Unsupervised Tuning of Filter Parameters Without Ground-Truth Applied to Aerial Robots

Li, Shushuai; De Wagter, Christophe; de Croon, Guido C. H. E.

DOI
10.1109/LRA.2019.2930480

Publication date
2019

Document Version
Accepted author manuscript

Published in
IEEE Robotics and Automation Letters

Unsupervised Tuning of Filter Parameters without Ground-truth Applied to Aerial Robots

Shushuai Li, Christophe De Wagter, and Guido C.H.E. de Croon

***Abstract*—Autonomous robots heavily rely on well-tuned state estimation filters for successful control. This paper presents a novel automatic tuning strategy for learning filter parameters by minimizing the innovation, i.e., the discrepancy between expected and received signals from all sensors. The optimization process only requires the inputs and outputs of the filter without ground-truth. Experiments were conducted with the Crazyflie quadrotor, and all parameters of the extended Kalman filter (EKF) are well tuned after one 10-second manual flight. The proposed method has multiple advantages, of which we demonstrate two experimentally. First, the learned parameters are suitable for each individual drone, even if their particular sensors deviate from the standard, e.g., by being noisier. Second, this manner of self-tuning allows one to effortlessly expand filters when new sensors or better drone models become available. The learned parameters result in a better state estimation performance than the standard Crazyflie parameters.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Localization, Sensor Fusion.**

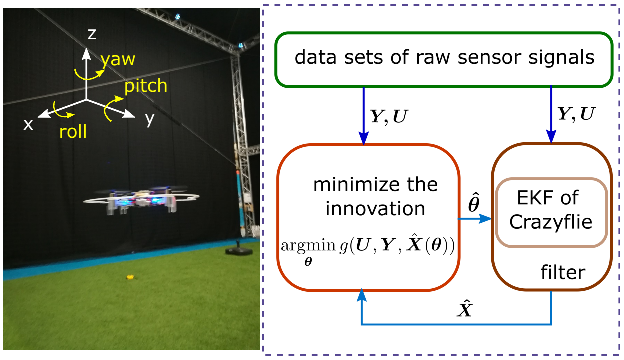
z
yaw
pitch
x
roll
y
data sets of raw sensor signals
$\boldsymbol{Y}, \boldsymbol{U}$
$\boldsymbol{Y}, \boldsymbol{U}$
minimize the innovation
$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, g(\boldsymbol{U}, \boldsymbol{Y}, \hat{\boldsymbol{X}}(\boldsymbol{\theta}))$
$\hat{\boldsymbol{\theta}}$
EKF of Crazyflie
filter
$\hat{\boldsymbol{X}}$

Fig. 1. Overview of the unsupervised filter parameter tuning method. **Left:** experimental setup where a small Crazyflie quadrotor is flown manually to get an onboard data set of observations $\boldsymbol{Y}$ and control inputs $\boldsymbol{U}$, while the OptiTrack motion tracking system provides ground-truth for validation. **Right:** the unsupervised tuning process where the filter operates with parameters $\hat{\boldsymbol{\theta}}$, while the estimated states $\hat{\boldsymbol{X}}$ are utilized by the optimization block until the innovation based goal function is minimal.

I. Introduction

PERCEPTION heavily relies on the difference between predictions and sensory observations [1]. For robots, perception is typically implemented in the form of a filter that estimates the states based on an optimal combination of predictions and observations. For example, aerial robots combine a model of their dynamics with gyroscope and accelerometer observations for attitude estimation [2] and optical flow for motion estimation [3]. For the performance of state estimation filters such as the Kalman filter, it is essential that the parameters representing the covariance of both the prediction and observation model are set correctly. These unknown parameters are mostly determined by the human designer, with the help of expensive external measurement setups that give “ground-truth” measurements corresponding to the observations. Determining a model’s parameters in this manner is time-consuming, while the parameters are in principle only valid for the single robot with which the measurements were made. This makes robotic perception less autonomous than animal and human perception. Automatic tuning of filter parameters is an important challenge for achieving intelligent perception of robots.

A less well-known use of the difference between prediction and observation, termed “innovation” in the filtering literature, is to learn the filter’s parameters. Adaptive filters have been proposed to identify the noise covariance parameters based on algebraic methods, such as correlation [4], Bayesian [5], covariance matching [6], [7], and maximum likelihood [8], [9]. The advantage of these methods is that theoretical guarantees can be given, when the assumptions of the filter are correct. Initially, mathematical derivations were made for “simple” systems, e.g., linear systems with a single output [10], [11]. Later work extended these methods to more complex, non-linear systems [12], even giving guarantees on upper bounds of the covariance [13]. However, the above methods require a particular filter formula that is amenable to mathematical derivation, in order to calculate the covariance or likelihood function. This leads to a loss of generality to arbitrary filters or systems.

