A Computationally Efficient Moving Horizon Estimator for Ultra-Wideband Localization on Small Quadrotors

Pfeiffer, Sven; Wagter, Christophe De; Croon, Guido C.H.E.De

DOI
10.1109/LRA.2021.3095519

Publication date
2021

Document Version
Final published version

Published in
IEEE Robotics and Automation Letters

A Computationally Efficient Moving Horizon Estimator for Ultra-Wideband Localization on Small Quadrotors

Sven Pfeiffer, Christophe de Wagter, and Guido C.H.E. de Croon, *Member, IEEE*

***Abstract*—We present a computationally efficient moving horizon estimator that allows for real-time localization using Ultra-Wideband measurements on small quadrotors. The estimator uses only a single iteration of a simple gradient descent method to optimize the state estimate based on past measurements, while using random sample consensus to reject outliers. We compare our algorithm to a state-of-the-art Extended Kalman Filter and show its advantages when dealing with heavy-tailed noise, which is frequently encountered in Ultra-Wideband ranging. Furthermore, we analyze the algorithm's performance when reducing the number of beacons for measurements and we implement the code on a 30 g Crazyflie drone, to show its ability to run on computationally limited devices.**

***Index Terms*—Sensor fusion, localization, aerial systems: perception and autonomy, optimization and optimal control.**

I. Introduction

OVER the past few years, the continuing miniaturization of computational hardware has advanced the development of small, agile quadrotors. This new generation of flying robots performs tasks in environments, that were previously inaccessible to them due to size and safety constraints. Most notably, this concerns indoor environments, such as warehouses, greenhouses or industrial facilities, which present numerous new potential use-cases.

A key component required for autonomous operation in these environments is a robust and accurate localization system. Ultra-Wideband (UWB) technology has been gaining a lot of attention for indoor positioning, as it offers high accuracy at an affordable cost. UWB systems allow for the localization of a mobile tag, by exchanging messages with a set of static beacons at known locations in the environment. Although a position can be calculated from a sufficient number of independent UWB measurements through multilateration, better results can be achieved when using additional filtering. Commonly, a state estimator that can fuse the UWB measurements with data collected by the on-board inertial measurement unit (IMU) is employed.

Manuscript received February 23, 2021; accepted June 17, 2021. Date of publication July 8, 2021; date of current version July 22, 2021. This letter was recommended for publication by Associate Editor J. Guivant and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by Royal Brinkman. *(Corresponding author: Sven Pfeiffer.)*

The authors are with the Faculty of Aerospace Engineering, Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: s.u.pfeiffer@tudelft.nl; c.dewagter@tudelft.nl; guido.de.croon@gmail.com).

Digital Object Identifier 10.1109/LRA.2021.3095519

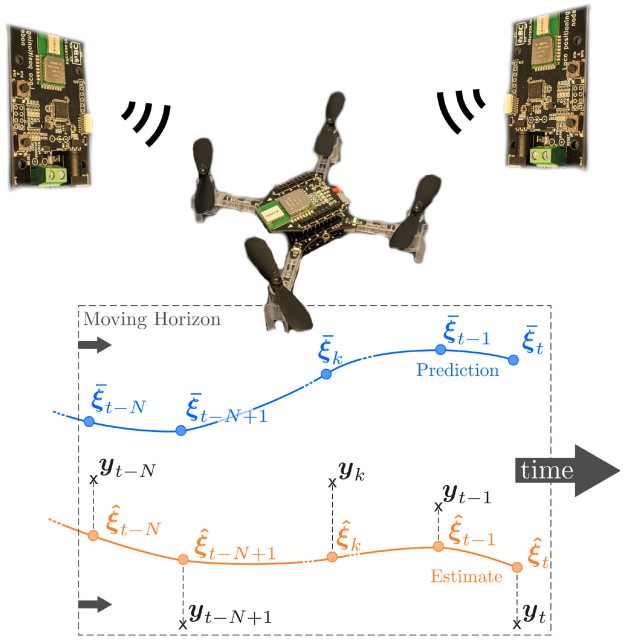

Fig. 1. Moving horizon estimators can optimize the evolution of a drone's trajectory based on a limited number of previous UWB measurements. We propose a novel formulation for UWB localization that is efficient enough to run on a 30 g Crazyflie drone and performs robustly in scenarios with very noisy or limited measurements. In the plot, $\boldsymbol{\xi}$ stands for the state and $\boldsymbol{y}$ for a measurement of the state.

For linear systems with Gaussian noise, the Kalman Filter yields an optimal estimate in the sense that it minimizes a weighted two-norm of the expected value of the estimation error [1]. Different formulations have been developed to approximate this result for more general systems, while retaining the computationally efficient, iterative structure of the Kalman Filter. The Extended Kalman Filter (EKF) [1] or Unscented Kalman Filter (UKF) [2] are often used to address non-linearities, but fail to address the issues that can arise from uncertain system models and unknown or non-Gaussian noise. Since UWB measurements often exhibit heavy-tailed noise due to multi-pathing and non-line-of-sight (NLOS) conditions [3], [4], this can be a significant shortcoming.

For highly non-linear problems and systems with non-Gaussian noise, Moving Horizon Estimation (MHE) yields better results at the cost of requiring more computational power.

MHE aims at finding a sequence of states, that minimizes the noise and disturbances required to explain the observed measurements over a moving time horizon [5]. A multitude of options exist for the formulation of the cost-function to this minimization problem, but most commonly, a batch least-squares formulation is used [6]. In addition, past data can be included in the cost-function through an "arrival cost". Unfortunately, the requirement of solving an optimization problem at every time step currently make MHE impractical for real-time applications on computationally limited devices or for systems with fast dynamics.

