Genetic Algorithm based Kalman Filter Adaptation Algorithm for Magnetic and Inertial Measurement Unit

Simone A. Ludwig
North Dakota State University
Fargo, ND, USA
simone.ludwig@ndsu.edu

***Abstract*—A magnetic and inertial measurement unit (MIMU) usually measures acceleration, rotation rate, and earth's magnetic field in order to determine a body's attitude. In order to find the orientation information using all sensor information a fusion algorithm is used. Extended Kalman filtering is a well known technique that has been widely applied in many applications used for state estimation. The main idea behind the algorithm is that a series of observations over time is used to produce estimates of unknown states leading to more precise orientation information than compared to when estimated states were only based on single observations. However, one problem exists, namely the Extended Kalman filtering solution becomes very poor when abrupt acceleration motions occur. In order to avoid this problem, an optimization algorithm can be integrated into the filtering mechanism as a dynamic model correction. Thus, this paper introduces a genetic algorithm to be utilized as a noise-adaptive mechanism in order to tune the Extended Kalman filter process.**

I. Introduction

Orientation information in a three-dimensional space is one of the most important components required for navigation, guidance and control of an object such as an unmanned air vehicle, a drone, etc. An Attitude and Heading Reference System (AHRS) is used to determine the orientation of an object which it is attached to. In the last decade, investigations of attitude estimation have been conducted with low-cost Micro Electro-Mechanical Systems (MEMS) [1][2]. Even though MEMS sensors are light weight and small in size and thus applicable to many areas (e.g., human motion tracking), however, they suffer from noise and errors that get accumulated over time. Therefore, the calibration and validation of the AHRS is very important in order to achieve good performance and accuracy.

For an AHRS, sensor data measured by a gyroscope, accelerometer, and magnetometer also known as MIMU (Magnetic and Inertial Measurement Unit) can be used. An AHRS consists of an algorithm which provides the orientation of the sensors with respect to a navigation frame. The orientation is usually represented as Euler angles (roll, pitch and yaw). The aim of an AHRS is to combine the sensor data from the gyroscope, accelerometer and magnetometer to obtain the orientation. An AHRS is conceptually divided into two blocks in order to provide the orientation: (1) from the gyroscope, and (2) from the accelerometer and magnetometer. These two blocks need to be weighted in order to retrieve the optimal orientation information. There are different types of filters, orientation filters and general purpose Bayesian estimation filters. Examples of orientation filters are the complementary filter [3], and the Kalman filter [4]. Examples of general purpose Bayesian estimation filters are Mahony [5], and Madgwick [6].

Extended Kalman filtering is a well known technique that has been widely used in many applications for state estimation. The main idea is that the algorithm uses a series of observations over time to produce estimates of unknown states. In particular, the estimate of a series of observations is more precise than one based on a single observation alone. The problem of the Extended Kalman filtering technique, however, is when abrupt acceleration motions occur, then the filtering solutions become very poor. In order to avoid this problem, an optimization algorithm can be embedded into the filtering mechanism to serve as a dynamic model correction unit. Thus, this paper uses a genetic algorithm to be utilized as a noise-adaptive mechanism in order to tune the Extended Kalman filter process.

II. Adaptive Extended Kalman Filter

This section introduces the Extended Kalman filter first, followed by the discrete-time extended version as well as the adaptive extended version. Afterwards, the genetic algorithm is described in general followed by a description on how the algorithm is used to assist the Extended Kalman filter process.

A. Extended Kalman Filter

Figure 1 shows the block diagram of the Extended Kalman filter. The figure shows a closed loop system consisting of four steps: (1) prediction, (2) Kalman gain, (3) update, and (4) quaternion normalization. The MIMU sensor input is provided during the update step. After each of the four steps are executed, the output in the form of an estimated orientation is provided.