To reduce the dependency on the mathematical structure of the filter, more recently tuning methods have been proposed that can learn filter parameters based on input and output data of the filter, treating the filter as a black box. Many of these methods require ground-truth measurements of states which are not available to the robot itself. Examples include a deep learning method proposed for covariance estimation [14], and another filter tuning method represented by an optimization problem [15]. Other methods assume knowledge of some of the noise characteristics. For example, in [16], an optimization based EKF estimated process noise without prior knowledge, while the measurement noise was assumed to be known. Finally, there are methods that can function without prior knowledge or ground-truth measurements. A self-tuning mixture model was proposed in [17], which not only tuned the distribution parameters but also estimated the states without

All authors are with Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands (e-mail: s.li-6@tudelft.nl; c.deWagter@tudelft.nl; g.c.h.e.decroon@tudelft.nl)

Digital Object Identifier (DOI): 10.1109/LRA.2019.2930480.

any prior knowledge. In [18], several criteria for parameter learning of Kalman filter were proposed, and the selected one maximized the measurement likelihood. A tuning method for an unscented Kalman filter (UKF) was proposed based on optimization method without ground-truth to maximize the likelihood of measurements [19]. Until now, these studies have only considered the measurement likelihood rather than the whole sensor likelihood meaning that the filter inputs are not considered for determining the process noise. And since most of these studies focus only on simulation, their accuracy for real robots still remains largely unknown.

In this work, an unsupervised tuning method (illustrated in Fig. 1) is proposed for determining filter parameters. It only requires the information available to the robot, without a need for ground-truth measurements. The method is independent of a filter's structure such that it allows for different filters and different sensor arrangements. The use of control inputs can make process noise identification more accurate since acceleration sometimes is modelled as a control input to the filter. The contributions of this work consist of: 1) automatic filter parameter tuning without relying on ground-truth; 2) novel intuitive optimization with generality to any filter; 3) scalability to more sensor inputs and measurements, and the inclusion of control inputs; and 4) experimental results showing that the approach can improve upon the state estimation of a commercial flying robot.

The rest of this paper is organized as follows. Section 2 introduces the model preliminaries and the tuning problem. In Section 3 the optimization-based unsupervised tuning method is presented. Section 4 illustrates the effectiveness of the proposed filter tuning method using several tests. Section 5 presents our conclusions.

II. PRELIMINARIES

To introduce the tuning problem of filter parameters, a generalized nonlinear model of robots will be given in this section. This model shows the prediction and measurement process of state estimation, followed by a specific EKF filter implemented on a quadrotor as a study case, with its unknown noise covariance parameters.

A. Nonlinear Stochastic Modeling

Consider a robot with a nonlinear discrete model described by following equations:

$$\begin{aligned} \boldsymbol{x}_k &= f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{q}\,, \\ \boldsymbol{y}_k &= h(\boldsymbol{x}_k) + \boldsymbol{r} \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$, $\boldsymbol{q} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^p$ and $\boldsymbol{r} \in \mathbb{R}^p$ denote states, inputs, process noise, measurements and measurement noise, respectively. $f(\cdot)$ and $h(\cdot)$ are transition function from time step $k-1$ to k and observation function. Assume Gaussian distributions with zero means for the process and measurement noise and corresponding covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{R} \in \mathbb{R}^{p\times p}$, such that $\boldsymbol{y}_k \sim \mathcal{N}(h(\boldsymbol{x}_k), \boldsymbol{R})$ and $\boldsymbol{x}_k \sim \mathcal{N}(f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}), \boldsymbol{Q})$.

Based on the above assumptions, various filters are designed to estimate robot states represented by $\hat{\boldsymbol{x}}_k$. The tuning method of this paper will only rely on inputs $\boldsymbol{u}_k$, observations $\boldsymbol{y}_k$ and estimate states $\hat{\boldsymbol{x}}_k$ from an arbitrary filter. As EKF is arguably the most widely used estimator for the nonlinear systems, it is chosen for carrying out the filter parameter tuning in this paper. The specific definition of the commonly used EKF can be found in [15].

B. Problem Formulation

The model and EKF filter for state estimation of a Crazyflie commercial quadrotor, are in line with the general model Eq. (1) and the normal EKF in [15]. As the specific equations are given in [20], this paper will not repeat them except for some necessary variables.

In the EKF of Crazyflie, the state vector is $\boldsymbol{x} = [\boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\delta}]$, where $\boldsymbol{\xi} = [x, y, z] \in \mathbb{R}^3$ denotes the three dimensional position of the quadrotor in earth coordinate, $\boldsymbol{\rho} = [vx, vy, vz] \in \mathbb{R}^3$ is the 3-axis velocity in body coordinates, and $\boldsymbol{\delta} \in \mathbb{R}^3$ represents the attitude error vector which is used to update the three attitude angles of pitch ϕ, roll θ and yaw ψ. For simplicity, following analysis will consider $\boldsymbol{\delta}$ as attitude.

