Unmanned Aerial Vehicle Position and Attitude Estimation Using Computer Vision ⋆

Mateus R. da Silva * **Diego P. F. Correa** ** **Luiz S. Martins-Filho** ***

* *Universidade Federal do ABC, SP (e-mail: mateus.ribeiro@ufabc.edu.br)*
** *Universidade Federal do ABC, SP (e-mail: diego.ferruzzo@ufabc.edu.br)*
*** *Universidade Federal do ABC, SP (e-mail: luiz.martins@ufabc.edu.br)*

Abstract: This work proposes the computer vision application for the position and attitude estimation of an unmanned aerial vehicle (UAV) quadrotor navigating in an indoor Global Positioning System (GPS) denied environment. The system, composed of the quadrotor and one camera fixed outside the vehicle, was simulated in a 3D virtual environment. The results showed that computer vision improved the attitude and position estimation of the quadrotor.

Keywords: Unmanned Aerial Vehicle, Computer Vision, Kalman Filtering, Attitude Estimation, Position Estimation.

1. INTRODUCTION

The use of UAVs has increased in the last years due to their applications in several sectors such as agriculture, logistics, mapping and searching. Advancement in embedded systems technology allows UAVs to perform more complex and autonomous tasks, which demand more efficient control systems. However, an efficient control system requires knowledge of the system states obtained through the available sensors.

In general, the UAV autonomous navigation is based on Inertial Measurement Unit (IMU) and on GPS for vehicle attitude and position estimation. An IMU may be composed of two MEMS sensors (Micro Eletromechanical Systems), an accelerometer that measures the body linear velocity, and a gyroscope that provides angular speeds in the body frame. Nevertheless, these sensors are susceptible to errors, which accumulate over time, commonly known as drift, in addition to other random errors. One of the most common solutions is using GPS to track the vehicle position and compensate the error (Suwandi et al., 2017). However, in some indoor environments, the GPS signal is degraded, making its use impractical

Although the IMU containing gyroscope and accelerometer can estimate the roll and pitch angles (Figure 1), the integration of both does not provide the yaw angle accurately (Wilson et al., 2019). The yaw angle can be estimated using a magnetometer, a sensor capable of measuring the Earth magnetic field direction. However, this sensor can be affected by noise present in the environment (e.g., magnetic fields generated by electronic devices) (Nazarahari and Rouhani, 2021).

⋆ Partly financed by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)

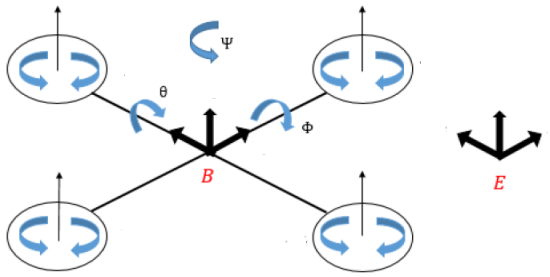

Figure 1. The angles that define the attitude of the UAV: Roll ϕ, Pitch θ and Yaw ψ angles. **Source**: (Oliveira et al., 2019)

.

In order to overcome the problem of estimating the position and yaw angle, computer vision has been included in the procedure. Despite the high computational cost required, several studies applied visual techniques to obtain system state estimation (Lee et al. (2012), Li et al. (2017), Santos and Gonçalves (2017), Botta and Quaglia (2020)) since its information is more reliable compared to MEMS sensors. As the computational power of the low-cost embedded system grows up in the last years, computer vision has become a feasible alternative when the inclusion of GPS and magnetometer are not available or have low reliability.

This work studies the application of computer vision combined with an IMU in the problem of UAV position and attitude estimation. The position estimation is performed exclusively by the camera fixed outside the vehicle, while a Multiplicative Extended Kalman Filter (MEKF) is applied to obtain the quadrotor attitude. The filter performed a sensor fusion between the camera and MEMS sensors,

simulated in a virtual environment based on Python and OpenCV (Bradski, 2000).

2. STATE ESTIMATION

2.1 Computer Vision

Among computer vision techniques, the application of fiducial markers has been widely used in several studies. Fiducial markers are geometric figures characterized by predetermined shapes and sizes (See an example in Figure 2). These markers have geometric simplicity, which facilitates their identification by the digital image processing algorithms.

Figure 2. Fiducial Marker ArUco. **Source**: (Garrido-Jurado et al., 2014)

The algorithm proposed in this work is based on the marker detection technique developed by Garrido-Jurado et al. (2014). The main goal of the detection algorithm is finding the marker location, in pixels units. Then, the algorithm searches for correspondences between the coordinates on the image plane and the points in the real world. This correspondence is called Perspective-n-Point Problem (PnP) (Eivazi Adli et al., 2020).