The Extended Kalman filter is used to model nonlinear systems that deal with cases that are guided by the nonlinear stochastic differential equations such as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{u}(t) \quad (1)$$

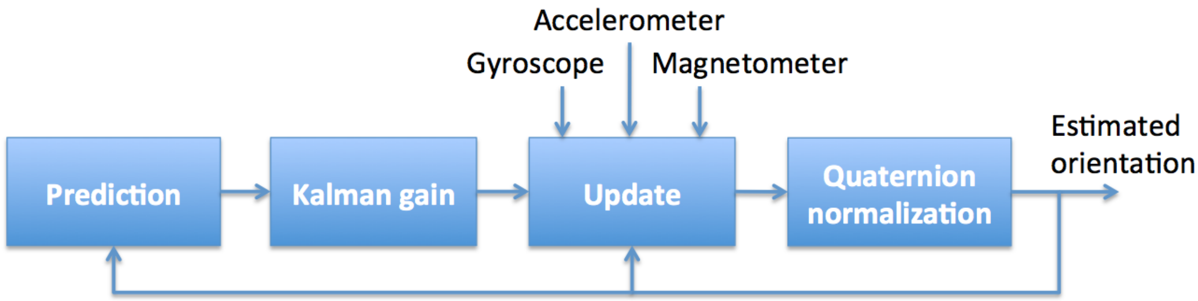

Fig. 1. Sensor data of gyroscope, accelerometer, magnetometer

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, t) + \mathbf{v}(t) \tag{2}$$

where $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are white noise sequences with zero means and they are mutually independent:

$$\begin{aligned} E[\mathbf{u}(t)\mathbf{u}^T(\tau)] &= q\delta(t-\tau); \quad E[\mathbf{v}(t)\mathbf{v}^T(\tau)] \\ &= r\delta(t-\tau); \quad E[\mathbf{u}(t)\mathbf{v}^T(\tau)] = \mathbf{0} \end{aligned} \tag{3}$$

where $\delta(t)$ is the Dirac delta function, $E[\cdot]$ is the expectation, and superscript T represents the matrix transpose.

In the discrete-time equivalent form, Equations (1) and (2) can be written as:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + \mathbf{w}_k \tag{4}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \tag{5}$$

where $\mathbf{x}_k \in \Re^n$ is the state vector, $\mathbf{w}_k \in \Re^m$ is the process noise vector, $\mathbf{z}_k \in \Re^m$ measurement vector, and $\mathbf{v}_k \in \Re^m$ measurement noise vector. In Equations (4) and (5), $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero mean Gaussian white noise sequences having zero cross-correlation with each other such as:

$$\mathbf{E}[\mathbf{w}_k\mathbf{w}_i^T] = \mathbf{Q}_k\delta_{ik} \tag{6}$$

$$\mathbf{E}[\mathbf{v}_k\mathbf{v}_i^T] = \mathbf{R}_k\delta_{ik} \tag{7}$$

$$\mathbf{E}[\mathbf{w}_k\mathbf{v}_i^T] = 0 \quad for \quad all \quad i \quad and \quad k \tag{8}$$

where $\mathbf{Q}_k$ is the process noise covariance matrix, and $\mathbf{R}_k$ is the measurement noise covariance matrix. The Kronecker delta function δ_{ik} is given by:

$$\delta_{ik} = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{9}$$

B. Discrete-Time Extended Kalman Filter

The Discrete-Time Extended Kalman Filter algorithm is given in Algorithm 1. Equations (10) - (12) are measurement update equations, whereas Equations (13) - (15) are time update equations from step k to $k+1$. The equations transform the measurement value into a priori estimation in order to calculate an improved posteriori estimation. $\mathbf{P}_k$ is the error covariance matrix defined by $\mathbf{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$, for which $\hat{\mathbf{x}}_k$ is an estimate of the system state vector $\mathbf{x}_k$ and $\mathbf{K}_k$ is the Kalman gain matrix. The algorithm starts with initial condition values $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. As soon as a new measurement $\mathbf{z}_k$ becomes available over time, the estimation of states and the corresponding error covariance follow recursively. More details on the Extended Kalman Filter can be found in [7], [8], [9].