Unlike theoretical physical models, most practical aerial robots utilize sensor signals as *control* inputs into the filter. For example, the accelerometer readings are inserted into the filter as controlled accelerations, while the gyro readings are represented as controlled rotation rates. Therefore, the process noise stems mainly from these sensors rather than from the uncertainty in the controlled motions. Suppose all elements of noise are stochastically uncorrelated. Then the unknown noise covariance parameters $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be formulated as diagonal matrices represented by $\boldsymbol{Q} = \mathrm{diag}[q_1\ q_2\ ...\ q_n]$ and $\boldsymbol{R} = \mathrm{diag}[r_1\ r_2\ ...\ r_p]$.

The unknown process noise covariance in the Crazyflie EKF is composed of 9 elements, in which $q_1 = q_2 = \sigma_{axy}t^2/2$, $q_3 = \sigma_{az}t^2/2$, $q_4 = q_5 = \sigma_{axy}t$, $q_6 = \sigma_{az}t$, $q_7 = q_8 = \sigma_{gxy}t$ and $q_9 = \sigma_{gz}t$, where σ_{axy} and σ_{az} denote the standard deviation of horizontal and vertical acceleration noise (m/s^2) in body coordinates. σ_{gxy} and σ_{gz} are standard deviation of gyroscope noise (rad/s) around x, y, and z axis shown in Fig. 1, respectively .

The measurement noise covariance matrix is defined by $\boldsymbol{R} = \mathrm{diag}[\sigma_{fxy}\ \sigma_{lz}]$, in which σ_{fxy} and σ_{lz} represents the standard deviation of the 2-axis optical flow noise in pixels and in which σ_{lz} represents the range noise (m) from a tiny laser pointing to ground. Therefore, the final filter parameters of quadrotor to be determined are given as follows:

$$\boldsymbol{\theta} = [\sigma_{axy}\ \sigma_{az}\ \sigma_{gxy}\ \sigma_{gz}\ \sigma_{fxy}\ \sigma_{lz}\ k_{va}], \tag{2}$$

where k_{va} is an auxiliary parameter to calculate acceleration with velocity, of which the role will be discussed later. Obviously, the parameters in $\boldsymbol{\theta}$ play an important role in estimation performance of EKF. In the next section, an optimization method will be designed to calculate the unknown parameters $\boldsymbol{\theta}$ in Eq. (2).

III. METHOD

This section will approach the filter parameter tuning as an optimization problem. First, a novel scalar goal function

$g(\cdot)$ is proposed to evaluate the filter performance. Then the tuning process would consist in iterations of minimizing the scalar goal function, realized by a stochastic gradient descent method.

A. Goal Function

The literature on adaptive Kalman filtering typically sets the goal function with innovation to maximize the likelihood of measurements as follows.

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \| h(\hat{\boldsymbol{x}}_i) - \boldsymbol{y}_i \|_2. \tag{3}$$

This function represents the average of measurement error norm.

This paper extends this traditional measurement based goal function in (3) by also maximizing the likelihood of input signals, such as in this case $\boldsymbol{u} = [ax \;\; ay \;\; az \;\; gx \;\; gy \;\; gz]$ including 3-axis acceleration and 3-axis gyroscope, which are not directly represented by states. To this end, we use the physical relationship between the filter states and inputs, i.e., in this case the approximated linear relation between velocity and drag acceleration denoted by $[ax \;\; ay] = -k_{va}[vx \;\; vy]$ in wind still conditions. Thus, one can obtain $\boldsymbol{u} = \mathcal{F}(\boldsymbol{x}) = [-k_{va}vx \;\; - k_{va}vy \;\; \dot{vx} \;\; \dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}]$. Incorporating this ”input innovation” into the goal function Eq. (3), the overall goal function is given by

$$g(\boldsymbol{U}, \boldsymbol{Y}, \hat{\boldsymbol{X}}(\boldsymbol{\theta})) = \frac{1}{N} \sum_{i=1}^{N} (\| h(\hat{\boldsymbol{x}}_i) - \boldsymbol{y}_i \|_1 + \| \mathcal{F}(\hat{\boldsymbol{x}}_i) - \boldsymbol{u}_i \|_1) \tag{4}$$

$$\hat{\boldsymbol{\theta}}_{\text{best}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \, g(\boldsymbol{U}, \boldsymbol{Y}, \hat{\boldsymbol{X}}(\boldsymbol{\theta})) \tag{5}$$

Where data sets $\boldsymbol{Y}$, $\boldsymbol{U}$ and $\hat{\boldsymbol{X}}$ are length N sequences of observations $\boldsymbol{y}$, control inputs $\boldsymbol{u}$ and estimated states $\hat{\boldsymbol{x}}$, respectively. Here, we select the L1 loss function as it does not overpenalize large but unlikely errors, and is therefore more robust to non-Gaussian distributions with outliers [21], which are typical for many real-world sensors.