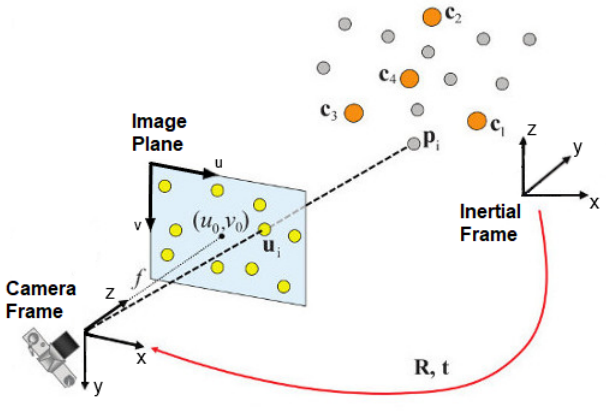

Figure 3. The coordinate systems of the estimation problem. **Source**: Adapted from Bradski (2000)

The projection of the n real points on the image plane (Figure 3), according to the pinhole camera model is defined by:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (1)$$

where $(u,\ v)$ are the image plane coordinates in 2D, $(X_W,\ Y_W,\ Z_W)$ are the real world coordinates in 3D and s is the scale factor. The first matrix represents the camera intrinsic parameters and provides information regarding the central points of the image $(c_x,\ c_y)$ and also the focal lengths, f_x and f_y (Gonzalez and Woods, 2007). These parameters are obtained through camera calibration processes, therefore it is particularly important having this information *a priori*. Then, the homogeneous transformation matrix of the camera is presented, where the terms r_{ij} refers to the rotation matrix that indicates the orientation of the camera, and the parameters t_i its translation with respect to the detected object.

The points in the marker frame can be related to the camera frame using

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (2)$$

where $(X_C,\ Y_C\ ,Z_C)$ are the camera frame coordinates.

The marker provides four known points both in the image plane and in the real world. The OpenCV has a function called *solvePnP* that returns the rotation vector $\mathbf{r}$ and the translation vector $\mathbf{t}$, both of them w.r.t the marker reference. In order to obtain the rotation matrix from $\mathbf{r}$, the Rodrigues formula is applied (Piña, 2011):

$$\mathbf{R} = \mathbf{I} + [\mathbf{r}]_x \sin\theta + \mathbf{r}^2\,(1 - \cos\theta) \quad (3)$$

where $\mathbf{R}$ is the rotation matrix, the $\theta = \sqrt{r_x^2 + r_y^2 + r_z^2}$, and $[\mathbf{r}]_x$ is the skew-matrix of $\mathbf{r}$ defined as,

$$[\mathbf{r}]_x = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

In order to find the marker coordinates w.r.t the inertial frame, a coordinate transformation has to be applied, the relation between the frames is shown in Figure 4.

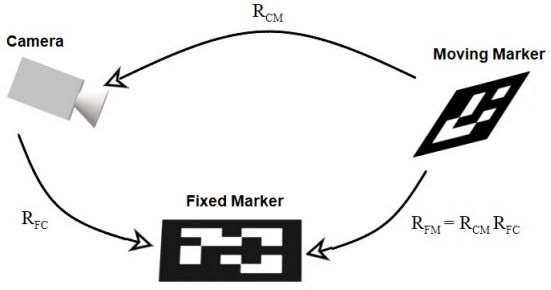

Figure 4. The coordinate transformations involved in the procedure.

In Figure 5, we present the pseudo-algorithm that describes the entire process from marker detection to its orientation w.r.t the inertial frame.

Algoritmo 1: Marker's Orientation Estimation

Entrada: Camera Image
Saída: Marker's Rotation Matrix w.r.t Inertial Frame

```
início
    Import Camera's Intrinsic Parameters
    Define Marker's Dictionary
    Define the Real Size of the Marker
fim
while Capture Image do
    Convert Image to Gray Scale;
    Detect ArUco Marker;
    if Fixed Marker ID then
        Extract 4 Image Plane Marker's Points;
        Apply solvePnP;
        Convert rvec to Rotation Matrix R_CF;
        Find the inverse of R_FC;
    end
    if Moving Marker ID then
        Extract 4 Image Plane Marker's Points;
        Apply solvePNP;
        Convert rvec to Rotation Matrix R_CM;
        if R_FC is not null then
            Find the Moving Marker's Orientation w.r.t the Inertial Frame
            R_FM = R_CM R_FC
        end
    end
end
```

Figure 5. Marker Detection and Orientation Estimation Pseudo-Algorithm.