C. Adaptive Extended Kalman Filter

As mentioned previously, poor knowledge of the noise statistics may seriously degrade the performance of the Extended Kalman filter. Thus, an adaptive extended Kalman filter can be used to make use of an noise-adaptive filter in order to estimate the noise covariance matrices as closely as possible. In [10], adaptive approaches are categorized into:

- Bayesian
- Maximum likelihood
- Correlation and covariance matching

The idea behind the correlation and covariance matching is to match the actual value of the covariance of the residual consistent with its theoretical value. Thus, this category of adaptive approaches has been widely studied. From the incoming measurement $\mathbf{z}_k$, and the optimal prediction $\hat{\mathbf{x}}_k^-$ obtained during the previous step, the innovation sequence is defined as:

$$\mathbf{v}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k^- \tag{16}$$

The innovation sequence represents the discrepancy between the predicted measurement and the actual measurement. In particular, it constitutes the additional information available to the filter as a consequence of the new observation $\mathbf{z}_k$. The innovation sequence $\mathbf{v}_k$ is a zero-mean Gaussian white noise

Algorithm 1 Discrete-Time Extended Kalman Filter Algorithm

1. Initialize the state vector and state covariance matrix:

$$\hat{\mathbf{x}}_{\mathbf{0}}^{-} \text{ and } \mathbf{P}_{\mathbf{0}}^{-}$$

2. Compute the Kalman gain matrix from the state covariance and estimated measurement covariance:

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{10}$$

3. The prediction error vector from the Kalman gain matrix is multiplied to obtain the state correction vector and update state vector as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{\mathbf{k}}^{-} + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k^{-}], \quad with \quad \hat{\mathbf{z}}_k^{-} = \mathbf{h}(\hat{\mathbf{x}}_k^{-}, k) \tag{11}$$

4. Update of error covariance:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^{-} \tag{12}$$

5. Prediction of new state vector and state covariance matrix:

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{f}(\hat{\mathbf{x}}_k, k) \tag{13}$$

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k\mathbf{P}_k\mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{14}$$

where the linear approximation equations for the system and measurement matrices are obtained by the following, respectively:

$$\mathbf{\Phi}_k \approx \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_k^{-}}; \quad \mathbf{H}_k \approx \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_k^{-}} \tag{15}$$

sequence. By taking the variances of both sides, we obtain the theoretical covariance given as:

$$\mathbf{C}_{vk} = \mathbf{E}[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{R}_k \tag{17}$$

which can be written as:

$$\mathbf{C}_{vk} = \mathbf{H}_k(\mathbf{\Phi}_k\mathbf{P}_k\mathbf{\Phi}_k^T + \mathbf{Q}_k)\mathbf{H}_k^T + \mathbf{R}_k \tag{18}$$

Then, the estimate of $\mathbf{R}_k$ can be calculated by:

$$\hat{\mathbf{R}}_k = \hat{\mathbf{C}}_{vk} - \mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T \tag{19}$$

where $\hat{\mathbf{C}}_{vk}$ is the statistical sample variance estimate of $\mathbf{C}_{vk}$. $\mathbf{C}_{vk}$ can be computed by averaging the values inside a moving estimation window of size N:

$$\hat{\mathbf{C}}_{vk} = \frac{1}{N}\sum_{j=j_0}^{k}\mathbf{v}_j\mathbf{v}_j^T \tag{20}$$

where N is the number of samples (also referred as the window size), $j_0 = k - N + 1$ is the first sample inside the estimation window. Thus, based on the residual based estimate, the estimate of the process noise $\mathbf{Q}_k$ can be calculated by:

$$\hat{\mathbf{Q}}_k = \frac{1}{N}\sum_{j=j_0}^{k}\triangle\mathbf{x}_j\triangle\mathbf{x}_j^T + \mathbf{P}_k - \mathbf{\Phi}_k\mathbf{P}_{k-1}\mathbf{\Phi}_k^T \tag{21}$$

where $\triangle\mathbf{x}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k^{-}$. This equation can be written in terms of the innovation sequence as:

$$\hat{\mathbf{Q}}_k \approx \mathbf{K}_k\hat{\mathbf{C}}_{vk}\mathbf{K}_k^T \tag{22}$$