In real implementation, expectations $h(\hat{\boldsymbol{x}}_i)$ and $\mathcal{F}(\hat{\boldsymbol{x}}_i)$ tend to be smooth with $\hat{\boldsymbol{\theta}}_{\text{best}}$. Therefore, noisy signals are smoothed generally to keep the goal function more relevant to filter results. Also, since the laser measurement is considerably precise and it is coupled with the optical flow innovation, the laser cost function is weighted as $50|h_{lz}(\hat{\boldsymbol{x}}_i) - y_{lz}|$ in Eq. (4), in which we have left out the weights for brevity.

B. Tuning Process

The entire unsupervised tuning process is shown in Fig. 1. It starts with an initial parameters $\boldsymbol{\theta}_0$ in which all elements are set to zero. The filter runs once with the initial parameter and outputs the sequence of estimated states $\hat{\boldsymbol{X}}$. Then the optimization block utilizes $\hat{\boldsymbol{X}}$ to infer all sensor data, and minimizes the difference between inferred and real sensor signals for tuning $\hat{\boldsymbol{\theta}}$. This tuning step iterates until a minimal value of the goal function is obtained.

Given the goal function, this paper deploys stochastic gradient descent as the optimization method. Since our proposed method regards the goal function as a black box problem, it would be possible to use any other metaheuristic optimization method.

The incorporated random process can help jump out of local minima to some extent. Because after one gradient descent, the method will select a fixed number of random parameter initalizations for comparison to see if there exists a parameter set that has a smaller goal value. A more systematic description of this tuning method is given in the following algorithm. $\boldsymbol{\theta}_{lb}, \boldsymbol{\theta}_{ub}, N_s, fm()$, and $random()$ denote the lower bound, upper bound, number of searches that improve the current estimation result, fmincon() function and random() function in matlab, respectively.

Algorithm 1 Unsupervised tuning of EKF

```
procedure $f_{\text{opti}}(\boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{lb}}, \boldsymbol{\theta}_{\text{ub}}, N_{\text{s}})$    ▷ Tuning function
    $\hat{\boldsymbol{\theta}}_{\text{opti}} \leftarrow fm(g(\boldsymbol{u}, \boldsymbol{y}, \hat{\boldsymbol{x}}(\boldsymbol{\theta})), \boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{lb}}, \boldsymbol{\theta}_{\text{ub}})$    ▷ gradient descent function e.g. fmincon in matlab
    $g_{\text{opti}} \leftarrow g(\boldsymbol{u}, \boldsymbol{y}, \hat{\boldsymbol{x}}(\hat{\boldsymbol{\theta}}_{\text{opti}}))$
    $n_{\text{loop}} \leftarrow 0$
    while $n_{\text{loop}} < N_{\text{s}}$ do
        $\hat{\boldsymbol{\theta}}_{\text{new}} \leftarrow \boldsymbol{\theta}_{\text{ub}} \times random(length(\boldsymbol{\theta}), 1)$
        if $g(\boldsymbol{u}, \boldsymbol{y}, \hat{\boldsymbol{x}}(\hat{\boldsymbol{\theta}}_{\text{new}})) \leq g_{\text{opti}}$ then
            $\hat{\boldsymbol{\theta}}_{\text{opti}} \leftarrow fm(g(\boldsymbol{u}, \boldsymbol{y}, \hat{\boldsymbol{x}}(\boldsymbol{\theta})), \boldsymbol{\theta}_{\text{new}}, \boldsymbol{\theta}_{\text{lb}}, \boldsymbol{\theta}_{\text{ub}})$
            $g_{\text{opti}} \leftarrow g(\boldsymbol{u}, \boldsymbol{y}, \hat{\boldsymbol{x}}(\hat{\boldsymbol{\theta}}_{\text{opti}}))$
            $n_{\text{loop}} \leftarrow 0$
        else
            $n_{\text{loop}} \leftarrow n_{\text{loop}} + 1$
    $\hat{\boldsymbol{\theta}}_{\text{best}} \leftarrow \hat{\boldsymbol{\theta}}_{\text{opti}}$    ▷ return the best parameter
```

C. Hardware Setup

To validate the effectiveness of the proposed tuning method for EKF, experiments are conducted on the commercial quadrotor Crazyflie 2. It uses a MPU-9250 IMU consisting of a 3-axis gyroscope and a 3-axis accelerometer. These are considered as inputs of the onboard EKF. Also, VL53L0x and PMW3901MB sensors provide the range to ground and optical flow measurements, respectively. These sensors are fused by the onboard EKF with preset parameters from the factory.