In this work, we present a computationally efficient Moving Horizon Estimator with RANSAC [7] outlier rejection that can perform real-time localization with UWB on a small quadrotor with limited computational power. We compare it with a state-of-the-art EKF in simulated noise environments and show its advantages when dealing with heavy-tailed noise, which is observed on UWB measurements when multi-pathing and NLOS effects occur. We also show that the MHE is still able to perform state estimation when the number of UWB beacons in the environment is drastically reduced.

In Section II we will present an overview of the related work. Section III outlines the design of our computationally efficient moving horizon estimator, while Section IV will cover its implementation for position estimation with UWB. We will then compare our MHE with a state-of-the-art EKF in Section V. Finally we will conclude on our findings in Section VI.

II. Related Work

Shortly after the original paper on the Kalman Filter appeared in 1960 [8], the concept of limited memory filters was introduced by Andrew Jazwinski to address cases in which the Kalman Filter diverges due to modelling errors [9]. Several strategies have since been proposed to reduce the high computational cost of such limited memory filters.

The Unbiased Finite Impulse Response (UFIR) Filter was developed with a computationally efficient iterative structure like the Kalman Filter. It offers robustness with respect to unknown noise and initial position, but still relies on linearization when dealing with non-linear system models. Furthermore, the UFIR filter is only unbiased and not optimal [10].

Moving Horizon Estimation (MHE) on the other hand solves an online optimization problem and can therefore accomodate non-linear system models. However, the solution of these optimization problems can quickly become expensive, which has mostly limited the use of MHE to slow processes, e.g., in the chemical industry. There are however approaches to improve the computational complexity of moving horizon estimation. Zavala, Laird and Biegler [11] have used sensitivity analysis to formulate and solve a reference optimization problem based on previous measurements, while waiting for a new measurement to arrive. Once the new measurement is known, the solution to the real problem can be calculated more quickly. Kühl *et al.* [12] have demonstrated a scheme that only relies on a single iteration of a sequential quadratic programming method per time step. Both papers demonstrate performance on examples from the chemical industry, which deal with more states but involve significantly slower dynamics than quadrotors.

For this work, the use of computationally more efficient gradient descent methods was investigated, which where proven to be stable by Alessandri and Gaggero [13]. Similar to Kühl *et al.* they were able to show, that also with simple gradient descent minimization, a single iteration at every time step is sufficient to get a stable state estimate.

On autonomous flying robots, MHE is rarely used due to the before mentioned computational complexity. Girrbach *et al.* have investigated the influence of horizon length when fusing GNSS and IMU with MHE, but only perform comparisons offline and without addressing the challenges of computational cost [14]. Shuo *et al.* formulate the MHE problem with position measurements and a simplified model in such a way, that there is an analytical solution to the optimization problem. This allows them to use their MHE on a 72 g quadrotor in a drone racing context [15].

Instead of using MHE, state estimation with UWB measurements on quadrotors is usually based on variations of the Kalman Filter. The Crazyflie drone which we use as our testing platform in this work for example uses an EKF for fusing data from a wide variety of on-board sensors [16]. As an alternative to Kalman Filters, Xu *et al.* [17], [18] recently started to investigate different forms of UFIR filters for the purpose of fusing UWB and IMU on drones.

In our work, we are approaching the problem of sensor fusion on quadrotors by using a moving horizon estimator, which uses a single iteration gradient descent algorithm to optimize its state-estimate. We will show that this reduces the computational load of MHE sufficiently, while providing accurate position estimates on computationally limited drones.

III. Moving Horizon Estimator

A. Problem Formulation

Let us consider a non-linear, discrete-time dynamic system as described by (1):

$$\begin{cases} \boldsymbol{\xi}_{t+1} = \boldsymbol{f}(\boldsymbol{\xi}_t, \boldsymbol{u}_t) + \boldsymbol{w}_t \\ \boldsymbol{y}_t \quad\;\; = \boldsymbol{h}(\boldsymbol{\xi}_t, \boldsymbol{u}_t) + \boldsymbol{v}_t \end{cases} \tag{1}$$

where $t = 0, 1, \ldots$ is the time step, $\boldsymbol{\xi}_t \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_t \in \mathbb{R}^p$ is the input vector, and $\boldsymbol{y}_t \in \mathbb{R}^m$ is the measurement vector. Finally $\boldsymbol{w}_t \in \mathbb{R}^n$ and $\boldsymbol{v}_t \in \mathbb{R}^m$ are the process and measurement noise vectors, all at time step t respectively.

The classical MHE problem can be formulated, as finding the state $\hat{\boldsymbol{\xi}}_{t-N|t}$, that minimizes the cost function $J(\hat{\boldsymbol{\xi}}_{t-N|t})$, where $\hat{\boldsymbol{\xi}}_{t-N|t}$ is the estimated state at time step $t-N$, based on information available at time step t, N being the length of the moving horizon. The cost function typically includes an arrival cost, which takes into account a prior estimate of the state, $\bar{\boldsymbol{\xi}}_{t-N|t}$, the sum of squares of the process noise and the sum of squares of the measurement noise. In this work, we simplify the problem to an output noise MHE, which neglects the process noise terms. This simplification is necessary to reduce the size of the optimization problem sufficiently for it to be solved on

an embedded system. It removes the ability of the estimator to adjust inputs to better fit the measurements and is therefore likely to reduce the accuracy of the estimate. Since it does however not impact the convergence of the estimate, this can be an acceptable trade-off for many systems.