Regarding the window size N, if the value is too small, then the estimation of the the measurement covariance will be too noisy. If the window is large, then the estimation of the measurement covariance will be smoother, however, at the expense of a long transient time. The window size is usually empirically determined in order to provide statistical smoothing. An adaptive extended Kalman filter can be used as a noise adaptive filter to estimate the noise covariance matrices to make the results of the Kalman filter more accurate. The main benefit of the adaptive algorithm lies in that it keeps the covariance consistent with the actual performance of the filter. For more details please refer to [11].

Another approach for the adaptivity of the algorithm is the use of fading memory [11]. The idea is to apply a factor matrix to the predicted covariance matrix in order to deliberately increase the variance of the predicted state vector, i.e., to incorporate a fading memory:

$$\mathbf{P}_{k+1}^{-} = \lambda_k\mathbf{\Phi}_k\mathbf{P}_k\mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{23}$$

where $\lambda_k = diag(\lambda_1, \lambda_1, ..., \lambda_m)$. The main difference between the fading memory algorithm is the calculation of the scale factor matrix λ_k. One way would be to assign the scale factors as constants. When $\lambda_i \leq 1$ $(i = 1, 2, ..., m)$ the filtering is in steady processing, while for $\lambda_i > 1$ the filtering might be unstable. For the case where $\lambda_i = 1$, the standard Kalman filtering deteriorates.

Another type of adaptation can be accomplished by introducing a scale factor directly to the $\mathbf{Q}_k$ and/or $\mathbf{R}_k$ matrices [11]. In order to account for greater uncertainty, the covariances need to be updated. This can be done by one of the following methods:

- $\mathbf{Q}_k \rightarrow \mathbf{Q}_{k-1} + \triangle\mathbf{Q}_k; \quad \mathbf{R}_k \rightarrow \mathbf{R}_{k-1} + \triangle\mathbf{R}_k$
- $\mathbf{Q}_k \rightarrow \mathbf{Q}_k\alpha^{-(k+1)}; \quad +\mathbf{R}_k \rightarrow \mathbf{R}_k\beta^{-(k+1)},$ $\alpha \geq 1; \quad \beta \geq 1$
- $\mathbf{Q}_k \rightarrow \alpha\mathbf{Q}_k; \mathbf{R}_k \rightarrow \beta\mathbf{R}_k$

Thus, Equations (4) and (5) can be updated as follows:

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k\mathbf{P}_k\mathbf{\Phi}_k^T + \alpha\mathbf{Q}_k \tag{24}$$

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \beta\mathbf{R}_k]^{-1} \tag{25}$$

When $\alpha = \beta = 1$, then we have the standard Kalman Filter.

D. Optimization Approach: Genetic Algorithm

The optimization method referred to as genetic algorithm is part of a group called evolutionary algorithms. Evolutionary algorithms are inspired by natural phenomena of biological evolution whereby the common idea is that given a population of individuals, natural selection (biologically referred to as survival of the fittest) is used to improve the fitness of the overall population. For example, given a function to be maximized, a set of candidate solutions is randomly created and a fitness function is used as a fitness measure (the higher the better) is applied. Based on this fitness measure, some of the better candidates are chosen to undergo recombination and mutation (recombination is applied to two candidates and results in two new candidates, whereas mutation is only applied to one candidate and results in one new candidate). After recombination and mutation are applied, the newly created candidates replace the old ones and the next generation begins. This process is repeated until a candidate with sufficient quality is determined or a predefined number of iterations is reached [12].