The tuning method relies on one manual flight of this vehicle. Maneuvering the drone in all dimensions is key to collecting a suitable sensor data set. Data will be stored on a SD card, which is further used by an offline computer to calculate all required filter parameters. This process takes several minutes for laptop with i7-6600U CPU at 3.40 GHz. In principle, a slower version of this optimization process could take place onboard the Crazyflie when not in flight.

To validate filter performance, a motion capture system OptiTrack is used for providing external measured ground-truth of 3-axis attitude and 3-axis position.

IV. Experimental Results

The performance of the proposed unsupervised tuning method is illustrated by several experimental scenarios shown in this section, in order to validate its tuning results, estimation performance, robustness to extra noise and efficiency for expanding filters. The data sets $\boldsymbol{Y}$ and $\boldsymbol{U}$ for tuning are collected from 20 manual flights of the quadrotor operated in all dimensions, whereas half data sets are used for tuning and the other half are used for testing. The duration of these data sets ranges from 5 to 15 seconds with 1000Hz sampled frequency.

A. Results of unsupervised tuning

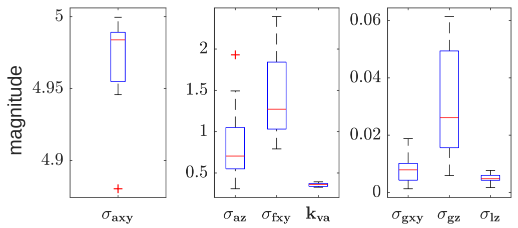

Fig. 2. Distribution of filter parameters tuned based on 10 training data sets, in which corresponding units are $\sigma_{axy}(m/s^2)$, $\sigma_{az}(m/s^2)$, $\sigma_{gxy}(rad/s)$, $\sigma_{gz}(rad/s)$, $\sigma_{fxy}(pixel)$, and $\sigma_{lz}(m)$.

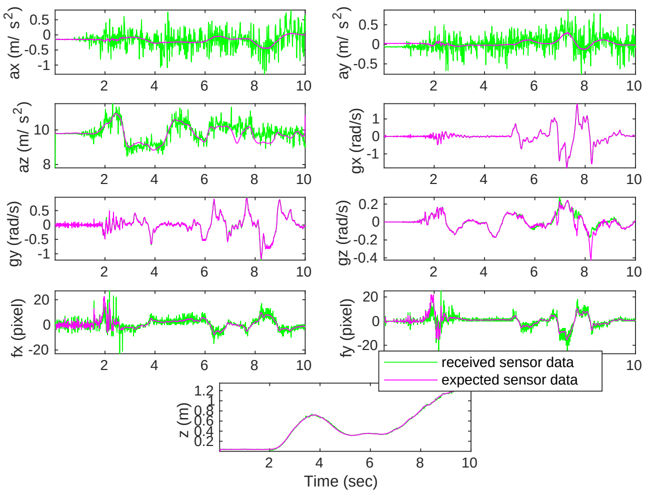

Fig. 3. Expected (magenta) and received (green) sensor data for all sensors including acceleration of ax, ay and az, gyroscope of gx, gy and gz, flow of fx and fy, and altitude range z, based on a training data set.

By implementing Algorithm 1 on sensor data collected from a real Crazyflie quadrotor, all related noise parameters are calculated and given as $\hat{\boldsymbol{\theta}}_{\text{best}}$. Table I gives insight into the optimized parameter values, by showing the median values over the 10 different data sets. The distribution of $\hat{\boldsymbol{\theta}}_{\text{best}}$ tuned from 10 training data sets is shown in Fig. 2 with 25th and 75th percentiles as the boundaries and the few outliers can be neglected. σ_{axy} tends to be far from the preset value in the Crazyflie simply because horizontal acceleration only works for the onboard EKF when the quadrotor is in freefall or carried.

At the same time, expected and real sensor data are shown in Fig. 3 to show the effectiveness of the goal function on a training data set. From the figure, we can see that all dimensional raw sensor data are well inferred by utilizing this tuning method.

B. Filter results with tuned parameters

In this subsection, we compare the performance of the tuned parameter set with that of the onboard EKF with preset parameters. Estimation performance on position, velocity and attitude using the filter with $\hat{\boldsymbol{\theta}}_{\text{best}}$ is shown in the Fig. 4. Instead of taking a specific parameter set, we here take the median values as shown in Table I for the tuned filter.

Fig. 4 indicates that both the EKF with tuned parameters and onboard EKF with original parameters have accurate position estimation. The tuned EKF tends to be closer to the ground-truth at various times. We can also see the tuned filter has less attitude estimation error than that with original parameters within some time. Filters with both sets of parameters can estimate the velocity with similar precision as seen from Fig. 4. Traditional tuning only with measurement likelihood has larger error than our method, especially at x, y and yaw axes due to the lack of the input likelihood. The corresponding tuned parameter $\hat{\boldsymbol{\theta}}_{\text{noInput}}$ in Table I also shows its wrong estimation of the yaw covariance and the drag coefficient.

