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

Please cite the original version:
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 the position and the orientation of quadrotors, using estimation and control for horizontal translational motion and heading angle for 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 \( H_\infty \) 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 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 \( A \succeq 0 (A > 0) \) means that matrix \( A \) is semi-positive (positive) definite. The identity matrix is represented by \( I \). \( E\{\cdot\} \) represents the expectation of its argument. Given any vector norm \( \| \cdot \|_A \), a weighted vector norm can be written as \( \|x\|_A \triangleq \| Ax \|_2 \), where \( A \) is an arbitrary nonsingular matrix. The sine and cosine of an angle \( \theta \) 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 nonlinear complex system. Each rotor consists a propeller and a motor which produces an angular velocity \( \omega_i \) 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 (\( \psi \)) 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., \( \xi = (x, y, z) \in \mathbb{R}^3 \)) and rotational (i.e., \( \eta = (\phi, \theta, \psi) \in \mathbb{R}^3 \)) 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 \( \phi, \theta \) and \( \psi \) represent the orientation of the quadrotor. As it can be seen from Fig.2; \( \phi \) is the angle about the \( x \)-axis and it is called “roll”, \( \theta \) is the angle about the \( y \)-axis and is called “pitch”, and \( \psi \) 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]:

\[
m \ddot{\mathbf{r}} = f_B - f_G, \tag{1}
\]

where \( m \) denotes the mass of the quadrotor, \( f_B \) 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

\[
R \mathbf{r}_B = \mathbf{r}_E, \tag{2}
\]

which implies:

\[
m \ddot{\mathbf{r}}_E = R f_B - R f_G, \tag{3}
\]

where \( f_G \) is the gravitational force: \( f_G = m g \mathbf{e}_z \).

The rotational equations of motion in the Earth’s frame are described by the Euler’s equations:

\[
J \ddot{\mathbf{\omega}} = \sum_i f_i \mathbf{e}_i - M, \tag{4}
\]

where \( J \) is the quadrotor’s inertia matrix, \( \mathbf{\omega} \) is the angular velocity, \( f_i \) is the thrust force, \( \mathbf{e}_i \) is the unit vector and \( M \) is the matrix of the moments of the forces acting about the origin of the Body frame.

Multiplying the rotation matrix \( R \) by the angular velocity gives:

\[
R \mathbf{\omega}_B = \mathbf{\omega}_E, \tag{5}
\]

which implies:

\[
J \mathbf{\ddot{\omega}}_E = R (J R^{-1} \ddot{\mathbf{\omega}}_B) R^{-1} R f_B - R M, \tag{6}
\]

where \( J R^{-1} \) is the inertia matrix of the quadrotor in the Earth’s frame.

Finally, the control law of the quadrotor is given as:

\[
\ddot{\mathbf{r}}_E = -J^{-1} J \mathbf{\ddot{\mathbf{\omega}}} - J^{-1} J M - \mathbf{K}_p (R \mathbf{r}_E - \mathbf{r}_d), \tag{7}
\]

where \( \mathbf{K}_p \) is the feedback gain.
$R$ 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):

$$
\begin{bmatrix}
    c\phi c\theta & s\phi & -s\phi c\psi - s\phi c\phi & s\phi c\psi + c\phi c\theta & s\phi c\theta - s\phi s\psi \\
    s\psi c\theta & c\phi & s\phi c\theta c\psi - c\phi s\phi & c\phi c\psi + s\phi s\theta & c\phi s\psi - c\phi c\theta \\
    -s\phi & 0 & c\phi s\theta + s\phi c\theta & -s\phi c\theta - c\phi s\phi & c\phi c\theta \\
    s\theta & 0 & s\phi & -c\phi & 0 \\
    0 & 0 & 0 & 0 & f_T
\end{bmatrix}
$$

(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:

$$
f_B = f_T \begin{bmatrix} c\phi c\psi + s\phi s\theta & c\phi c\psi + s\phi s\theta & c\phi c\psi + s\phi s\theta \end{bmatrix}^T.
$$

(3)