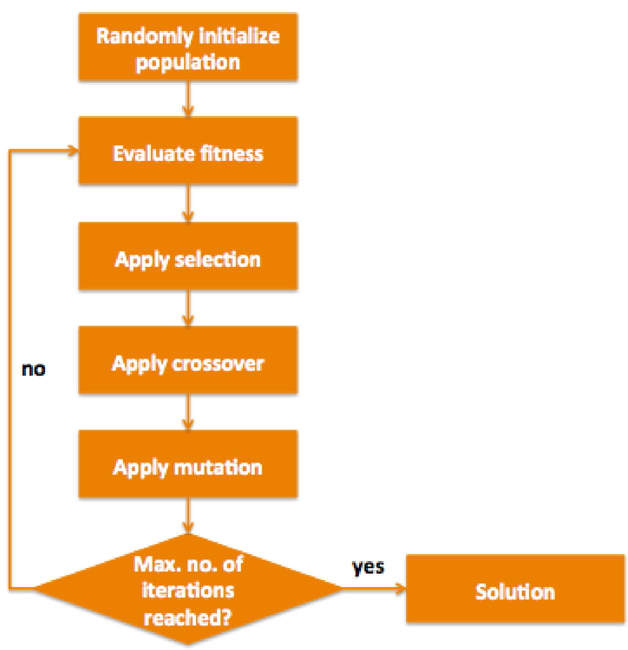

Fig. 2. Flowchart diagram of Genetic Algorithm

Figure 2 shows the overview diagram of the steps in a Genetic algorithm. First, the problem (see next subsection) needs to be encoded using a chromosome representation, and a fitness equation needs to be defined (see next subsection). Afterwards, the selection method needs to be chosen, and the crossover and mutation operations need to be defined. The overall flow of the algorithm is as follows: first, a randomly generated population is initialized, then the fitness of each chromosome (solution) is evaluated, afterwards the selection process is run whereby the roulette wheel selection method was chosen. Then, crossover and mutation operations are applied in order to recombine potential better solutions. The algorithm terminates once the maximum number of iterations has been reached.

III. Genetic Algorithm based Extended Kalman Filter (GA-EKF)

The innovation information is used for assisting the GA optimization. In particular, the difference between $\hat{\mathbf{C}}_{vk}$ and $\mathbf{C}_{vk}$ is detected, referred to as $detect_1$, as the trace of the innovation covariance matrix:

$$detect_1 = \frac{tr(\hat{\mathbf{C}}_{vk})}{tr(\mathbf{C}_{vk})} \tag{26}$$

The $detect_1$ parameter can be used to detect any divergence or outliers during the adaptive filtering process. The second parameter for detection, $detect_2$, is given by:

$$detect_2 = \frac{1}{N}\sum_{j=j_0}^{k}(\boldsymbol{\Delta}\hat{\mathbf{x}}_k) \tag{27}$$

where:

$$\boldsymbol{\Delta}\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^- = \mathbf{K}_k\mathbf{v}_k \tag{28}$$

$detect_2$ is utilized to assist $detect_1$. Both parameters are used to identify the degree of dynamical change of motion. Thus, α is adapted based on the following fitness function:

$$fitness = \frac{tr(\hat{\mathbf{C}}_{vk})}{tr[\mathbf{H}_k(\boldsymbol{\Phi}_k\mathbf{P}_k\boldsymbol{\Phi}_k^T + \alpha\mathbf{Q}_k)\mathbf{H}_k^T + \mathbf{R}_k]} \tag{29}$$

An optimized α will keep the predicted covariance matrix consistent with the actual one. The details of the filtering process of GA-EKF is given in Figure 3.

IV. Experiments

In this section, the description of the data set used is given, followed by the parameters of the simulation experiments, and the results that were obtained.

A. Data Description

We evaluated the accuracy of the orientation estimation algorithm using the publicly available MAV data sets [13] with ground-truth orientation in Euler angles from a motion-capture system. The datasets are recorded using an AscTec 'Pelican' quadrotor (see Figure 4), flying in an indoor environment of size $10m \times 10m \times 10m$. The quadrotor is equipped with eight Vicon cameras, and the data sets were collected by performing 1, 2, and 3 loops, respectively.