Since the measurement frequency of UWB varies unpredictably, some measurements in $\boldsymbol{y}_k$ might be outdated. We can accommodate for this fact, by writing the cost function in terms of individual measurements y_i, $i = 0, \ldots, m$, and including the binary switching sequences $\theta_{i,k}$, which are equal to 1 if a new measurement arrives and 0 otherwise [19].

$$\theta_{i,k} = \begin{cases} 1 \text{ if } y_{i,k} \neq y_{i,k-1} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

To express the cost function in term of the state at the beginning of the horizon, $\hat{\boldsymbol{\xi}}_{t-N|t}$, it is helpful to define the composite prediction and measurement equations at time step k:

$$\boldsymbol{\mathcal{F}}_k(\bar{\boldsymbol{\xi}}_{t-N|t}) = \boldsymbol{f}^{u_k} \circ \ldots \circ \boldsymbol{f}^{u_{t-N}}(\bar{\boldsymbol{\xi}}_{t-N|t}) \tag{3}$$

$$\mathcal{H}_{i,k}(\bar{\boldsymbol{\xi}}_{t-N|t}) = h_i^{u_k} \circ \boldsymbol{\mathcal{F}}_{k-1}(\bar{\boldsymbol{\xi}}_{t-N|t}) \tag{4}$$

For compactness, we omit the input vectors $\boldsymbol{u}_{t-N}$ to $\boldsymbol{u}_k$ in the composite functions, as the output error structure of the MHE does not allow them to change anyway. For the composing functions we note the corresponding input vector in the superscript. This results in the final cost function shown in (5).

$$J_t(\hat{\boldsymbol{\xi}}_{t-N|t}) = \mu \|\hat{\boldsymbol{\xi}}_{t-N|t} - \bar{\boldsymbol{\xi}}_{t-N|t}\|^2 + \sum_{k=t-N}^{t} \sum_{i=1}^{m} \left(\theta_{i,k} \cdot \|y_{i,k} - \mathcal{H}_{i,k}(\hat{\boldsymbol{\xi}}_{t-N|t})\|^2 \right) \tag{5}$$

B. Computationally Efficient MHE

To solve the MHE problem efficiently, we use a single iteration of the gradient method at every time step [13]. The prior estimate $\bar{\boldsymbol{\xi}}_{t-N|t}$ is obtained by applying the prediction equations on the last available state estimate.

$$\hat{\boldsymbol{\xi}}_{t-N|t} = \bar{\boldsymbol{\xi}}_{t-N|t} - \alpha \nabla J_t(\bar{\boldsymbol{\xi}}_{t-N|t}) \tag{6}$$

$$\bar{\boldsymbol{\xi}}_{t-N|t} = \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{t-N-1|t-1}, \boldsymbol{u}_{t-N-1}) \tag{7}$$

If the number of states n becomes large, the numerical calculation of $\nabla J_t(\bar{\boldsymbol{\xi}}_{t-N|t})$ becomes very expensive. This is due to the fact, that every evaluation of $J_t(\bar{\boldsymbol{\xi}}_{t-N|t})$ requires the prediction of the state vector's evolution over the complete horizon. We are therefore using an iterative, analytical solution based on the chain rule, which only requires a single prediction cycle over the full horizon.

In the analytical expression of the gradient of the cost function, the first term (arrival cost) disappears since we evaluate the gradient at $\bar{\boldsymbol{\xi}}_{t-N|t}$. What remains is a sum of terms that stem from the individual measurements:

$$\nabla J_t(\bar{\boldsymbol{\xi}}_{t-N|t}) = -2 \sum_{k=t-N}^{t} \sum_{i=1}^{m} \left[\theta_{i,k} \left(y_{i,k} - \mathcal{H}_{i,k}(\bar{\boldsymbol{\xi}}_{t-N|t}) \right) \nabla \mathcal{H}_{i,k}(\bar{\boldsymbol{\xi}}_{t-N|t}) \right] \tag{8}$$

By stepping through the full horizon, we can calculate the sequence of prior estimates $\bar{\boldsymbol{\xi}}_{k|t}$, $k = t - N, \ldots, t$ in an iterative manner, which makes calculating the predicted measurement $\mathcal{H}_{i,k}(\bar{\boldsymbol{\xi}}_{t-N|t}) = h_i(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k)$ trivial.

To calculate the gradient of $\mathcal{H}_{i,k}$, we make use of the chain rule to split the equation into two factors: The gradient based on the current predicted state and input, $\nabla h_i(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k)$, and the Jacobian of the composite prediction equation $D\boldsymbol{\mathcal{F}}_{k-1}(\bar{\boldsymbol{\xi}}_{t-N|t})$:

$$\nabla \mathcal{H}_{i,k}(\bar{\boldsymbol{\xi}}_{t-N|t}) = \left(D\boldsymbol{\mathcal{F}}_{k-1}(\bar{\boldsymbol{\xi}}_{t-N|t}) \right)^T \cdot \nabla h_i(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k) \tag{9}$$

$$\begin{aligned} D\boldsymbol{\mathcal{F}}_{k-1}(\bar{\boldsymbol{\xi}}_{t-N|t}) = {} & D\mathbf{f}\left(\bar{\boldsymbol{\xi}}_{k-1|t}, \boldsymbol{u}_{k-1}\right) \\ & \cdot D\mathbf{f}\left(\bar{\boldsymbol{\xi}}_{k-2|t}, \boldsymbol{u}_{k-2}\right)) \\ & \cdot \ldots \cdot D\mathbf{f}\left(\bar{\boldsymbol{\xi}}_{t-N|t}, \boldsymbol{u}_{t-N}\right) \end{aligned} \tag{10}$$

Similar to the predicted state, the Jacobians of the prediction equation can be calculated in an iterative manner by premultiplying the previous value with the Jacobian evaluated at the current prediction. In the end, calculating the Jacobian of the cost function therefore requires the following computations for each time step k in the horizon (i.e. N times):

- Forward prediction of the prior estimate

$$\bar{\boldsymbol{\xi}}_{k|t} = \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{k-1|t}, \boldsymbol{u}_{k-1}) \tag{11}$$