Fig. 5 summarizes the comparison between the automatically tuned filter parameters and the preset parameters, showing the error distribution of all estimated states on 10 test data sets. Root mean square error (RMSE) is used in this figure. The state estimates of the automatically tuned filter attain a lower RMSE for all states.

C. Tuning results with extra noise and expanded model

This section continues analyzing the effectiveness of the method with regards to extra noise and expanded filter structure on a training data set.

The first scenario is adding extra noise to some dimensions of the data sets. For example, a white Gaussian noise with 5dB is added to the normally accurate altitude measurement, which is created by AWGN function in MATLAB. Fig. 6 shows that the EKF with previous tuned parameter has a larger vertical velocity and position estimation error due to the extra noise. After retuning the filter with the extra noise, the new parameters enable the filter to largely reduce the estimation error. This can also be seen from Table I; the standard deviation of the laser measurement increases with a factor 10 due to the extra noise. Moreover, σ_{az} slightly decreases so that the vertical velocity relies more on the accelerometer readings. Also, Fig. 7 illustrates that the retuned EKF has less estimation error in all states compared to that with unchanged parameters.

Another test is constructed for an expanded filter model, which uses thrust from command rather than from accelerometer to predict velocity. Because the latter one is sometimes noisy due to mechanical vibrations. The thrust command is smooth but mapping it to thrust requires knowledge of mass which varies significantly between drones. In fact, the Crazyflie filter has a term for the thrust, but this is commented out in the code mentioning this mass variability problem. The proposed self-tuning method can simply learn the thrust function for the specific drone by adding a mapping coefficient k_{Ta} into $\hat{\boldsymbol{\theta}}$. After training, the unsupervised tuning method gives an exact $k_{Ta} = 1.4654$ that maps thrust command to

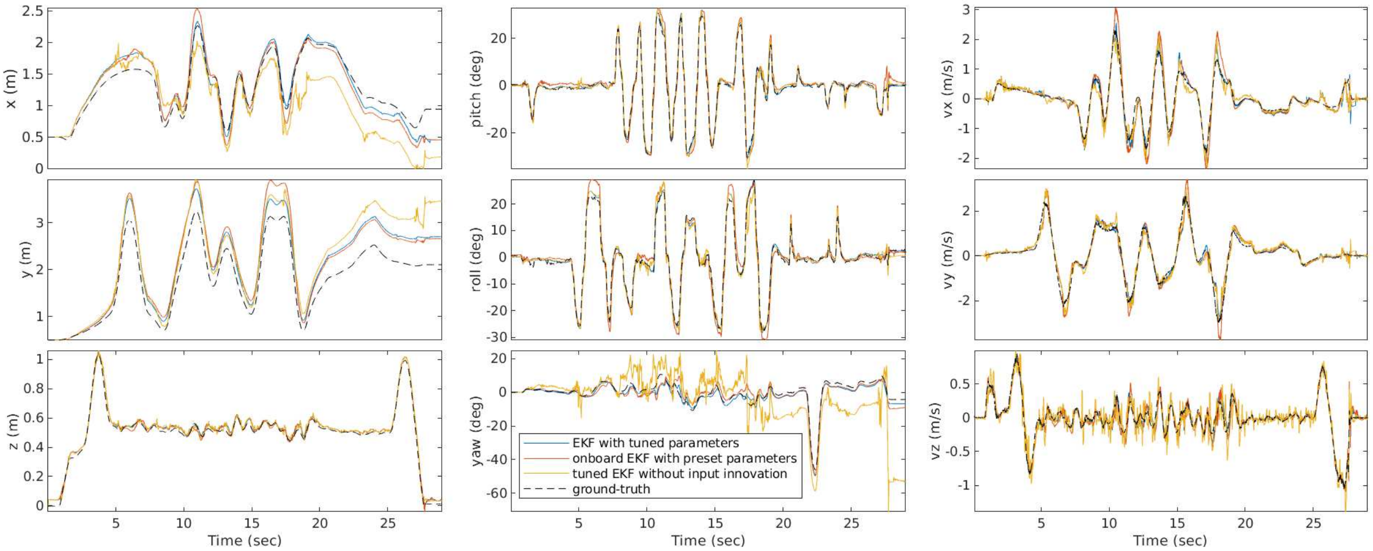

Fig. 4. The 3-dimensional position, attitude and velocity estimated by EKF with tuned parameters (blue), EKF with Crazyflie parameters (red), tuned EKF without the input innovation term (yellow), as well as ground-truth from OptiTrack (black), respectively, based on a test data set.

