Makridis, Evagoras; Charalambous, Themistoklis

Towards Robust Onboard Control for Quadrotors via Ultra-Wideband-based Localization

Published in:
Proceedings of the 16th IEEE International Wireless Communications and Mobile Computing, IWCMC 2020

DOI:
10.1109/IWCMC48107.2020.9148351

Published: 01/06/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Makridis, E., & Charalambous, T. (2020). Towards Robust Onboard Control for Quadrotors via Ultra-Wideband-based Localization. In *Proceedings of the 16th IEEE International Wireless Communications and Mobile Computing, IWCMC 2020* (pp. 1630-1635). Article 9148351 IEEE. https://doi.org/10.1109/IWCMC48107.2020.9148351

Towards Robust Onboard Control for Quadrotors via Ultra-Wideband-based Localization

Evagoras Makridis and Themistoklis Charalambous

***Abstract*—This paper describes an indoor navigation approach using estimation and control for horizontal translational motion and heading angle for quadrotor Unmanned Aerial Vehicles (UAVs) via Ultra-Wideband (UWB)-based localization. In particular, to cope with noisy measurements, emanating from model uncertainties, and Non-Line-Of-Sight (NLOS) conditions, a Linear Quadratic Regulator (LQR) is deployed along with a Maximum Correntropy Criterion Kalman Filter (MCC-KF). This approach has proven improved robustness compared to the traditional Kalman Filter (KF) against non-Gaussian noise. A testbed with a quadrotor was developed for evaluating the performance of our proposed approach. We demonstrate, via the experimental setup, that the MCC-KF outperforms the use of KF in the presence of shots of mixed noise and communication delays, enabling onboard robust estimation and control via UWB-based localization.**

***Index Terms*—Quadrotor control, ultra-wideband communications, linear quadratic regulator, maximum correntropy criterion Kalman filter.**

I. Introduction

During the past decades, multi-rotor UAVs managed to gain the attention of commercial and scientific communities as a result of the myriad indoor and outdoor applications, such as monitoring and inspection, warehouse inventory management, and delivery. Such tasks require autonomy, high accuracy, and robustness which add extra constraints to the design and development of control algorithms. The prospect of conducting such tasks autonomously, has been driving research into automating UAV navigation. Assigning to a UAV a target position or a trajectory in space imposes the need for a feedback controller to compute the control command based on the actual and the target *pose* (note: pose, in contrast to position, contains also orientation). The actual position of a UAV in space can be measured using localization techniques, while the orientation can be measured with Inertial Measurement Units (IMUs).

Several technologies for UAV localization such as Global Positioning Systems (GPS), vision-based systems and wireless-based systems have been proposed and evaluated in the literature. Although GPS are the most common localization systems for outdoor environments, they are not suitable for indoor applications, as the satellite signals can not pass through buildings, nor they are accurate enough to be used for precision tasks [1]. Vision-based approaches to measure a UAV's pose in real-time also exist, fusing inertial sensing together with either off-board or on-board visual sensing. The former set of techniques rely on expensive motion-capture systems with carefully calibrated off-board cameras [2]–[4], limiting the portability and generality of such methods. On-board techniques, on the other hand, rely on on-board cameras, often monocular [5]–[8], creating a cheaper alternative, albeit less robust to changes in illumination and viewpoint [9]. Although computer vision techniques are remarkably popular to measure the position and the orientation of quadrotors, several barriers (such as, the cost, the needed computational power and lack of accuracy) prevent their use.

E. Makridis is with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. E-mail: evagoras@kth.se

T. Charalambous is with the School of Electrical Engineering, Aalto University. E-mail: themistoklis.charalambous@aalto.fi

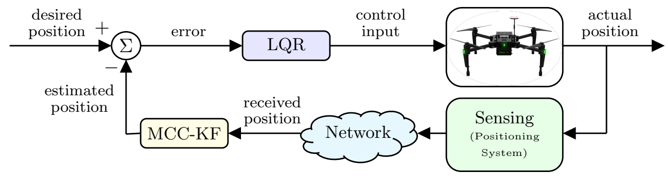

Fig. 1: Block diagram of the feedback loop.

From another strand of research, Ultra-Wideband (UWB) wireless technology for quadrotor localization has gained significant attention during the past few years [10]–[13], because of the low positioning error (i.e., usually $< 20cm$), the low cost and the ease of installation. UWB for localization is a large bandwidth radio technology which measures the position of an object carrying a transmitter (tag) from several receivers (anchors). Despite its accuracy in static scenarios, UWB-based localization usually induces communication delays to the sensor-controller links, which in addition to sensor faults, measurement noise, and non-line of sight (NLOS), affect the performance of UAV.

This paper presents linear optimal and robust control techniques based on state estimation from a KF and a MCC-KF to attenuate noise, reject external disturbances to track a predefined reference horizontal position and heading (yaw angle) of a quadrotor UAV using IMU and UWB-based localization measurements.

