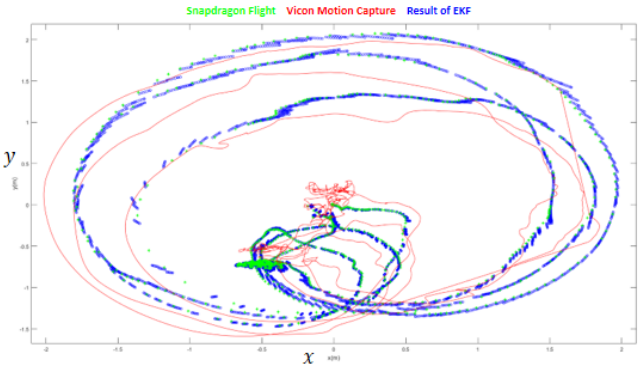

Fig. 9. XY-position filter result

We can see that after some time period, the bias terms are correctly compensated for and the Kalman Filter provides valid results within finer intervals against the Vicon motion capture system.

VII. Conclusion

To deal with the low-frequency nature of GPS and visual-inertial localization, we present a solution using position-attitude hierarchy and frequent propagation. We asymptotically prove that the former method reduces computation time by 60.49%, and show that the new filter can achieve IMU's sampling rate while providing valid estimations between each sensor measurements. This not only provides localization results within finer time intervals, but also helps in achieving faster feedback rates for control of highly dynamic aerial robots.

We then tested the filter on a UAV platform equipped with a visual-inertial sensor of Snapdragon Flight, which provides position and orientation measurement, and VectorNav IMU, which provides acceleration and angular velocity measurement. The results show that the implemented filter is valid against the Vicon motion capture system, with high sampling rates.

It is notable that the same method of frequent propagation can be utilized for most formulations of the Kalman Filter involving a propagation step, and a measurement (estimation) step. This process will always help achieve more estimations within finer intervals in scenarios where the two steps have drastically different sampling rates. In contrast, the attitude-hierarchy will not be usable for systems that do not have direct orientation measurements, such as GPS-inertial. Most IMUs, however, come with magnetometers that provide this measurement. In this case the whole attitude filter runs at a much faster sampling rate compared to the position filter.

Acknowledgment

I would like to thank Professor Soon-Jo Chung, Dr. Kyunam Kim, and Karena Cai for helping me with the research. It has been a great experience for me to learn the theoretical foundations for state estimation, and also to handle real-life data to verify the formulation. I would also like to thank Dr. Kiyo Tomiyasu and Eiko Tomiyasu for funding my work and further motivating me towards a better researcher. This research was performed as a part of SURF (Summer Undergraduate Research Fellowship) at California Institute of Technology.

References

[1] Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering. 82: 35.

[2] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments, in 2012 IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 957964

[3] S. Leutenegger, P. Furgale, V.Rabaud, M. Chli, K. Konolige, R. Siegwart, "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization", Proceedings of Robotics: Science and Systems, 2013

[4] C. Forster, L.Carlone, F.Dellaert, D.Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation", Proceedings of Robotics: Science and Systems, 2015

[5] M. Shelley, "Monocular Visual Inertial Odometry on a Mobile Device", Munchen Technical University Master Thesis, 2014
[6] J. Yang, A. Dani, S. Chung, S. Hutchinson, "Vision-based Localization and Robot-centric Mapping in Riverine Environments", Journal of Field Robotics, 2017
[7] Coppersmith, Don; Winograd, Shumuel (1990) "Matrix Multiplication via Arithmetic Progressions", Journal of Symbolic Computation, 9(3): 251
[8] Phil Kim (2010), "Kalman Filter for Beginners with MATLAB Examples", ISBN 1463648359
[9] Paul Zarchan, Howard Musoff (2000), "Fundamentals of Kalman Filtering: A Practical Approach" American Institute of Aeronautics and Astronautics, Incorporated. ISBN 978-1-56347-455-2.