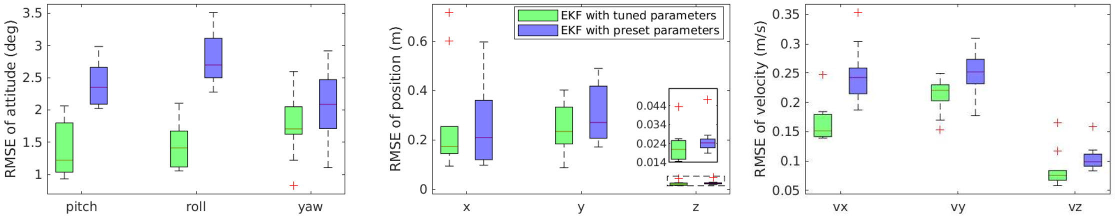

Fig. 5. Comparison between the self-tuned filter and the standard Crazyflie filter based on 10 test data sets.

TABLE I

EFK PARAMETERS FROM CRAZYFLIE, MEDIAN OF 10 TUNED TESTS, TUNING WITHOUT INPUT INNOVATION, TUNING WITHOUT EXTRA NOISE, TUNING WITH EXTRA NOISE, TUNING WITH THRUST MODEL, RESPECTIVELY.

$\boldsymbol{\theta}$	σ_{axy}	σ_{az}	σ_{gxy}	σ_{gz}	σ_{fxy}	σ_{lz}	k_{va}	k_{Ta}
$\boldsymbol{\theta}_{\mathrm{CF}}$	0.5000	1.0000	0.1000	0.1000	0.2500	0.0025		
$\hat{\boldsymbol{\theta}}_{\mathrm{median}}$	4.9839	0.7048	0.0079	0.0261	1.2720	0.0048	0.3621	
$\hat{\boldsymbol{\theta}}_{\mathrm{noInput}}$	3.4744	2.5060	0.0723	1.8408	0.8837	0.0013	2.4981	
$\hat{\boldsymbol{\theta}}_{\mathrm{noNoise}}$	4.9832	0.7513	0.0056	0.0500	2.5656	0.0095	0.3193	
$\hat{\boldsymbol{\theta}}_{\mathrm{noise}}$	4.1874	0.4380	0.0069	0.0461	2.6429	0.0926	0.3514	
$\hat{\boldsymbol{\theta}}_{\mathrm{thrust}}$	4.5725	0.7404	0.0157	0.0897	3.7352	0.0300	0.3233	1.4654

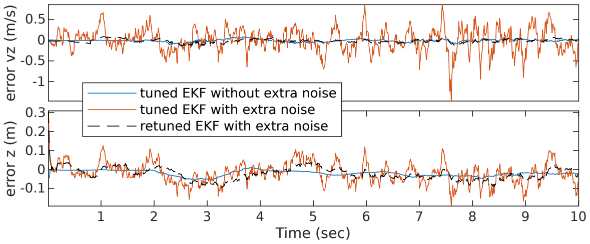

Fig. 6. Altitude and corresponding velocity estimation error with respect to extra noise on laser range measurement on a training data set.

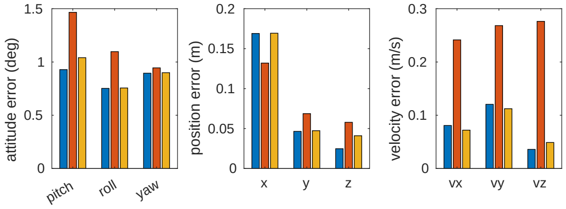

Fig. 7. Estimation error on all states of the EKF with tuned parameter without noise (blue), previous tuned parameter with noise (red), and retuned parameter with noise (yellow), respectively, on a training data set.

acceleration as $az = k_{Ta} * T_{cmd}/65536$ shown in Fig. 8. We use this thrust model only when it is flying because the ground also provides a force to the robot when it is not flying between 0s-2s in Fig. 8. Also, this figure shows that even if the input is replaced by the thrust model, the vertical position estimation error is still comparable.

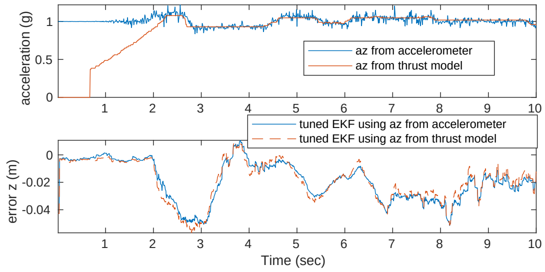

Fig. 8. Tuning of a filter with the trust model on a training data set.