Figure 5 shows the trajectory traveled by the quadrotor, tracked by the motion capture system, during the '1loop' experiment (2D top view (left) and 3D side view (right) of the trajectory traveled by the quadrotor during the '1LoopDown' experiment). The data sets include acceleration and angular velocity readings from the IMU.

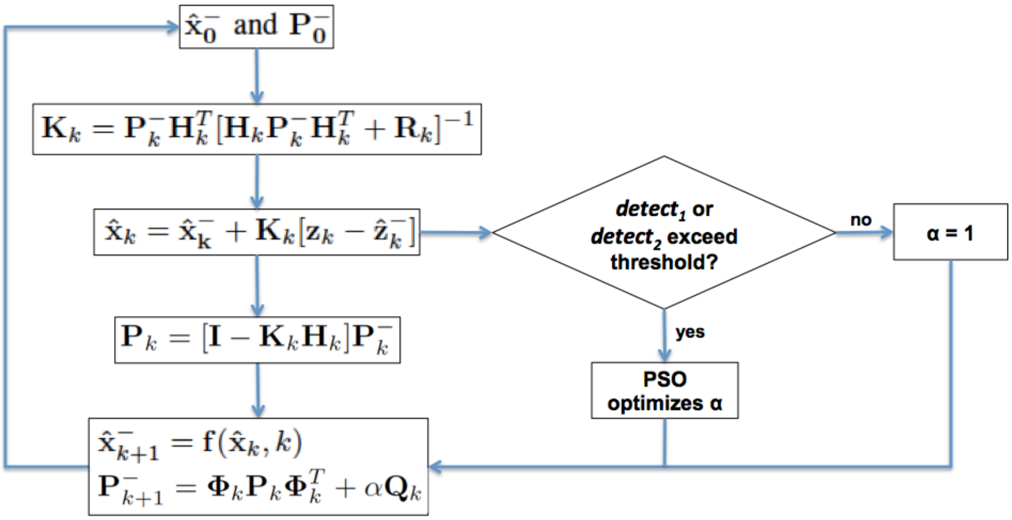

Fig. 3. Flowchart diagram of GA-assisted Kalman filter process

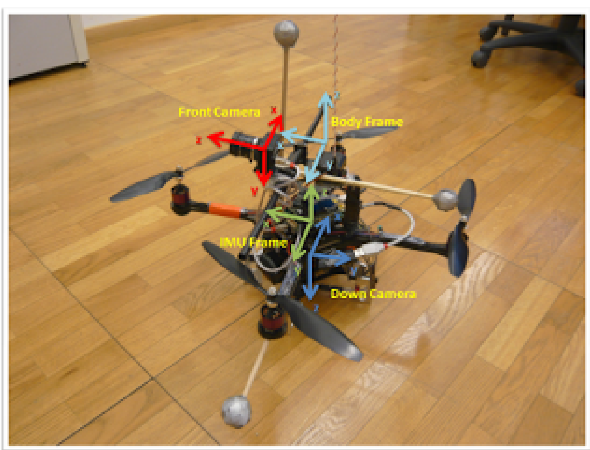

Fig. 4. Quadrotor frames [14]

B. Simulation Experiments

The parameters of the GA that were established after preliminary experiments are the following:

- Size of a chromosome population = 30
- Number of genes in a chromosome = 8
- Crossover probability = 0.8
- Mutation probability = 0.001

C. Results of Experiments

We compare our method (GA-EKF) against the orientation estimation proposed by Madgwick [16], the Extended Kalman Filter (EKF) proposed by Sabatini [17], and the algorithm provided by the low-level processor of the AscTec quadrotor [13], whose output is provided by the data sets in Euler angles. The comparison values for Madgwick, AscTech and EKF were obtained from [13].

Tables I to III show the Root Mean Square Error (RMSE) for roll, pitch, and yaw, respectively. From the values it can be observed that for all three datasets GA-EKF scores best outperforming the comparison methods. Please note that the Madgwick filter is a constant gain filter and thus, the performance varies based on the value chosen for the gain. This gain was obtained by minimizing the RMSE.