The remainder of this paper is organized as follows: related work regarding quadrotor control techniques are presented and discussed in Section II. Notation and system description including the definitions of the coordinate systems, quadrotor dynamics and system identification are given in Section III and Section IV respectively. Section V presents the proposed state estimation algorithms and the controller. These control techniques are evaluated with the experimental setup and results presented in Section VI. Finally, conclusions and directions for future work are given in Section VII.

II. Related work

Bouabdallah *et al.* in a series of works [14]–[16] were among the first who designed and applied linear and non-linear control for the position and orientation of quadrotors. In particular, they used Lyapunov theory, Proportional-Integral-Derivative (PID) controllers, optimal control theory, backstepping, sliding-mode and integral backstepping techniques to control the position and the orientation of a quadrotor.

Raffo *et al.* [17], [18] proposed non-linear robust control strategies to track a reference trajectory of a quadrotor. In [17] they used an integral model predictive controller with disturbance rejection to control the translational movements of the quadrotor, while in [18] they introduced a control law based on backstepping approach. In both works, they used an inner loop non-linear $\mathcal{H}_\infty$ controller for the stabilization of the rotational movements.

In [19], the authors studied the hovering performance of a quadrotor after the performed dynamic system identification, state estimation using a complementary filter and control using PID controllers. They presented computationally efficient control algorithms comparable with the current state-of-the-art techniques. Papastratis *et al.* in [12], studied the interplay between communication and control of a quadrotor using measurements from an Ultra-Wideband (UWB) positioning system. In particular, they studied how the performance of a PID controller affects the stabilization of a quadrotor (DJI M100) with the presence of communication delays from the UWB positioning system.

More recently, Sa *et al.* [4] performed dynamic system identification and control for the DJI M100. They used only a built-in IMU to identify the dynamics of the vehicle and the DJI M100 autopilot (N1 flight controller). Based on this model, they designed a Model Predictive Controller (MPC) to control the quadrotor using IMU and motion capture system (Vicon) measurements. In [20], the authors proposed and evaluated a state-depended Linear Quadratic Regulator (LQR) controller which unifies the control of the rotational and translational states from a Visual Inertial Odometry and IMU. A small quadrotor using an onboard ARM platform and running the full state estimation and the LQR computation was used for evaluation.

III. Notation

In this work, vectors, matrices and sets, are denoted by bold lowercase, uppercase and calligraphic uppercase letters, respectively. Real and nonnegative real numbers sets are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. The transpose matrix of matrix A is denoted with A^T and its inverse with A^{-1}. The notation $A \succeq 0$ ($A \succ 0$) means that matrix A is semi-positive (positive) definite. The identity matrix is represented by I. $\mathbb{E}\{\cdot\}$ represents the expectation of its argument. Given any vector norm $\|\cdot\|$, a weighted vector norm can be written as $\|x\|_A \triangleq \|Ax\|$, where A is an arbitrary nonsingular matrix. The sine and cosine of an angle θ is denoted by $s\theta \equiv \sin(\theta)$ and $c\theta \equiv \cos(\theta)$, respectively.

IV. System Description

A. Coordinate Systems

First, we determine two coordinate frames using the standard right-handed robotics convention as shown in Fig.2. The Earth's inertial frame $\{E\}$ follows the East-North-Up (ENU) reference system where $+x$ axis points to the east, $+y$ to the north and $+z$ points upwards based on the right-hand rule. The quadrotor's Body frame $\{B\}$, which is coincident to the origin and thus to the absolute position of the quadrotor (i.e., $[x,\ y,\ z]$), follows the Forward-Left-Up (FLU) which gives forward horizontal, left horizontal and up vertical movement along its $+x$, $+y$ and $+z$ axis respectively.

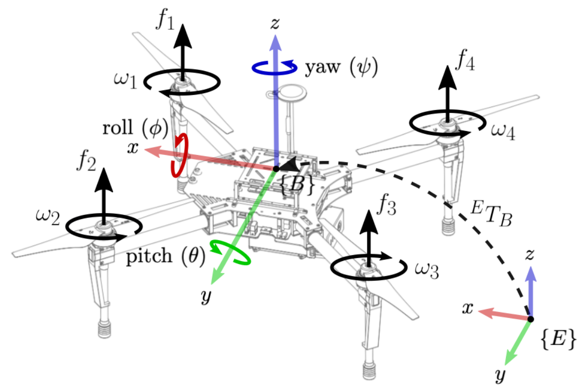

Fig. 2: Quadrotor model with robotic frames diagram.

B. Quadrotor Rigid-Body Dynamics

The open-loop system of a quadrotor is an unstable non-linear complex system. Each rotor consists a propeller and a motor which produces an angular velocity ω_i which implies a thrust force f_i, where i denotes the number of the motor as shown in Fig.2. Note that, two rotors rotate clockwise, while the other two rotate counter-clockwise to achieve a zero net angular momentum and thus cancel the yaw (ψ) rotation along the z-axis of the quadrotor's body. With different values of angular velocity on each motor, the quadrotor moves to different translational (i.e., $\boldsymbol{\xi} = (x, y, z) \in \mathbb{R}^3$) and rotational (i.e., $\boldsymbol{\eta} = (\phi, \theta, \psi) \in \mathbb{R}^3$) coordinates in the Earth's frame where x, y and z represent the position coordinates of the center of mass of the quadrotor from the Earth's inertial frame $\{E\}$. The Euler's angles ϕ, θ and ψ represent the orientation of the quadrotor. As it can be seen from Fig.2; ϕ is the angle about the x-axis and it is called "roll", θ is the angle about the y-axis and is called "pitch", and ψ is the angle about the z axis and is called "yaw".