The gravitational force $f_G$ acts only on the $z$-axis when hovering and is described by $f_G = [0 \ 0 \ mg]^T$, where $g$ is the acceleration of the gravity. Finally, substituting $f_G$ and $f_B$ into (1) we get the translational equation of motion which is given by:

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

(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., $f_T \simeq mg; \phi, \theta, \psi \simeq 0; s\psi \simeq \psi, s\theta \simeq \theta, s\phi \simeq \phi; c\phi = 1$). In addition, we assume the yaw angle ($\psi$) to be fixed (i.e., $\psi_0$), 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:

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

(5)

C. System Identification

For our experimental setup, we use a DJI Matrice 100 (M100) which runs a low level built-in NI 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., $u_\phi, u_\theta$ and $u_\psi$) that are given through the Onboard-SDK and the measured output signals (i.e., $\phi, \theta$ and $\psi$), we can identify the dynamical model coefficients $\alpha_{0,\phi}, \alpha_{1,\phi}, \alpha_{0,\theta}, \alpha_{1,\theta}, \alpha_{0,\psi}, \alpha_{1,\psi}, \beta_{0,\phi}$ 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):

$$\dot{\phi} = -\alpha_{0,\phi}\phi - \alpha_{1,\phi}\dot{\phi} + \beta_{0,\phi}u_\phi,$$

$$\dot{\theta} = -\alpha_{0,\theta}\theta - \alpha_{1,\theta}\dot{\theta} + \beta_{0,\theta}u_\theta,$$

$$\dot{\psi} = -\alpha_{0,\psi}\psi + \beta_{0,\psi}u_\psi.$$

(6)

V. OPTIMAL AND ROBUST CONTROL DESIGN

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

$$\dot{x}(t) = Ax(t)dt + Bu(t)dt + dw(t),$$

$$y(t) = Cx(t)dt + dv(t),$$

(7)

where $x \in \mathbb{R}^{10}$ is the system state vector (i.e., $x = [\phi \ \theta \ \theta \ \psi \ x \ y \ y^T]$), $u \in \mathbb{R}^3$ is the control input vector (i.e., $u = [u_\phi \ u_\theta \ u_\psi]^T$), $y \in \mathbb{R}^6$ is the measurement vector (i.e., $y = [\phi \ \theta \ z \ x \ y^T]$), while $w \in \mathbb{R}^{10}$ and $v \in \mathbb{R}^6$ 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:

$$x_{k+1} = \Phi x_k + \Gamma u_k + w_k,$$

$$y_k = Cx_k + v_k,$$

(8)

where the discrete equivalent system matrix $\Phi = e^{Ah}$, and the control input matrix $\Gamma = \left[ \begin{array}{cc} I & \alpha \end{array} \right] e^{Ah}$, and $h$ is the sampling period (i.e., control loop period). The measurement matrix $C \in \mathbb{R}^{6 \times 10}$. The process and measurement noise levels are represented by $w_k$ and $v_k$, respectively. Both are assumed to be white Gaussian random sequences with zero mean, with $E\{w_k\} = 0$, $E\{v_k\} = 0$, $E\{ww^T\} = W \succeq 0$, and $E\{vv^T\} = V > 0$.

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

$$P_{k|k-1} = E\{(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T\},$$

$$P_{k|k} = E\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T\}.$$  

(9)

(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]:

$$\hat{x}_{k|k-1} = \hat{x}_{k|k-1} + \Gamma u_k,$$

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

$$L_{k}^{mcc} = \frac{G_{\sigma}(\|y_k - C\hat{x}_{k|k-1}\|v_k^{-1})}{G_{\sigma}(\|\hat{x}_{k|k-1} - \hat{F}_{k}^{mcc}V_k^{-1}p_{k|k-1}^{-1}\|v_k^{-1})},$$

$$K_k = \left( P_{k|k-1}^{-1} + L_{k}^{mcc} C^T V_k^{-1} C L_{k}^{mcc} C^T V_k^{-1} \right)^{-1} L_{k}^{mcc} C^T V_k^{-1},$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}),$$

$$P_{k|k} = (I-K_k C) P_{k|k-1} (I-K_k C)^T + K_k V_k K_k^T,$$

(11)

(12)

(13)

(14)

(15)

(16)
where $G_\sigma$ is the Gaussian kernel, i.e.,

$$G_\sigma(\|x_i - y_i\|) = \exp\left(-\frac{\|x_i - y_i\|^2}{2\sigma^2}\right),$$

with kernel size $\sigma$ (representing a weighting parameter between the second and higher-order moments). Note that $L_k^{nec}$ is the minimized centorrntropy estimation cost function and $K_k$ is the Kalman gain.