- Calculation of the $n \times n$ Jacobian of the prediction function f at $t = k - 1$

$$D\mathbf{f}(\bar{\boldsymbol{\xi}}_{k-1|t}, \boldsymbol{u}_{k-1}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_{k-1|t}, \boldsymbol{u}_{k-1}) \tag{12}$$

- Calculation of the composite Jacobian of $\boldsymbol{\mathcal{F}}_{k-1}$ by multiplication of two $n \times n$ matrices (chain rule)

$$D\boldsymbol{\mathcal{F}}_{k-1}(\bar{\boldsymbol{\xi}}_{t-N|t}) = D\mathbf{f}(\bar{\boldsymbol{\xi}}_{k-1|t}, \boldsymbol{u}_{k-1}) \cdot D\boldsymbol{\mathcal{F}}_{k-2}(\bar{\boldsymbol{\xi}}_{t-N|t}) \tag{13}$$

- Up to m measurement predictions for measurements that arrived at $t = k$,

$$h_i(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k) \tag{14}$$

- Calculation of up to m measurement gradients for measurements that arrived at $t = k$,

$$\nabla h_i(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k) = \frac{\partial h_i}{\partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_{k|t}, \boldsymbol{u}_k) \tag{15}$$

- Calculation of up to m composite measurement gradients by multiplication of an $n \times n$ matrix with a size n vector, (9)
- Up to m subtractions, $m \cdot n$ multiplications and $m \cdot n$ additions to form the full measurement sum at $t = k$, (8)

To complete the estimation process, the gradient descent step needs to be performed (6), which requires n multiplications and n subtractions. Finally, a forward prediction of $\hat{\boldsymbol{\xi}}_{t-N|t}$ must be performed to actually know the current state $\hat{\boldsymbol{\xi}}_{t|t}$, which requires k evaluations of the prediction equation $\boldsymbol{f}$.

From these steps, the most expensive calculation in terms of computational complexity is the multiplication of two $n \times n$ matrices, which behaves as $\mathcal{O}(n^3)$ without using specialized algorithms. Neglecting lower order terms, our MHE therefore ends up with a computational complexity of $\mathcal{O}(N \cdot n^3)$. With respect to the number of states, our MHE therefore scales just as well as the EKF and the increase in complexity only comes from the number of time steps in the moving horizon.

Algorithm 1: Computationally efficient MHE.

input : u_t, y_t
output: $\hat{\xi}_t$
$N_t \leftarrow 0$
$\bar{u} \leftarrow [\,]$, $\bar{y} \leftarrow [\,]$
$\bar{\xi}_{\text{prior}} \leftarrow \xi_0$
while *true* **do**
  Update the moving window
  if $N_t < N_{max}$ **then**
    $\bar{u}[N_t] \leftarrow u_t$
    $\bar{y}[N_t] \leftarrow y_t$
    $N_t \leftarrow N_t + 1$
  else
    $\bar{\xi}_{\text{prior}} \leftarrow f(\hat{\xi}_{t-N}, \bar{u}[0])$
    `shift_left` $(\bar{u})$
    `shift_left` $(\bar{y})$
    $\bar{u}[N_t] \leftarrow u_t$
    $\bar{y}[N_t] \leftarrow y_t$
  Calculate the gradient of the cost function
  $\xi \leftarrow \bar{\xi}_{\text{prior}}$
  $D\text{F} \leftarrow \mathcal{I}_{n \times n}$
  $M_\Sigma \leftarrow 0$
  for $k \in [0, \ldots, N_t]$ **do**
    if $k \neq 0$ **then**
      $D\text{F} \leftarrow D\text{f}(\xi, \bar{u}[k-1]) \cdot D\text{F}$
      $\xi \leftarrow f(\xi, \bar{u}[k-1])$
    for $i \in [0, \ldots, m-1]$ **do**
      $\mathcal{H}_{i,k} = h_i(\xi, \bar{u}[k])$
      $\nabla \mathcal{H}_{i,k} = D\text{F}^T \cdot \nabla h_i(\xi, \bar{u}[k])$
      $M_\Sigma \leftarrow M_\Sigma - \theta_{i,k} \left(\bar{y}[k][i] - \mathcal{H}_{i,k}\right) \cdot \nabla \mathcal{H}_{i,k}$
  Calculate the new estimate
  $\nabla J \leftarrow -2 \cdot M_\Sigma$
  $\hat{\xi}_{t-N} \leftarrow \bar{\xi}_{\text{prior}} - \alpha \nabla J$
  $\hat{\xi}_t \leftarrow \hat{\xi}_{t-N}$
  for $k \in [0, \ldots, N_t]$ **do**
    $\hat{\xi}_t \leftarrow f(\hat{\xi}_t, \bar{u}[k])$
  `yield` $(\hat{\xi}_t)$

C. RANSAC Outlier Rejection

The occurrence of outliers in UWB measurements is a problem for any state estimator, as they can have a big impact on the estimated state. To deal with outliers, other authors have used the Mahalanobis distance [16], [17], which weighs the measurement error with its variance. To eliminate any reliance on the knowledge of error statistics, we have decided to use a random sample consensus (RANSAC) scheme instead.

To do this in an efficient manner, we split the measurement terms in (8) in two sums: A common sum and a RANSAC sum. The common sum includes measurements that are not subject to RANSAC outlier rejection (e.g., altitude measurements) while the RANSAC sum includes only a fraction of the terms from UWB measurements. By adding to multiple RANSAC sums at the same time, multiple gradients of the cost function can be computed after going through only one prediction cycle for the full horizon.