The translational equations of motion in the Earth's frame are given by the Newton-Euler formalism [21], [22]:

$$m\ddot{\boldsymbol{\xi}} = \boldsymbol{f}_B - \boldsymbol{f}_G, \tag{1}$$

where m denotes the mass of the quadrotor, $\boldsymbol{f}_B$ denotes the forces acting on the quadrotor in the Earth's frame. Therefore, it is needed to transform the thrust forces f_i acting on the quadrotor's body, to the Earth's inertial frame $\{E\}$ using the rotation matrix R given in (2). Multiplying the rotation matrix

R with the total thrust force f_T (acting only on the vertical z-axis), we have the forces acting on the quadrotor, in the Earth's frame $\{E\}$:

$$\boldsymbol{f}_B = \underbrace{\begin{bmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\theta & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - c\psi s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}}_{R} \begin{bmatrix} 0 \\ 0 \\ f_T \end{bmatrix}, \quad (2)$$

where $f_T = k_F(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$ is the total thrust which in hovering state is equal to the gravitational force mg, and k_F is the aerodynamic drag coefficient due to air density. Thus, the total non-gravitational force acting on the quadrotor's Body frame is given by:

$$\boldsymbol{f}_B = f_T \begin{bmatrix} c\phi s\theta c\psi + s\psi s\phi & c\phi s\theta s\psi - c\psi s\phi & c\phi c\theta \end{bmatrix}^T. \quad (3)$$

The gravitational force $\boldsymbol{f}_G$ acts only on the z-axis when hovering and it is described by $\boldsymbol{f}_G = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T$, where g is the acceleration of the gravity. Finally, substituting $\boldsymbol{f}_G$ and $\boldsymbol{f}_B$ into (1) we get the translational equation of motion which is given by:

$$m\ddot{\boldsymbol{\xi}} = f_T \begin{bmatrix} (c\phi s\theta c\psi + s\psi s\phi) \\ (c\phi s\theta s\psi - c\psi s\phi) \\ (c\phi c\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \quad (4)$$

Assuming that the quadrotor will be hovering, we can linearize the aforementioned non-linear equations in (4) around the hovering operating point using small angle assumptions (i.e., $f_T \simeq mg$; $\dot{\phi}, \dot{\theta}, \dot{\psi} \simeq 0$; $s\psi \simeq \psi$, $s\theta \simeq \theta$, $s\phi \simeq \phi$; $c\phi = 1$). In addition, we assume the yaw angle (ψ) to be fixed (i.e., ψ_0), while the quadrotor is moving in the horizontal plane (i.e., x and y axes) in order to cancel of the non-linear term in (5). Finally, we assume that our control algorithms will be applied only on the translational horizontal dynamics and not on the vertical z-axis. Hence, the linearized equations of translational motion for the horizontal plane are given by:

$$\ddot{x} = g\left(c\psi_0\theta + s\psi_0\phi\right), \quad \ddot{y} = g\left(s\psi_0\theta - c\psi_0\phi\right). \quad (5)$$

C. System Identification

For our experimental setup, we use a DJI Matrice 100 (M100) which runs a low level built-in *N1 flight controller* to stabilize the roll, pitch angles, the yaw rate and the vertical velocity. However, there is no information provided regarding these dynamics in the project source code of DJI (Onboard-SDK). Thus, to estimate the dynamic model of the low level built-in Euler angles' controller of the quadrotor we use system identification as firstly done in [4] and [23]. Recording the desired input signals (i.e., u_ϕ, u_θ and $u_{\dot{\psi}}$) that are given through the Onboard-SDK and the measured output signals (i.e., ϕ, θ and $\dot{\psi}$), we can identify the dynamical model coefficients $\alpha_{0,\phi}$, $\alpha_{1,\phi}$, $\alpha_{0,\theta}$, $\alpha_{1,\theta}$, $\alpha_{0,\psi}$, $\beta_{0,\phi}$, $\beta_{0,\theta}$ and $\beta_{0,\psi}$. The dynamical model to be identified is given by the following differential equations (roll and pitch are described by a second order transfer function, while the yaw rate by a first order):

$$\begin{aligned} \ddot{\phi} &= -\alpha_{0,\phi}\phi - \alpha_{1,\phi}\dot{\phi} + \beta_{0,\phi}u_\phi, \\ \ddot{\theta} &= -\alpha_{0,\theta}\theta - \alpha_{1,\theta}\dot{\theta} + \beta_{0,\theta}u_\theta, \\ \ddot{\psi} &= -\alpha_{0,\psi}\psi + \beta_{0,\psi}u_{\dot{\psi}}. \end{aligned} \quad (6)$$