2.2 Attitude Kinematics

In aerospace systems, it is common to represent the attitude through quaternions since it is free of singularities and has an algebra that requires low computational power. A quaternion can be defined as

$$\mathbf{q} = \begin{bmatrix}\mathbf{q}_v & q_s\end{bmatrix}^T$$

where $\mathbf{q}_v = \begin{bmatrix}q_x & q_y & q_z\end{bmatrix}^T$ is the vector component and q_s is the scalar component.

The dynamics of a quaternion can be expressed by,

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix}-[\boldsymbol{\omega}]_x & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0\end{bmatrix}$$

where $\boldsymbol{\omega}$ is the angular speed in body frame and $[\boldsymbol{\omega}]_x$ is the skew-matrix of $\boldsymbol{\omega}$.

When the sampling frequency of the angular velocity has high rate, the kinematics can be discretized employing a series of powers, thus (Crassidis and Junkins, 2012),

$$\mathbf{q}_{k+1}^{-} = \boldsymbol{\Omega}(\boldsymbol{\omega}_k^{+})\mathbf{q}_k^{+} \tag{4}$$

with

$$\overline{\boldsymbol{\Omega}}(\boldsymbol{\omega}_k^{+}) = \begin{bmatrix}\cos\left(\frac{1}{2}||\boldsymbol{\omega}_k^{+}||\Delta t\right)\mathbf{I} - [\boldsymbol{\psi}_k^{+}]_x & \boldsymbol{\psi}_k^{+} \\ -\boldsymbol{\psi}_k^{+T} & \cos\left(\frac{1}{2}||\boldsymbol{\omega}_k^{+}||\Delta t\right)\end{bmatrix}$$

$$\boldsymbol{\psi}_k^{+} = \frac{\sin\left(\frac{1}{2}||\boldsymbol{\omega}_k^{+}||\Delta t\right)\boldsymbol{\omega}_k^{+}}{||\boldsymbol{\omega}_k^{+}||}$$

where $\boldsymbol{\omega}_k^{+}$ is the gyroscope angular speed in k and Δt is the sensor sample rate. The superscript symbol '+' indicates that such variable was compute after the measurement, i.e *a posteriori*, whereas the subscript symbol '−' implies that the variable was obtained before the measurement, i.e *a priori*.

2.3 Sensors Model

MEMS gyroscopes are capable of providing the angular velocity on each axis of the sensor and can be expressed in a simplified way by the equation (Sabatini, 2011),

$$\boldsymbol{\omega}_m = \mathbf{K}_g\boldsymbol{\omega}_b + \boldsymbol{\beta}_g + \mathbf{v}_g \tag{5}$$

(6)

where $\boldsymbol{\omega}_m$ is the measured angular speed in each axis, $\mathbf{K}_g$ is the scale factor matrix, $\boldsymbol{\omega}_b$ is the angular speed vector in body frame and $\mathbf{v}_g$ is the white Gaussian noise vector. $\boldsymbol{\beta}_g$ is the drift (or bias) vector, modeled by:

$$\dot{\boldsymbol{\beta}}_g = \mathbf{v}_u \tag{7}$$

where $\mathbf{v}_u$ is a white Gaussian process noise vector.

The accelerometer measures the linear acceleration of a body and can be modeled as follow (Sabatini, 2011),

$$\mathbf{a}_m = \mathbf{K}_a\mathbf{R}_E^B\left(\mathbf{g} + \mathbf{a}_b\right) + \boldsymbol{\beta}_a + \mathbf{v}_a \tag{8}$$

where $\mathbf{a}_m$ is the linear acceleration measured, $\mathbf{K}_a$ is the scale factor matrix, $\mathbf{R}_E^B$ is the rotation matrix from inertial frame to body frame, $\mathbf{g}$ is the gravity acceleration, considered constant and given by $\begin{bmatrix}0 & 0 & -9.81\end{bmatrix}^T$, $\boldsymbol{\beta}_a$ is the bias vector and $\mathbf{v}_a$ is the white Gaussian measurement noise vector.