Since this removes the need to step through the full horizon several times, the only real contribution of the outlier rejection to the computational cost is from the evaluation of the different RANSAC estimates. The evaluation of every RANSAC estimate still requires an independent prediction cycle over the full horizon, to choose the best estimate.

IV. Implementation

In our implementation we limit ourselves to the estimation of the drone's position and velocity, while the attitude is estimated using Mahony's AHRS algorithm [20].

A. Prediction Equations

We use a slightly simplified version of the model in [16], where we represent the attitude with quaternions and neglect the dependence of the drag on the propellers' angular velocities.

$$\dot{\boldsymbol{x}} = \left[\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\rho} \end{bmatrix} \otimes \boldsymbol{q}^{-1}\right]_v \tag{16}$$

$$\dot{\boldsymbol{\rho}} = \frac{f}{m}\boldsymbol{e_3} + \left(\mathbf{K}_a - [[\boldsymbol{\omega}_\times]]\right)\boldsymbol{\rho} - \left[\boldsymbol{q}^{-1} \otimes \begin{bmatrix} 0 \\ \boldsymbol{g} \end{bmatrix} \otimes \boldsymbol{q}\right]_v \tag{17}$$

In this model, our six dimensional state $\boldsymbol{\xi} = [\boldsymbol{x}^T, \boldsymbol{\rho}^T]^T$ encompasses the position $\boldsymbol{x} \in \mathbb{R}^3$ in the global frame and the local velocities $\boldsymbol{\rho} \in \mathbb{R}^3$ in the body frame. The inputs to the system are the total thrust f, the attitude quaternion $\boldsymbol{q} \in \mathbb{R}^4$, and the angular velocities as measured by the gyroscopes, $\boldsymbol{\omega} \in \mathbb{R}^3$. The drone's mass m and drag coefficients $\mathbf{K}_\text{a} = \text{diag}(k_\perp, k_\perp, k_\parallel)$ enter as parameters. $\otimes$ is the quaternion product and the subscript 0 and v refer to the scalar and vector portion of the quaternion respectively. Finally, $\boldsymbol{g}$ is the directed acceleration of gravity and $[[\boldsymbol{\omega}_\times]]$ is the skew symmetric matrix defined as

$$[[\boldsymbol{\omega}_\times]] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{18}$$

Since inputs are usually available at higher rates than the estimator's update rate, we accumulate these inputs between two estimation cycles and average them to reduce input noise. For the thrust and each of the gyroscope's axes, this is done by simply calculating the mean, while for the attitude quaternion, the average is found by calculating the dominant eigenvector of

the matrix Q [21]:

$$Q = \sum_i \boldsymbol{q}_i \boldsymbol{q}_i^T \tag{19}$$

B. Measurement Equations

We consider two different types of UWB measurements that can be used with the estimator. In two-way ranging (TWR), the UWB system returns the distance between the tag at its current position, and the (known) location of an anchor, $\boldsymbol{p}_{\text{uwb},i}$. The measurement equation, including measurement noise η_{twr} is then given by

$$y_{\text{twr}} = \|\boldsymbol{p}_{\text{uwb},i} - \boldsymbol{x}\| + \eta_{\text{twr}} \tag{20}$$

Alternatively, time-difference of arrival (TdoA) measurements can be used. In TdoA schemes, the measurement yields the difference between the distances to two different anchors:

$$y_{\text{tdoa},ij} = \|\boldsymbol{p}_{\text{uwb},i} - \boldsymbol{x}\| - \|\boldsymbol{p}_{\text{uwb},j} - \boldsymbol{x}\| + \eta_{\text{tdoa}} \tag{21}$$

Since in most indoor environments, the distribution of anchors is mostly in the xy-plane with comparatively small variations in height, altitude estimation based on UWB alone is often difficult and noisy. We therefore also include the measurement equations for dedicated altitude measurements from a barometer or from laser time-of-flight (ToF) measurements:

$$y_{\text{baro}} = x_3 + z_{\text{ref}} + \eta_{\text{baro}} \tag{22}$$

$$y_{\text{tof}} = x_3 \cdot \cos\theta \cos\phi + \eta_{\text{tof}} \tag{23}$$

Note that the ToF measurement equation must take into account the current pitch θ and current roll ϕ since the laser is not measuring the shortest distance to the ground, but the distance on the body frame's z-axis.

V. Results

A. Setup

For testing and evaluation, we implemented our own algorithm and the EKF from [16] (which is used on the Crazyflie) in Python.[1] We compared their real-time position estimates at each timestep t, $\hat{\boldsymbol{\xi}}_{t|t}$, with ground-truth from an optitrack motion capture system. The evaluation was performed on real IMU and UWB data, which was collected on a Crazyflie drone at 100 Hz, and optitrack data collected in the "Cyberzoo" testing area at the Aerospace Faculty of TU Delft[2] (seen in Fig. 2). In total, 24 data sets were gathered on 6 different trajectories, containing either two-way ranging (TWR) or time-difference of arrival (TdoA) data from the UWB system. The different trajectories used were a square, a triangle, an octagon, an hourglass, a five-pointed star and a random sequence of points. On each trajectory, we recorded four runs, two for TWR data and two for TdoA data. The UWB setup included eight beacons, positioned roughly in the eight corners of a cube, enclosing our testing area (cf. Table I). An example for a run of the Octagon trajectory where we test the estimator performance offline but on real TWR data is shown in Fig. 3.

[1]https://github.com/SUPfeiffer/uwb-simulator

[2]Data available at https://doi.org/10.4121/14 827 680

Fig. 2. Flight data was collected in the "Cyberzoo" at TU Delft using a Crazyflie drone and the loco positioning system by bitcraze. Groundtruth is provided by an Optitrack motion capture system.