V. Optimal and Robust Control Design

The continuous-time state-space model of the quadrotor is given by:

$$\begin{aligned} \mathrm{d}\boldsymbol{x}(t) &= A\boldsymbol{x}(t)\mathrm{d}t + B\boldsymbol{u}(t)\mathrm{d}t + \mathrm{d}\boldsymbol{w}(t), \\ \mathrm{d}\boldsymbol{y}(t) &= C\boldsymbol{x}(t)\mathrm{d}t + \mathrm{d}\boldsymbol{v}(t), \end{aligned} \quad (7)$$

where $\boldsymbol{x} \in \mathbb{R}^{10}$ is the system state vector (i.e., $\boldsymbol{x} = [\phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ x\ \dot{x}\ y\ \dot{y}]^T$), $\boldsymbol{u} \in \mathbb{R}^3$ is the control input vector (i.e., $\boldsymbol{u} = [u_\phi\ u_\theta\ u_{\dot{\psi}}]^T$), $\boldsymbol{y} \in \mathbb{R}^6$ is the measurement vector (i.e., $\boldsymbol{y} = [\phi\ \theta\ \psi\ z\ x\ y]^T$), while $\boldsymbol{w} \in \mathbb{R}^{10}$ and $\boldsymbol{v} \in \mathbb{R}^6$ are zero-mean disturbance stochastic processes, representing the process and measurement noise levels. The discrete-time equivalent state-space model of the quadrotor is given by:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \Phi\boldsymbol{x}_k + \Gamma\boldsymbol{u}_k + \boldsymbol{w}_k, \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k + \boldsymbol{v}_k, \end{aligned} \quad (8)$$

where the discrete equivalent system matrix $\Phi = e^{Ah}$, and the control input matrix $\Gamma = \int_{s=0}^{h} e^{As}Bds$, and h is the sampling period (i.e., control loop period). The measurement matrix $C \in \mathbb{R}^{6\times10}$. The process and measurement noise levels are represented by $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$, respectively. Both are assumed to be white Gaussian random sequences with zero mean, with $\mathbb{E}\{\boldsymbol{w}_k\} = 0$, $\mathbb{E}\{\boldsymbol{v}_k\} = 0$, $\mathbb{E}\{\boldsymbol{w}\boldsymbol{w}^T\} = W \succeq 0$, and $\mathbb{E}\{\boldsymbol{v}\boldsymbol{v}^T\} = V \succ 0$.

The *a priori* and *a posteriori* state estimates are denoted by $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{x}}_{k|k}$, respectively. The corresponding error covariance matrices are defined by:

$$P_{k|k-1} \triangleq \mathbb{E}\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^T\}, \quad (9)$$

$$P_{k|k} \triangleq \mathbb{E}\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})^T\}. \quad (10)$$

In this work, we propose the use of the optimal linear controller (i.e., LQR) and a robust estimator (i.e., MCC-KF) to compensate sensor noises and reject external disturbances applied on an indoor flying quadrotor.

A. Maximum Correntropy Criterion Kalman Filter (MCC-KF)

In this section an alternative Kalman filter using the Maximum Correntropy Criterion (MCC) for state estimation is used to deal with process and measurement noises that are non-Gaussian, e.g. shot noise or mixture of Gaussian noise [24], [25]. The correntropy criterion measures the similarity of two random variables using information from high-order signal statistics in contrast with the classical Kalman filter that considers only the second-order moment of stability [26]. The equations for the MCC-KF are summarized below [25]:

$$\hat{\boldsymbol{x}}_{k|k-1} = \Phi\hat{\boldsymbol{x}}_{k-1|k-1} + \Gamma\boldsymbol{u}_k, \quad (11)$$

$$P_{k|k-1} = \Phi P_{k-1|k-1}\Phi^T + W_k, \quad (12)$$

$$L_k^{mcc} = \frac{G_\sigma\left(\|\,\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k|k-1}\|_{V_k^{-1}}\right)}{G_\sigma\left(\|\,\hat{\boldsymbol{x}}_{k|k-1} - \Phi\hat{\boldsymbol{x}}_{k-1|k-1}\|_{P_{k|k-1}^{-1}}\right)}, \quad (13)$$

$$K_k = (P_{k|k-1}^{-1} + L_k^{mcc}C^TV_k^{-1}C)^{-1}L_k^{mcc}C^TV_k^{-1}, \quad (14)$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + K_k(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k|k-1}), \quad (15)$$

$$P_{k|k} = (I - K_kC)P_{k|k-1}(I - K_kC)^T + K_kV_kK_k^T, \quad (16)$$

where G_σ is the Gaussian kernel, i.e.,

$$G_\sigma(\| \boldsymbol{x}_i - \boldsymbol{y}_i \|) = \exp\left(-\frac{\| \boldsymbol{x}_i - \boldsymbol{y}_i \|^2}{2\sigma^2}\right),$$

with kernel size σ (representing a weighting parameter between the second and higher-order moments). Note that L_k^{mcc} is the *minimized correntropy estimation cost function* and K_k is the Kalman gain.