D. Tuning of a UKF on the EuroC MAV dataset

The method is further tested on the commonly used UKF formulated in [19] for attitude estimation, and the IMU data is from the public EuroC MAV dataset. The unknown 3-axis gyroscope covariance $\boldsymbol{Q}$ and 2-axis accelerometer covariance $\boldsymbol{R}$ are tuned from initial [1,1,1,1,1] to [5.8e-8,2.8e-7,0.6443,2.1987,2.5532]. Results from the Fig. 9 show that the tuned UKF can estimate the attitude accurately, which validates the efficacy of the proposed method on different filters and data sets.

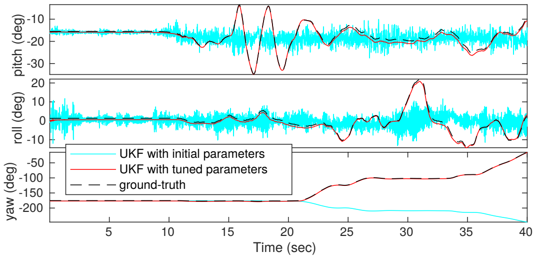

Fig. 9. Attitude estimation of UKF with initial parameters, tuned parameters, and ground-truth on the public EuroC MAV dataset.

V. Conclusions

We have proposed an unsupervised parameter tuning method for arbitrary filters by minimizing the difference between predictions and observations. The effectiveness of the proposed method has been validated using real-world data of a tiny Crazyflie quadrotor. The results show that the filter with self-tuned parameters has more precise estimation than the filter with preset parameters. Moreover, we have shown that this tuning method can reject extra noise, by adding noise to the sensor data, and it can also easily extend to an expanded filter model, identifying new unknown parameters. Finally, the method is validated to work on other filters like a UKF.

However, there are some limitations of this method. For example, in the current article we have based the input innovation term on our knowledge of the physical system. Future work could focus on learning the relation between states and control inputs if this relationship is unknown. Of course, if there is no inherent relationship between the state and the control inputs, then the innovation term will not be beneficial. Furthermore, the performance may degrade if some sensors have a covariance that changes heavily over time. Future work could address issues such as largely varied covariance and automatic selection of the innovation weight for the laser.

References

[1] H. E. Den Ouden, P. Kok, and F. P. De Lange, "How prediction errors shape perception, attention, and motivation," *Frontiers in psychology*, vol. 3, p. 548, 2012.

[2] M. Karásek, F. T. Muijres, C. De Wagter, B. D. Remes, and G. C. de Croon, "A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns," *Science*, vol. 361, no. 6407, pp. 1089–1094, 2018.

[3] A. Santamaria-Navarro, J. Sola, and J. Andrade-Cetto, "High-frequency mav state estimation using low-cost inertial and optical flow measurement units," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1864–1871.

[4] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic press, 1982, vol. 2.

[5] H. Hu and G. Kantor, "Introspective evaluation of perception performance for parameter tuning without ground truth," *Proceedings of Robotics: Science and Systems, Boston, USA*, 2017.

[6] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 520–523, 1976.

[7] B. J. Odelson, A. Lutz, and J. B. Rawlings, "The autocovariance least-squares method for estimating covariances: application to model-based control of chemical reactors," *IEEE transactions on control systems technology*, vol. 14, no. 3, pp. 532–540, 2006.

[8] R. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on automatic control*, vol. 17, no. 5, pp. 693–698, 1972.

[9] A. Mohamed and K. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of geodesy*, vol. 73, no. 4, pp. 193–203, 1999.

[10] R. Reynolds, "Robust estimation of covariance matrices," *IEEE Transactions on Automatic Control*, vol. 35, no. 9, pp. 1047–1051, 1990.

[11] P. R. Bélanger, "Estimation of noise covariance matrices for a linear time-varying stochastic process," *Automatica*, vol. 10, no. 3, pp. 267–275, 1974.

[12] F. V. Lima, M. R. Rajamani, T. A. Soderstrom, and J. B. Rawlings, "Covariance and state estimation of weakly observable systems: Application to polymerization processes," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1249–1257, 2013.

[13] J. Duník, O. Straka, and M. Šimandl, "On autocovariance least-squares method for noise covariance matrices estimation," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 967–972, 2017.

[14] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1436–1443.

[15] T. D. Powell, "Automated tuning of an extended kalman filter using the downhill simplex algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, pp. 901–908, 2002.

[16] M. Karasalo and X. Hu, "An optimization approach to adaptive kalman filtering," *Automatica*, vol. 47, no. 8, pp. 1785–1793, 2011.

[17] T. Pfeifer and P. Protzel, "Robust sensor fusion with self-tuning mixture models," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3678–3685.

[18] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative training of kalman filters." in *Robotics: Science and systems*, vol. 2, 2005, p. 1.

[19] L. A. Scardua and J. J. Da Cruz, "Complete offline tuning of the unscented kalman filter," *Automatica*, vol. 80, pp. 54–61, 2017.

[20] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1730–1736.

[21] C. Robert, *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.