B. Linear Quadratic Regulator (LQR) Control

To follow a reference signal $r_k = [r_x, r_y, r_\psi]^T$, we augment the state-space model by adding integral state, $i_{k+1} = i_k + r_k - E y_k$:

$$\begin{bmatrix} \bar{x}_{k+1} \\ \bar{i}_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi - K_k C & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{x}_{k-1} \\ \bar{i}_k \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 0 \end{bmatrix} r_k,$$

where $E \in \mathbb{R}^{3 \times 6}$ is a matrix that selects the observations to be controlled (i.e., $E = [e_1, e_2, e_3]^T$, where $e_1 = [0, 0, 0, 0, 1, 0]$, $e_2 = [0, 0, 0, 0, 0, 1]$, $e_3 = [0, 0, 1, 0, 0]$). The optimal control law $u_k^* = -L \bar{x}_k = -L^x \dot{x}_k - L^i i_k$ can be found by minimizing the linear quadratic criterion given in (18):

$$J = E \left[ \bar{x}_N^T Q_N \bar{x}_N + \sum_{k=0}^{N-1} (\bar{x}_k^T Q_k \bar{x}_k + u_k^T R_k u_k) \right],$$

where $Q_N \succeq 0$, $Q_k \succeq 0$ 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 $L = [L^x \quad L^i]$ is the state-feedback controller gain given by the solution of the Ricatti equation.

VI. EXPERIMENTAL VALIDATION

To evaluate the performance of the proposed control algorithms, we setup a real testbed. 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.

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 multitilateration 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.

B. Experimental Results

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., $y$) and the previous control signals, and then calculates the next control action (i.e., $u$) based on the current state estimate (i.e., $\dot{x}$). When the control action is calculated, the node publishes the control signals which are then subscribed and applied to the quadrotor by the onboard onboard N1 flight controller as shown in Fig.3. For 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 $r_k = [r_x, r_y, r_\psi]^T$ where $r_x = 2m$, $r_y = 4m$ and $r_\psi = 0$ rads. Note that the attitude 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 $q_1 = q_3 = 10^{-5}, q_2 = 10^{-4}$.
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 $|u_\phi| \leq 0.1 \text{rads}$, $|u_\theta| \leq 0.1 \text{rads}$ and $|u_\psi| \leq 2 \text{rads/s}$. The initial error covariance matrix were set to $P_0 = 4I$ while the initial state vector were set to $x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3 \ 0 \ 3 \ 0]^T$. 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.

<table>
<thead>
<tr>
<th>roll ($\phi$)</th>
<th>pitch ($\theta$)</th>
<th>yaw rate ($\psi$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_{0,\phi} = 37.70$</td>
<td>$\alpha_{0,\theta} = 40.00$</td>
<td>$\alpha_{0,\phi} = 6.90$</td>
</tr>
<tr>
<td>$\beta_{0,\phi} = 36.14$</td>
<td>$\alpha_{0,\theta} = 38.24$</td>
<td>$\alpha_{1,\phi} = 6.51$</td>
</tr>
</tbody>
</table>

**TABLE I: Dynamic model coefficients.**

Fig.4 shows results using both estimators with the measurements ($y$) 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.

<table>
<thead>
<tr>
<th>(x, y)-plane ($m$)</th>
<th>No filter</th>
<th>MCC-KF</th>
<th>KF</th>
<th>MCC-KF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average median filter</td>
<td>13.036</td>
<td>0.087</td>
<td>1.079</td>
<td>0.048</td>
</tr>
<tr>
<td>mse decrease (%)</td>
<td>55.140</td>
<td>55.140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE II: Mean squared error (mse).**

Throughout 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**


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

Fig. 5: Control performance without pre-filtering.