TABLE I
UWB Beacon Positions

ID	x [m]	y [m]	z [m]
0	-4.48	-4.87	1.16
1	-4.78	4.52	0.67
2	4.81	4.57	0.81
3	4.91	-4.77	0.96
4	-4.34	-5.05	2.60
5	-4.91	4.41	2.65
6	4.56	4.77	2.59
7	4.91	-4.77	2.53

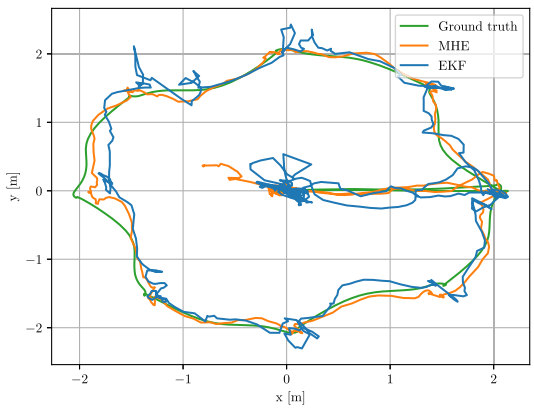

Fig. 3. Example of one of the collected trajectories (Octagon) with offline performance of MHE and EKF on real TWR data from 4 beacons.

B. Heavy-Tailed Measurement Noise

While the noise on TWR measurements can be assumed to be Gaussian in an ideal scenario, it is more realistically modelled as being heavy-tailed. This is due to multi-pathing and non-line-of-sight (NLOS) effects, which cause UWB signals to arrive with a slight delay compared to the shortest path [3], [4]. To simulate heavy-tailed noise for TWR, we adapt the noise model by Jimenez and Seco [3], which models the heavy-tailed noise as the sum of a Gaussian distribution for the LOS component, a Gamma distribution for the NLOS component, and a constant

for outliers. To avoid dealing with a probability density function (PDF) that does not integrate to 1, we remove the constant, and rely on getting outliers from the Gamma distribution alone. To show the effect of an increasingly non-Gaussian distribution, we multiply the Gamma distribution with a scale factor s_{ht}, which we vary between 0 and 1.25. Since a reduction in the NLOS component should also cause the mean to go towards zero, we multiply μ by the same factor, which causes the noise distribution to represent the ideal LOS condition (i.e. a pure, zero-mean Gaussian) for $s_{\text{ht}} = 0$. Finally, we divide the PDF by $(1 + s_{\text{ht}})$, which causes the integral to be equal to 1. The resulting model is given in (24).

$$f(x) = \frac{1}{1+s_{\text{ht}}} \cdot \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - s_{\text{ht}}\mu)^2}{2\sigma^2}\right) + \frac{s_{\text{ht}}}{1+s_{\text{ht}}} \cdot \frac{\lambda^k}{(k-1)!} x^{k-1} \exp\left(-\lambda x\right) \tag{24}$$

Even though there are significant differences between individual anchors, we find that using the parameters suggested by Jimenez and Seco ($\mu = 0.1$, $\lambda = 3.5$, $k = 2$) and a standard deviation of $\sigma = 0.1$ (determined from our own data), our TWR data is best fit with a heavy-tail scale factor of $s_{\text{ht}} = 0.2$. The main differences between anchors are the need for some higher values for λ and k, but we also noted some anchors with shifted or multiple peaks in the noise distribution. These odd behaviours might originate from localized biases on different trajectories and were not modeled in this research.

Since TdoA measurements can be seen as resulting from two TWR measurements to different beacons, the heavier tail appears on both sides and is better modeled by using a Cauchy distribution. To vary the intensity of the heavy-tail, we model the noise distribution for TdoA as the weighted sum of a Cauchy distribution ($\gamma = 0.3$) and a Gaussian distribution ($\sigma = 0.3$), both centered around $\mu = 0$. The resulting noise model, shown in (25) best matches the recorded TdoA data for a relative weight of the Cauchy distribution of $r_{\text{ht}} = 0.5$.

$$f(x) = (1 - r_{\text{ht}}) \cdot \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) + r_{\text{ht}} \cdot \left[\pi\gamma\left(1 + \left(\frac{x}{\gamma}\right)^2\right)\right]^{-1} \tag{25}$$

Using these noise models, we compare the performance of MHE and EKF on a mix of real IMU data combined with simulated UWB measurements from 4 anchors (IDs 1, 3, 4, 6) that provide measurements on average every 0.05 s (TWR) or 0.1 s (TdoA) each. We run the simulation 5 times on every set of IMU data collected, for a total of 120 runs per value of s_{ht} and r_{ht}. Fig. 5 and Fig. 6 show the average RMSE of these runs for TWR and TdoA measurements respectively, with the shaded regions representing the standard error of the mean.

In the case of TWR, both estimators perform similarly well, with the EKF having a slight edge for Gaussian noise and the MHE showing an advantage for noise with a stronger heavy tail. In the case of TdoA, the MHE performs better over the full range of simulated noise, with the difference increasing at higher values of r_{ht}. The cause for this difference between TWR and TdoA (also for the Gaussian noise) most likely stems from the larger standard deviation for TdoA, which also leads to more outliers.

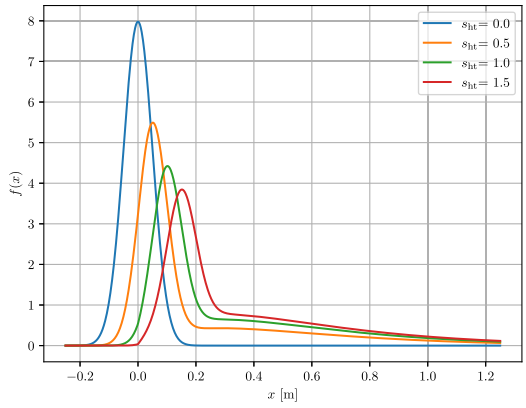

Fig. 4. Probability Density Function (PDF) of the heavy-tailed noise distribution used for generating UWB measurements. The scale factor s_{ht} is used to change the influence of the heavy-tailed Gamma distribution added to the underlying Gaussian.