Figure 6 shows the norm RMSE values for Madgwick, AscTec, EKF, and the proposed GA-EKF, respectively. Again, the norm RMSE values for GA-EKF obtained the best results compared to the other three approaches.

TABLE I
RMSE OF ROLL ANGLE IN RADIANS

Data set	Madgwick	AscTec	EKF	GA-EKF
1LoopDown	0.037	0.0464	0.0287	0.0259
2LoopsDown	0.047	0.0338	0.0314	0.0303
3LoopsDown	0.405	0.0315	0.0331	0.0294

TABLE II
RMSE OF PITCH ANGLE IN RADIANS

Data set	Madgwick	AscTec	EKF	GA-EKF
1LoopDown	0.0336	0.0369	0.0284	0.0267
2LoopsDown	0.0369	0.0313	0.0384	0.0284
3LoopsDown	0.036	0.0329	0.0392	0.0311

TABLE III
RMSE OF YAW ANGLE IN RADIANS

Data set	Madgwick	AscTec	EKF	GA-EKF
1LoopDown	0.2543	0.3388	0.1888	0.1951
2LoopsDown	0.9229	0.3182	0.3345	0.2576
3LoopsDown	1.3327	0.3255	0.3545	0.3091

Table IV shows the computational time of Madgwick, EKF and GA-EKF. The most time consuming algorithm is GA-EKF

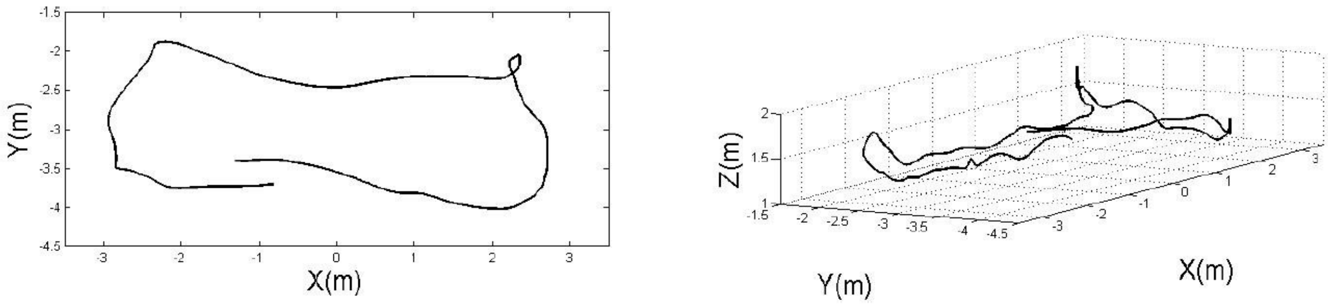

Fig. 5. Quadrotor during 1loop experiment [15]

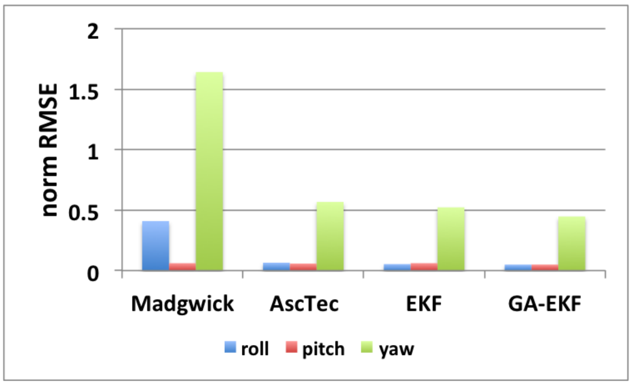

Fig. 6. Norm RMSE values for all approaches

followed by EKF, and the Madgwick filter by far performs the fastest.