B. Linear Quadratic Regulator (LQR) Control

To follow a reference signal $\boldsymbol{r}_k = [r_x, r_y, r_\psi]^T$, we augment the state-space model by adding integral state, $\boldsymbol{i}_{k+1} = \boldsymbol{i}_k + \boldsymbol{r}_k - E\boldsymbol{y}_k$:

$$\underbrace{\begin{bmatrix}\hat{\boldsymbol{x}}_{k+1|k}\\ \boldsymbol{i}_{k+1}\end{bmatrix}}_{\bar{\boldsymbol{x}}_{k+1}} = \begin{bmatrix}\Phi - K_kC & 0\\ 0 & I\end{bmatrix}\underbrace{\begin{bmatrix}\hat{\boldsymbol{x}}_{k|k-1}\\ \boldsymbol{i}_k\end{bmatrix}}_{\bar{\boldsymbol{x}}_k} + \begin{bmatrix}\Gamma\\ 0\end{bmatrix}\boldsymbol{u}_k + \begin{bmatrix}0 & K_k\\ I & -E\end{bmatrix}\begin{bmatrix}\boldsymbol{r}_k\\ \boldsymbol{y}_k\end{bmatrix}, \tag{17}$$

where $E \in \mathbb{R}^{3\times 6}$ is a matrix that selects the observations to be controlled (i.e., $E = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]^T$, where $\boldsymbol{e}_1 = [0,0,0,0,1,0]$, $\boldsymbol{e}_2 = [0,0,0,0,0,1]$, $\boldsymbol{e}_3 = [0,0,1,0,0,0]$). The optimal control law $\boldsymbol{u}_k^* = -L\bar{\boldsymbol{x}}_k = -L^{\hat{x}}\hat{\boldsymbol{x}}_{k|k-1} - L^i\boldsymbol{i}_k$ can be found by minimizing the linear quadratic criterion given in (18):

$$J = \mathbb{E}\left[\bar{\boldsymbol{x}}_N^{\mathrm{T}}\bar{Q}_N\bar{\boldsymbol{x}}_N + \sum_{k=0}^{N-1}\left(\bar{\boldsymbol{x}}_k^{\mathrm{T}}\bar{Q}_k\bar{\boldsymbol{x}}_k + \boldsymbol{u}_k^{\mathrm{T}}R_k\boldsymbol{u}_k\right)\right], \tag{18}$$

where $\bar{Q}_N \succeq 0$, $\bar{Q}_k \succeq 0$ are the final and stage state error weighting matrices respectively, and $R_k \succ 0$ is the stage control weighting matrix for the LQ problem. The optimal control gain $L = \begin{bmatrix}L^{\hat{x}} & L^i\end{bmatrix}$ is the state-feedback controller gain given by the solution of the *Riccati* equation.

VI. Experimental Validation

To evaluate the performance of the proposed control algorithms,we setup a real testbed[1]. For the following autonomous flights, we setup an onboard Raspberry Pi 3 B+ running Ubuntu server 16.04.6 LTS (Xenial Xerus) with the Robotic Operating System (ROS) Kinetic on arm64. The ROS is used to establish the communication between the main components of the system (i.e., quadrotor, sensors and controller) running the DJI Onboard-SDK ROS wrappers[2].

A. Sensing, Estimation and Control

For measuring the orientation of the quadrotor, we utilized the onboard IMU node of the Onboard-SDK. This node publishes quadrotor's attitude with a default update rate of 100Hz with respect to the quadrotor's Body frame $\{B\}$ in quaternions which are then transformed to radians.

For calculating the position of the quadrotor we developed a ROS node for publishing the measurements of the UWB Pozyx positioning system. In this work, we configured the Pozyx with the highest performance settings (i.e., bitrate of 6.81Mbits/sec, pulse repetition frequency of 64MHz, and preamble length of 64 symbols). For more detailed information about the available UWB settings of Pozyx see [12]. To get the published topics on the ROS node, we installed the Pozyx positioning system by placing the anchors in different locations in space and identifying their coordinates in space using the Pozyx software. We then fixed the tag on the quadrotor, to perform multilateration between the installed anchors and the tag and calculate the position of the quadrotor in space. The position is then published via the ROS node with a frequency depending on the UWB settings, the number of installed anchors and the pre-filtering strength of Pozyx. In this work, we considered two cases: (a) position measurements with average median pre-filtering and (b) no pre-filtering. The average median filter removes outliers from the position measurements by passing through the median value of the last k_{st}-sampled time window. Pozyx refer to this time window to be filtered, "filter strength". The higher the value of the filter strength, the longest the induced delays on the measurements. Based on these settings, the position is published with an average frequency of 20Hz which is also used as the estimation and control frequency.