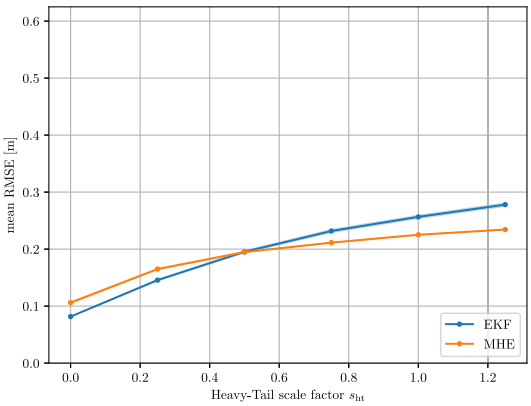

Fig. 5. Estimator performance on TWR measurements with increasingly heavy-tailed noise. Shaded regions represent the standard error of the mean.

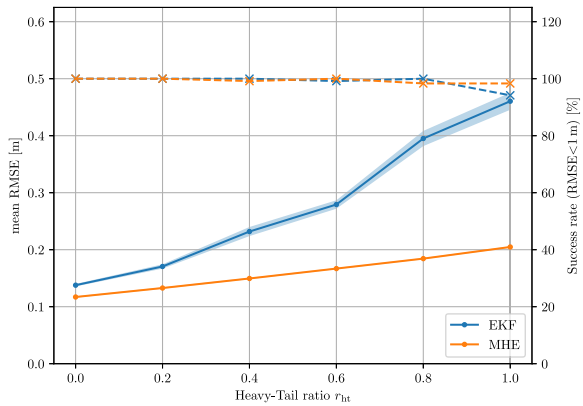

Fig. 6. Estimator performance on TdoA measurements with increasingly heavy-tailed noise. Dashed lines show success rate of the estimator (RMSE<1 m). Unsuccessful runs are excluded from the calculation of the mean RMSE.

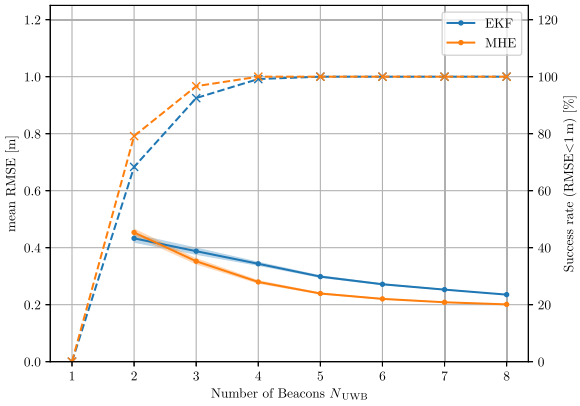

Fig. 7. Estimator performance on TWR measurements with varying number of beacons. Dashed lines show the success rate (RMSE<1 m), shaded regions represent the standard error of the mean.

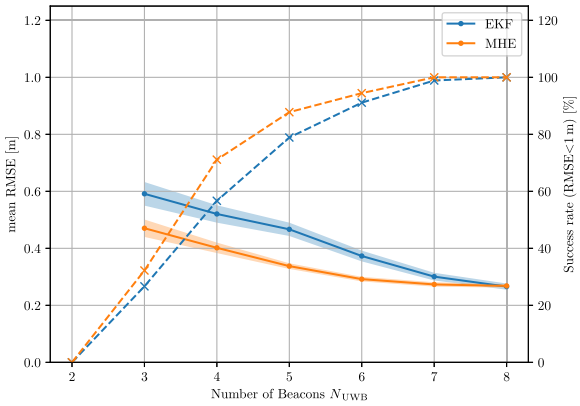

Fig. 8. Estimator performance on TdoA measurements with varying number of beacons. Dashed lines show the success rate (RMSE<1 m), shaded regions represent the standard error of the mean.

C. Removing UWB Anchors

Of particular interest in the field of UWB localization is the reduction of required fixed infrastructure, namely the number of beacons that need to be placed in the environment. To assess the performance of our estimator in applications with respect to the number of beacons, we use our full data sets (IMU and UWB) and remove the measurements from randomly selected beacons. Specifically, we run the filters with real UWB data with the number of enabled beacons ranging from 1-8 for TWR and 2-8 for TdoA (TdoA measurements require two beacons). We perform 10 runs for every number of beacons on each of the 12 corresponding (TWR or TdoA) data sets, for a total of 120 runs per number of beacons. For each individual run, we randomly select the beacons used, since the placement of beacons with respect to the trajectory was found to have a significant effect on performance for low numbers of beacons. Due to beacons losing connection on three TdoA trajectories, we only use 9 of the 12 data sets for the TdoA comparison. Results of these runs are shown in Fig. 7 and Fig. 8 respectively.

One of the downsides of the proposed gradient-descent method is that the magnitude of the cost function's gradient $\nabla J(\hat{\boldsymbol{\xi}}_{t-N|t})$ varies with the number of measurements. This means that the value of the step size α had to be slightly adjusted depending on the number of beacons.

As was to be expected, the lower number of beacons causes a significant increase in the estimation error. While the MHE still shows an advantage, the observed errors are higher than with simulated measurements. This is likely due to inaccuracies in measuring out the anchor positions, and spatially varying biases that have also been mentioned by [16]. In fact, we found that measurements from some beacons exhibit far more challenging noise than what is modeled by a heavy-tailed distribution, such as multiple peaks. For TDOA measurements, both estimators are failing quite often. Indeed, there seems to be a much larger discrepancy between simulated and real TDOA data than for TWR. This is probably due to the fact, that with a large number of beacons, the TDOA algorithm requires a lot of communication between beacons, causing the frequency of the individual measurements to go down as more beacons are present. When removing beacons after the measurements were taken, this leads to much lower measurement frequencies than otherwise expected. It is however still visible, that the reduced measurement frequency seems to have a smaller effect on the MHE's performance.