TABLE IV
COMPUTATIONAL TIME IN SECONDS

Data set	Time (average)	Std. dev.
Madgwick	1.2839	0.7101
EKF	7.0408	0.2342
GA-EKF	8.3827	0.2595

V. CONCLUSION

The use of the Extended Kalman filter to find the orientation information of data collected from a magnetic and inertial measurement unit (MIMU) is very common. The Kalman filter in particular has proved to be a well-established technique in the area of object tracking for UAVs. The idea behind the algorithm is that a series of observations over time is used to produce estimates of unknown states leading to more precise orientation information than compared to when only single observations were used. However, one problem with the Kalman filter exists, which is that the tracking solution becomes very poor when abrupt acceleration motions occur. In order to alleviate this problem, an optimization algorithm can be embedded into the filtering mechanism to act as a dynamic model correction unit. Thus, this paper introduced a genetic algorithm to be used as a noise-adaptive mechanism in order to take care of abrupt acceleration motion errors.

The simulation experiments investigated the orientation estimation of three data sets and compared the GA-EKF algorithm with the Madgwick filter, the Extended Kalman Filter (EKF), and the algorithm provided by the low-level processor of the AscTec quadrotor. The orientation estimation were given in Euler angles with the ground truth provided by the data sets. The results show that the proposed GA-EKF algorithm scored best in terms of the Root Mean Square Error (RMSE) for roll, pitch, and yaw, respectively. The same trend was seen for the norm RMSE results.

REFERENCES

[1] M. Wang, Y. C. Yang, R. R. Hatch, Y. H. Zhang, Adaptive filter for a miniature MEMS based attitude and heading reference system. In Proceedings of IEEE Position Location and Navigation Symposium, Monterey, CA, USA, April 26-29, 2004.
[2] R. Zhu, D. Sun, Z. Y. Zhou, D. Q. Wang, A linear fusion algorithm for attitude determination using low cost MEMS-based sensors. Measurement, 40, 322-328, 2007.
[3] M. Euston, P. Coote, R. Mahony, J. Kim, T. Hamel, A complementary filter for attitude estimation of a fixed-wing UAV. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 22-26 September 2008.
[4] G. Welch, G. Bishop, An Introduction to the Kalman Filter; Technical Report 95-041; University of North Carolina: Chapel Hill, NC, USA, 24 July 2006.
[5] S. Mahony, T. Hamel, J.-M. Pflimlin, Nonlinear complementary filters on the special orthogonal group. Auto. Control IEEE Transac., 53, 1203-1218, 2008.
[6] S. Madgwick, An Efficient Orientation Filter for Inertial and Inertial/-Magnetic Sensor Arrays; Technical Report; Report x-io and University of Bristol: Bristol, UK, 30 April 2010.
[7] R. Brown, P. Hwang, P. Introduction to Random Signals and Applied Kalman Filtering, Wiley, New York, NY, 1997.
[8] J. A. Farrell, M. Barth, The Global Positioning System & Inertial Navigation, McGraw-Hill, New York, NY, 1999.
[9] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.
[10] R. K. Mehra, Approaches to adaptive filtering, IEEE Transactions on Automatic Control, Vol. AC-17, pp. 693-8, 1972.
[11] A. H. Mohamed, K. P. Schwarz, Adaptive Kalman filtering for INS/GPS, Journal of Geodesy, Vol. 73, pp. 193-203, 1999.
[12] A. E. Eiben, and J. E. Smith, Introduction to Evolutionary Computing, Springer, Natural Computing Series, 1st edition, 2003.
[13] G. H. Lee, M. Achtelik, F. Fraundorfer, M. Pollefeys, and R. Siegwart, A Benchmarking Tool for MAV Visual Pose Estimation, International Conference on Control, Automation, Robotics and Vision (ICARCV'10), Singapore, December, 2010.
[14] MAV DataSet, https://sites.google.com/site/gimheelee/home/mavdataset, last retrieved January 2018.
[15] R. G. Valenti, I. Dryanovski, J. Xiao, Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs, Sensors 2015, 15(8), 19302-19330, 2015.

[16] S. O. H. Madgwick, A. J. L. Harrison, A. Vaidyanathan, Estimation of IMU and MARG orientation using a gradient descent algorithm. In Proceedings of the IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, July 2011.

[17] A. M. Sabatini, Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation. Sensors 2011, 11, 9182-9206, 2011.