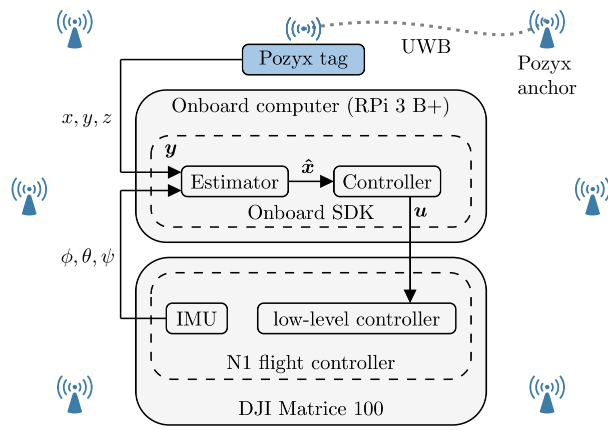

Fig. 3: Experimental setup.

For the state estimation and control of the quadrotor we developed a ROS node including the state estimators and the controller. This node calculates the state estimation based on the measurements (i.e., $\boldsymbol{y}$) and the previous control signals, and then it calculates the next control signals (i.e., $\boldsymbol{u}$) based on the current state estimate (i.e., $\hat{\boldsymbol{x}}$). When the control action is calculated, the node publishes the control signals which are then subscribed and applied to the quadrotor by the build-in low-level N1 flight controller as shown in Fig.3.

B. Experimental Results

In all experiments that follow, the mission of the quadrotor is to hover at a horizontal reference position with a fixed heading (i.e., yaw angle). The reference point used for all experiments was set at $\boldsymbol{r_k} = [r_x, r_y, r_\psi]^T$ where $r_x = 2m$, $r_y = 4m$ and $r_\psi = 0rads$. Note that the altitude is not controlled by our control technique, instead it is controlled by the embedded N1 flight controller. All experiments were done with the linear quadratic cost weighting matrices set to $Q_k = diag([q_1, q_2, \ldots, q_{10}])$ where $q_1 = q_3 = 10^{-5}$,

[1]A demonstration of the controlled DJI M100 via indoor UWB-based localization is available online: https://www.youtube.com/watch?v=I0dPAA83YNg

[2]https://github.com/dji-sdk/Onboard-SDK-ROS

while the value of the remaining elements along the diagonal are equal to 10^{-3}, and $R_k = diag([10, 10, 1])$. The process and measurement noise covariance matrices set to $V_k = diag([v_1, v_2, \ldots, v_{10}])$ where $v_1 = v_2 = v_3 = v_4 = 1$, $v_5 = v_7 = v_9 = 5$, $v_6 = v_8 = v_{10} = 2$ and $W_k = diag([500, 500, 1])$ respectively. The control inputs are limited to $|u_\phi| \leq 0.1 rads$, $|u_\theta| \leq 0.1 rads$ and $|u_{\dot{\psi}}| \leq 2 rads/s$. The initial error covariance matrix were set to $P_0 = 4I$ while the initial state vector were set to $\boldsymbol{x}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 3\ 0\ 3\ 0]^T$. Although the initial position of both x and y is at 3, due to the uncontrolled dynamics of the quadrotor for a couple of seconds at the beginning of the experiments, it slightly diverges from the initial position as shown in Fig.4 and Fig.5.

Before implementing the estimation and control algorithms, the coefficients of the dynamic model are needed to be identified as mentioned in Section IV-C. For the inputs of the N1 flight controller to be identified, we use square wave signals as they can describe sufficiently the behavior of first and second order systems. The sampling frequency of the signals is at 100Hz which corresponds to the rate that the IMU topic of the Onboard-SDK ROS node is published. The identified coefficients are presented in Table I.

roll (ϕ)	pitch (θ)	yaw rate ($\dot{\psi}$)
$\beta_{0,\phi} = 37.70$	$\beta_{0,\theta} = 36.14$	$\beta_{0,\psi} = 2.339$
$\alpha_{0,\phi} = 40.00$	$\alpha_{0,\theta} = 38.24$	$\alpha_{0,\psi} = 2.338$
$\alpha_{1,\phi} = 6.90$	$\alpha_{1,\theta} = 6.51$	

TABLE I: Dynamic model coefficients.

Fig.4 shows results using both estimators with the measurements ($\boldsymbol{y}$) to be filtered by an average median filter. Although this filter removes the outliers from Pozyx measurements by filtering the last 5 measurements, it induces delays to the measurements, due to computation time, that are then fed through the estimators. Fig.5 shows results of both estimators without any pre-filtering on the measurements. Note that, in the latter case, if the positioning system is not able to calculate the quadrotor's position due to insufficient number of anchors or increased computation time, then the published measurements will be 0. This induces impulsive shots on the measurements as shown in Fig.5 which pass through the estimators. In both cases, we disturb the measurements by covering one of the anchors around time $27s$.

As it can be seen from both Fig.4 and Fig.5, both estimators track the reference signal even though there is a disturbance over the position measurements. In Fig.4 the overshoot is induced as a result of the average median filtering of wrong measurements from the Pozyx positioning system as there are not enough sensors to compute the actual position of the quadrotor in space. The MCC-KF performs better in estimating the states than the KF, as it uses correntropy measure to deal with non-Gaussian measurement noise. In Fig.5, the Pozyx positioning system does not pre-filter the computed measurements taken from the sensors, and thus impulsive shots of noise are fed through the estimators. As it can be seen from Fig.5, the KF is influenced by these impulsive shots of noise and thus it passes wrong state estimation to the LQR controller. In contrast, the MCC-KF detects the impulsive shots by the correntropy measure, and it disregards these outliers. This leads to a filtered state estimate signal that are fed through the LQR to provide lower mean square error (mse) between estimated and reference signal as shown in Table II.