The measurement model given by an observation vector from a sensor is (Crassidis and Junkins, 2012),

$$\mathbf{b}_i = \mathbf{A}(\mathbf{q})\mathbf{r}_i + \mathbf{v}_i \tag{9}$$

where $\mathbf{b}_i$ is the i-th observation vector, $\mathbf{r}_i$ is the i-th reference vector, $\mathbf{v}_i$ is the i-th white Gaussian noise vector with variance σ_i, and $\mathbf{A}(\mathbf{q})$ is the rotation matrix obtained by $\mathbf{q}$,

$$\mathbf{A}(\mathbf{q}) = \left[q_s - \mathbf{q}_v^T\mathbf{q}_v\right]\mathbf{I} + 2\mathbf{q}_v\mathbf{q}_v^T - 2q_s[\mathbf{q}_v]_x$$

The measurement noise covariance matrix, in the instant k, is expressed as follow,

$$\mathbf{R}_{k,i} = \boldsymbol{\sigma}_i^2\mathbf{I} \tag{10}$$

2.4 Multiplicative Extended Kalman Filter

This work implemented an alternative version of Extended Kalman Filter (EKF) used, in general, for attitude estimation using the quaternion approach which is known as MEKF. In a classical approach, the EKF update step would violate the quaternion norm constraint. In order to clarify such a problem, an example proposed by Crassidis and Junkins (2012) is exposed. A real quaternion $\mathbf{q} = \begin{bmatrix}\sqrt{0.999} & 0 & 0 & \sqrt{0.001}\end{bmatrix}^T$ and a estimated quaternion $\hat{\mathbf{q}} = [1\ 0\ 0\ 0]^T$ are considered, where both are used in the additive EKF update step, given by the error,

$$\Delta\mathbf{q} = \hat{\mathbf{q}} - \mathbf{q}$$

Note that the operation results in a quaternion that does not have a unitary norm. To deal with this problem, the MEKF uses the definition of error between two quaternions, given by the product,

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{11}$$

where $\delta\mathbf{q}$ is the error quaternion, $\hat{\mathbf{q}}^{-1}$ is the estimated quaternion and $\mathbf{q}$ is the real quaternion.

In the MEKF, the quaternion is used to attitude propagation, whereas a tridimensional attitude vector $\boldsymbol{\alpha}$ is used as local attitude-error state in EKF. The estimation error is performed using the a priori quaternion as reference (Chang et al., 2016). The quaternion is updated as the product beetween the deviation calculated from $\boldsymbol{\alpha}$ and the estimated quaternion (Markley, 2003),

$$\mathbf{q} = \delta\mathbf{q}(\boldsymbol{\alpha}) \otimes \hat{\mathbf{q}} \tag{12}$$

where $\delta\mathbf{q}(\boldsymbol{\alpha})$ represents the rotation, parameterized by $\boldsymbol{\alpha}$, from the estimated quaternion to the real one. This operation preserves the unitary norm constraint.

The estimated states used in EKF are,

$$\Delta\tilde{\mathbf{x}} = \begin{bmatrix}\delta\boldsymbol{\alpha}^T & \Delta\boldsymbol{\beta}^T\end{bmatrix}^T \tag{13}$$

The discrete transition matrix of the EKF states can be expressed by (Crassidis and Junkins, 2012),

$$\boldsymbol{\Phi}_k = \begin{bmatrix}\boldsymbol{\Phi}_{k,1} & \boldsymbol{\Phi}_{k,2} \\ \mathbf{0} & \mathbf{I}_{3x3}\end{bmatrix} \tag{14}$$

where

$$\boldsymbol{\Phi}_{k,1} = \mathbf{I}_{3x3} - [\hat{\boldsymbol{\omega}}_k^+]_x \frac{\sin\left(||\hat{\boldsymbol{\omega}}_k^+||\Delta t\right)}{||\hat{\boldsymbol{\omega}}_k^+||} + [\hat{\boldsymbol{\omega}}_k^+]_x^2 \frac{\left[1 - \cos\left(||\hat{\boldsymbol{\omega}}_k^+||\Delta t\right)\right]}{||\hat{\boldsymbol{\omega}}_k^+||^2}$$

$$\boldsymbol{\Phi}_{k,2} = [\hat{\boldsymbol{\omega}}_k^+]_x \frac{\left[1 - \cos\left(||\hat{\boldsymbol{\omega}}_k^+||\Delta t\right)\right]}{||\hat{\boldsymbol{\omega}}_k^+||^2} - \mathbf{I}_{3x3}\Delta t - [\hat{\boldsymbol{\omega}}_k^+]_x^2 \frac{\left[||\hat{\boldsymbol{\omega}}_k^+||\Delta t - \sin\left(||\hat{\boldsymbol{\omega}}_k^+||\Delta t\right)\right]}{||\hat{\boldsymbol{\omega}}_k^+||^3}$$