D. Computational Complexity

To compare the computational complexity, we implement our algorithm as a separate task on a Crazyflie drone which performs these computations on an STM32F405 microcontroller. This task replaces the EKF task in the original Crazyflie firmware, so that only one estimator is running at the same time.

By analyzing task dumps, we can identify how much of the available computational resources a task uses. When using the same settings as in simulation, the MHE task accounted for about 15% of the CPU load, while the EKF only accounted for about 11%. This shows the potential of the algorithm to indeed be used on computationally limited devices. With more potent microprocessors being released continuously, we believe that computationally efficient MHE algorithms can become useful for many applications, for which they were previously considered too expensive.

VI. Conclusion

A computationally efficient MHE algorithm for quadrotor localization with UWB was developed and tested. We showed that by simplifying the problem to an output noise MHE and using a single-iteration gradient descent method, MHE can yield good results on computationally limited devices. While these simplifications helped us in reducing the computational cost of MHE, they also pose some limitations on the applications of the algorithm.

The simplification to an output noise MHE assumes that the process model is perfectly known and does not exhibit any process noise, which is rarely the case. Even though the algorithm performed well, there is potential for improvement, especially when only few measurements are available.

Using a simple single-iteration gradient descent algorithm causes the optimization for the correct state to happen over

several time steps, and requires the step-size α to be well chosen to keep the algorithm stable. Especially when the number of measurements is varying, this can become difficult, due to the varying magnitude of the gradient of the cost function. We found that conservative values of α can yield stable results for large variations in the number of measurements, but the performance will be worse when fewer measurements are available.

We plan to address these issues in future research to further improve the accuracy of the proposed method and make it useful for a wider variety of tasks.

REFERENCES

[1] D. Simon, *Optimal State Estimation*. Hoboken, NJ, USA: Wiley, 2006.

[2] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. Signal Process., Sensor Fusion, Target Recognit. VI*, I. Kadar, Ed., Int. Soc. Opt. Photon, vol. 3068, 1997, pp. 182–193.

[3] A. R. Jiménez and F. Seco, "Comparing ubisense, BeSpoon, and DecaWave UWB location systems: Indoor performance analysis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2106–2117, Aug. 2017.

[4] M. Kok, J. D. Hol, and T. B. Schon, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1293–1303, Apr. 2015.

[5] D. A. Allan and J. B. Rawlings, "Moving horizon estimation," in *Proc. Handbook Model Predictive Control*, S. V. Raković and W. S. Levine, Eds. Cham, Switzerland: Springer Nature, pp. 99–144, 2019.

[6] D. G. Robertson, J. H. Lee, and J. B. Rawlings, "A moving horizon-based approach for least-squares estimation," *AIChE J.*, vol. 42, no. 8, pp. 2209–2224, 1996.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[9] A. H. Jazwinski, "Limited memory optimal filtering," *IEEE Trans. Automat. Contr.*, vol. AC-13, no. 5, pp. 558–563, Oct. 1968.

[10] Y. S. Shmaliy, S. Zhao, and C. K. I. Ahn, "Unbiased finite impulse response filtering," *IEEE Control Syst. Mag.*, vol. 37, no. 5, pp. 70–89, Oct. 2017.

[11] V. M. Zavala, C. D. Laird, and L. T. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *J. Process Control*, vol. 18, no. 9, pp. 876–884, 2008.

[12] P. Kühl, M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock, "A real-time algorithm for moving horizon state and parameter estimation," *Comput. Chem. Eng.*, vol. 35, no. 1, pp. 71–83, 2011.

[13] A. Alessandri and M. Gaggero, "Fast moving horizon state estimation for discrete-time systems using single and multi iteration descent methods," *IEEE Trans. Automat. Contr.*, vol. 62, no. 9, pp. 4499–4511, 2017.

[14] F. Girrbach, J. D. Hol, G. Bellusci, and M. Diehl, "Optimization-based sensor fusion of GNSS and IMU using a moving horizon approach," *Sensors* (Basel, Switzerland), vol. 17, no. 5, pp. 1–17, 2017.

[15] S. Li, E. van der Horst, P. Duernay, C. De Wagter, and G. C. de Croon, "Visual model-predictive localization for computationally efficient autonomous racing of a 72-g drone," *J. Field Robot.*, vol. 37, no. 4, pp. 667–692, 2020.

[16] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1730–1736.

[17] Y. Xu, C. K. Ahn, Y. S. Shmaliy, X. Chen, and Y. Li, "Adaptive robust INS/UWB-integrated human tracking using UFIR filter bank," *Measurement: J. Int. Meas. Confederation*, vol. 123, pp. 1–7, Jul. 2018. [Online]. Available: https://doi.org/10.1016/j.measurement.2018.03.043

[18] Y. Xu, Y. S. Shmaliy, T. Shen, D. Chen, M. Sun, and Y. Zhuang, "INS / UWB-Based quadrotor localization under colored measurement noise," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6384–6392, Mar. 2021.

[19] A. Liu, W. A. Zhang, M. Z. Chen, and L. Yu, "Moving horizon estimation for mobile robots with multirate sampling," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1457–1467, Feb. 2017.

[20] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Automat. Contr.*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.

[21] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *J. Guid., Control, Dyn.*, vol. 30, no. 4, pp. 1193–1197, 2007.