	Average median filter		No filter	
	KF	MCC-KF	KF	MCC-KF
(x, y)-plane (m)	0.066	0.057	0.107	0.048
mse decrease (%)	13.636		55.140	

TABLE II: Mean squared error (mse).

Table II presents the mean squared error (mse) between the reference point and the estimated position of the (x, y)-plane. The mse with the average median pre-filtering when deploying the KF is $0.066m$, while for the MCC-KF $0.057m$. Without pre-filtering, for the KF and the MCC-KF, the mse is $0.107m$ and $0.048m$, respectively. These results show a decreased mse by 13.636% when using the MCC-KF with the average median pre-filtering and by 55.140% without pre-filtering, compared to the KF.

VII. Conclusions and Future Directions

In this work we proposed and developed onboard state estimation using MCC-KF to estimate the orientation and horizontal translational states of a quadrotor using measurements from IMU and wireless UWB positioning sensors. The estimated states are fed through an LQR controller which stabilizes the horizontal translational dynamics even when the measurements are driven with non-Gaussian noise and NLOS conditions. Finally, we compared the MCC-KF with the traditional KF, and we showed that MCC-KF outperforms the use of KF in the presence of shots of mixed noise and communication delays.

A significant extension of this work would be to design robust estimators from IMU, UWB and vision measurements to steer robust controllers for fully autonomous and high-precision flights. Based on these setup, different missions such as rectangular and spiral path tracking, but also different configurations of the UWB wireless network could be propitious to study the influence of communication on the estimation and control.

References

[1] Z. B. Tariq, D. M. Cheema, M. Z. Kamran, and I. H. Naqvi, "Non-gps positioning systems: A survey," *ACM Computing Surveys (CSUR)*, pp. 1–34, 2017.

[2] G. Ducard and R. D'Andrea, "Autonomous Quadrotor Flight using a Vision System and Accommodating Frames Misalignment," in *IEEE International Symposium on Industrial Embedded Systems*, 2009, pp. 261–264.

[3] F. Yu, G. Chen, N. Fan, Y. Song, and L. Zhu, "Autonomous Flight Control Law for an Indoor UAV Quadrotor," in *IEEE Chinese Control And Decision Conference (CCDC)*, 2017, pp. 6767–6771.

[4] I. Sa, M. Kamel, R. Khanna, M. Popovic, J. Nieto, and R. Siegwart, "Dynamic System Identification, and Control for a Cost Effective Open-Source VTOL MAV," in *Field and Service Robotics*, 2017, pp. 605–620.

[5] I. Sa and P. Corke, "100hz Onboard Vision for Quadrotor State Estimation," in *Australasian Conference on Robotics and Automation*, 2012.

[6] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-Time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 957–964.

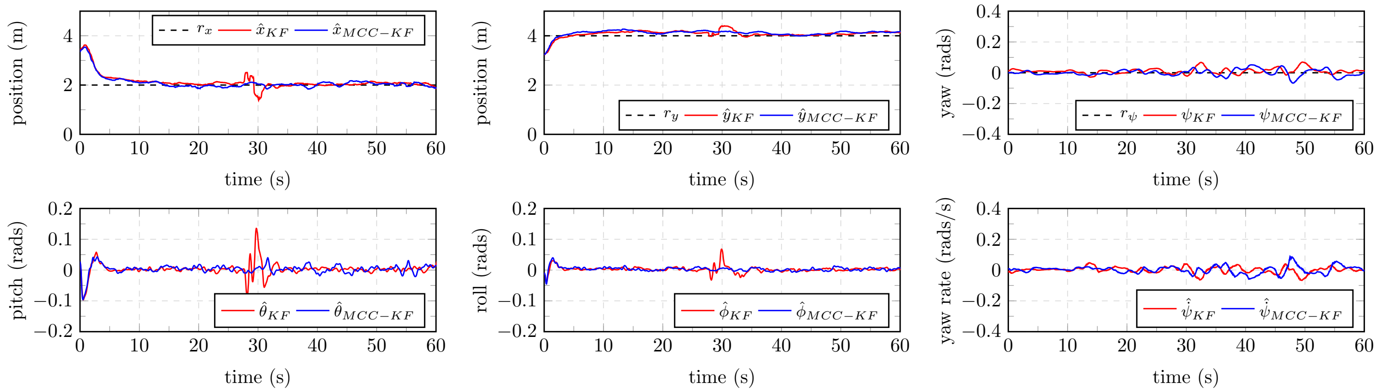

Fig. 4: Control performance with average median pre-filtering.