APPENDIX

1. Rotation Matrix derived from Quaternion

From a quaternion $\boldsymbol{q} = [q_w, q_x, q_y, q_z]$, we derive the rotation matrix $R(\boldsymbol{q})$ by

$$R(\boldsymbol{q}) = \begin{bmatrix} 1-2q_y^2-2q_z^2 & 2q_xq_y-2q_zq_w & 2q_xq_z+2q_yq_w \\ 2q_xq_y+2q_zq_w & 1-2q_x^2-2q_z^2 & 2q_yq_z-2q_xq_w \\ 2q_xq_z-2q_yq_w & 2q_yq_z+2q_xq_w & 1-2q_x^2-2q_y^2 \end{bmatrix}$$

2. Skew-symmetric Matrix Representation of Cross Product

From a vector $\boldsymbol{v} = [v_x, v_y, v_z]^T$ and $\boldsymbol{w} = [w_x, w_y, w_z]^T$, we recover the cross product $\boldsymbol{v} \times \boldsymbol{w}$ by using a skew-symmetric representation of $\boldsymbol{v}\times$, such that

$$\boldsymbol{v} \times \boldsymbol{w} = [\boldsymbol{v}]_\times \boldsymbol{w} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}$$

3. Derivative of Quaternion

We define the derivative of a quaternion by a matrix representation of the Hamiltonian product between the quaternion and the expanded angular velocity vector, such that $\boldsymbol{\omega}_q = [0, w_x, w_y, w_z]$ and

$$\frac{d\boldsymbol{q}}{dt} = \frac{1}{2}\boldsymbol{\omega} \otimes \boldsymbol{q} = \frac{1}{2}\Omega(\boldsymbol{\omega})\boldsymbol{q} = \frac{1}{2}\begin{bmatrix} -[\boldsymbol{\omega}]_\times & \boldsymbol{\omega} \\ -(\boldsymbol{\omega})^T & 0 \end{bmatrix}\boldsymbol{q}$$

4. Extended Kalman Filter (EKF)

We expand more on the Extended Kalman Filter (EKF) formulation for notation clarification and further explanation. As mentioned in section 3, upon defining the dynamics in terms of the derivative, the recursive notation is such that

$$\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, \boldsymbol{u})$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_k, \boldsymbol{u}_k) \cdot \Delta t + \boldsymbol{w}_k$$

where $\boldsymbol{w}_k$ is the stochastic Gaussian noise associated with propagation. While the system dynamic is non-linear, the measurement matrix is not and we define a measurement vector $\boldsymbol{z}_k$ such that

$$\boldsymbol{z}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k$$

where $\boldsymbol{v}_k$ is the stochastic Gaussian variable related to sensor noise. Then we carry out the prediction steps by

$$\boldsymbol{x}_{k+1|k} = \boldsymbol{x}_{k|k} + f(\boldsymbol{x}_k, \boldsymbol{u}_k) \cdot \Delta t$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k \boldsymbol{P}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q}_k$$

where $\boldsymbol{Q}_k$ is the covariance matrix of $\boldsymbol{w}_k$, and F is the discrete Jacobian such that

$$\boldsymbol{F} = \boldsymbol{I} + \frac{\partial f}{\partial \boldsymbol{x}} \times \Delta t$$

The sensor measurements are updated such that

$$\bar{\boldsymbol{y}}_{k+1} = \boldsymbol{z}_{k+1} - \boldsymbol{C}\boldsymbol{x}_{k+1|k}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^T(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{C}_k^T + \boldsymbol{R}_{k+1})^{-1}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}\bar{\boldsymbol{y}}_{k+1}$$

$$\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k)\boldsymbol{P}_{k+1|k}$$

where $\boldsymbol{K}$ is the near-optimal Kalman Gain for this step, and $\boldsymbol{R}_k$ is the covariance matrix related to the sensor stochastic variable $\boldsymbol{v}_k$.