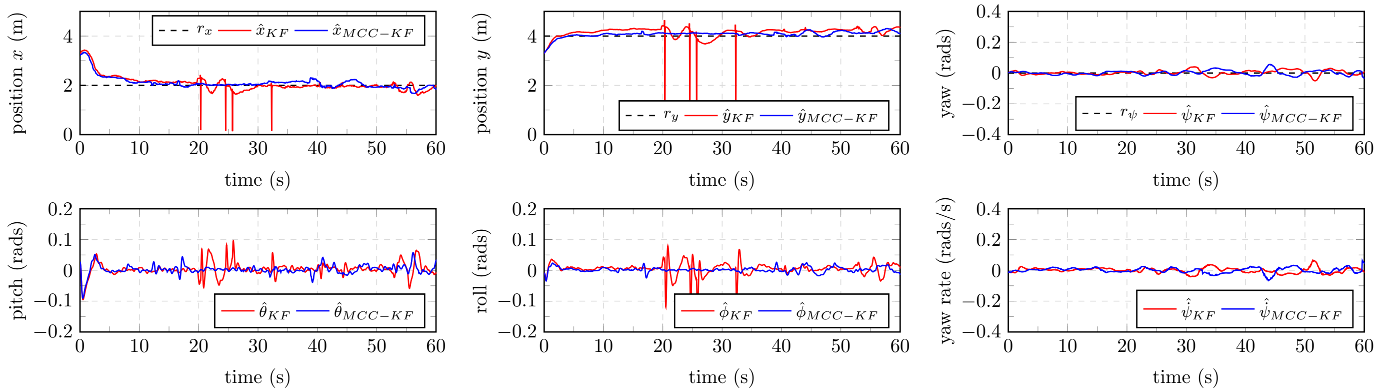

Fig. 5: Control performance without pre-filtering.

[7] A. G. Kendall, N. N. Salvapantula, and K. A. Stol, "On-board Object Tracking Control of a Quadcopter with Monocular Vision," in *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 404–411.

[8] X. Zhang, B. Xian, B. Zhao, and Y. Zhang, "Autonomous Flight Control of a Nano Quadrotor Helicopter in a GPS-denied Environment using On-board Vision," *IEEE Transactions on Industrial Electronics*, 2015, pp. 6392–6403.

[9] F. Maffra, L. Teixeira, Z. Chen, and M. Chli, "Real-Time Wide-Baseline Place Recognition Using Depth Completion," *IEEE Robotics and Automation Letters*, pp. 1525–1532, April 2019.

[10] J. Tiemann, F. Schweikowski, and C. Wietfeld, "Design of an UWB Indoor-Positioning System for UAV Navigation in GNSS-denied Environments," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–7.

[11] J. Tiemann and C. Wietfeld, "Scalable and Precise multi-UAV Indoor Navigation using TDOA-based UWB Localization," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–7.

[12] I. Papastratis, T. Charalambous, and N. Pappas, "Indoor Navigation of Quadrotors via Ultra-Wideband Wireless Technology," in *IEEE Advances in Wireless and Optical Communications (RTUWO)*, 2018, pp. 106–111.

[13] A. Ledergerber, M. Hamer, and R. D'Andrea, "A Robot Self-Localization System using One-Way Ultra-Wideband Communication," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3131–3137.

[14] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2451–2456.

[15] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and Control of an Indoor Micro Quadrotor," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 4393–4398.

[16] S. Bouabdallah and R. Y. Siegwart, "Full Control of a Quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 153–158.

[17] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An Integral Predictive/Nonlinear $\mathcal{H}_\infty$ Control Structure for a Quadrotor Helicopter," *Automatica*, 2010, pp. 29–39.

[18] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Backstepping/Nonlinear $\mathcal{H}_\infty$ Control for Path Tracking of a Quadrotor Unmanned Aerial Vehicle," in *IEEE American Control Conference*, 2008, pp. 3356–3361.

[19] I. Sa and P. Corke, "System Identification, Estimation and Control for a Cost Effective Open-Source Quadcopter," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2202–2209.

[20] P. Foehn and D. Scaramuzza, "Onboard State Dependent LQR for Agile Quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6566–6572.

[21] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano, "Nonlinear Control for Systems with Bounded Inputs: Real-Time Embedded Control Applied to UAVs," in *IEEE Conference on Decision and Control (CDC)*, 2006, pp. 5888–5893.

[22] H. Voos, "Nonlinear Control of a Quadrotor micro-UAV using Feedback-Linearization," in *IEEE International Conference on Mechatronics*, 2009, pp. 1–6.

[23] L. N. Vela, "Design and Implementation of a Development Platform for Indoor Quadrotor Flight Control," 2018.

[24] B. Chen, X. Liu, H. Zhao, and J. C. Principe, "Maximum Correntropy Kalman Filter," *Automatica*, 2017, pp. 70–77.

[25] R. Izanloo, S. A. Fakoorian, H. S. Yazdi, and D. Simon, "Kalman Filtering Based on the Maximum Correntropy Criterion in the Presence of non-Gaussian Noise," in *IEEE Annual Conference on Information Science and Systems (CISS)*, 2016, pp. 500–505.

[26] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: Properties and Applications in non-Gaussian Signal Processing," *IEEE Transactions on Signal Processing*, 2007, pp. 5286–5298.