Table 1. MEKF Recursive Estimation

Initialization: $\hat{\mathbf{q}}_0$, $\hat{\boldsymbol{\beta}}_0$, $\mathbf{P}_0$

Measurement and Covariance Update:

$\mathbf{h}_k(\hat{\mathbf{x}}_k^-) = \begin{bmatrix}A(\hat{\mathbf{q}}_k^-)\mathbf{r}_1 \\ A(\hat{\mathbf{q}}_k^-)\mathbf{r}_2\end{bmatrix}$

$\mathbf{H}_k(\hat{\mathbf{x}}_k^-) = \begin{bmatrix}\begin{bmatrix}A(\hat{\mathbf{q}}_k^-)\mathbf{r}_1\end{bmatrix}_x & \mathbf{0} \\ \begin{bmatrix}A(\hat{\mathbf{q}}_k^-)\mathbf{r}_2\end{bmatrix}_x & \mathbf{0}\end{bmatrix}$

$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\hat{\mathbf{x}}_k^-)\left[\mathbf{H}_k(\hat{\mathbf{x}}_k^-)\mathbf{P}_k^-\mathbf{H}_k(\hat{\mathbf{x}}_k^-) + \mathbf{R}\right]^{-1}$

$\mathbf{P}_k^+ = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k(\hat{\mathbf{x}}_k^-)\right]\mathbf{P}_k^-$

$\Delta\hat{\mathbf{x}}_k^+ = \mathbf{K}_k\left[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)\right] = \begin{bmatrix}\delta\boldsymbol{\alpha}_k^{+T} & \Delta\boldsymbol{\beta}_k^{+T}\end{bmatrix}^T$

Attitude Update:

$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{1}{2}\Omega(\delta\boldsymbol{\alpha}_k^+)\hat{\mathbf{q}}_k^-$

$\hat{\boldsymbol{\beta}}_k^+ = \hat{\boldsymbol{\beta}}_k^- + \Delta\hat{\boldsymbol{\beta}}_k^+$

Propagation:

$\hat{\boldsymbol{\omega}}_k^+ = \tilde{\boldsymbol{\omega}}_k - \hat{\boldsymbol{\beta}}_k^+$

$\hat{\boldsymbol{\beta}}_{k+1}^- = \hat{\boldsymbol{\beta}}_k^+$

$\hat{\mathbf{q}}_{k+1}^- = \boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_k^+)\hat{\mathbf{q}}_k^+$

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

where $\hat{\boldsymbol{\omega}}_k^+$ is the angular velocity estimated vector, which is determined through the angular velocity measured by the gyroscope $\tilde{\boldsymbol{\omega}}_k$ and the estimated bias $\hat{\boldsymbol{\beta}}_k$.

The discrete process noise transition matrix is,

$$\mathbf{G}_k = \begin{bmatrix}-\mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3}\end{bmatrix} \tag{15}$$

with discrete process noise covariance matrix being,

$$\mathbf{Q}_k = \begin{bmatrix}\left(\boldsymbol{\sigma}_g^2\Delta t + \frac{1}{3}{}^2_u\Delta t^3\right)\mathbf{I}_{3x3} & \left(\frac{1}{2}{}^2_u\Delta t^2\right)\mathbf{I}_{3x3} \\ \left(\frac{1}{2}{}^2_u\Delta t^2\right)\mathbf{I}_{3x3} & \left({}^2_u\Delta t\right)\mathbf{I}_{3x3}\end{bmatrix}$$

where $\boldsymbol{\sigma}_g^2$ and $\boldsymbol{\sigma}_u^2$ are the noise covariances of $\mathbf{v}_g$ and $\mathbf{v}_u$, respectively. The standard deviations of the sensors were obtained by collecting 3000 samples of the measurements when they were at rest. The values are shown in Table 2.

The recursive estimation process of MEKF is summarized in Table 1.

2.5 Simulation

The proposed system consists of a quadrotor and a camera fixed somewhere in the environment (Figure 6). Both were implemented virtually based on the work of Costa Fernandes et al. (2020). The camera is fixed in $\mathbf{p} = (0, -2.1, 7.5)\ m$, and rotated 30° around y-axis in camera's frame, considering that the standard orientation of the camera is facing down. The camera calibration process followed the same steps described by Costa Fernandes et al. (2020).

The position of the quadrotor is obtained directly by the camera measurements, in meters. In order to adjust the measurements, calibration curves were obtained using linear regression.

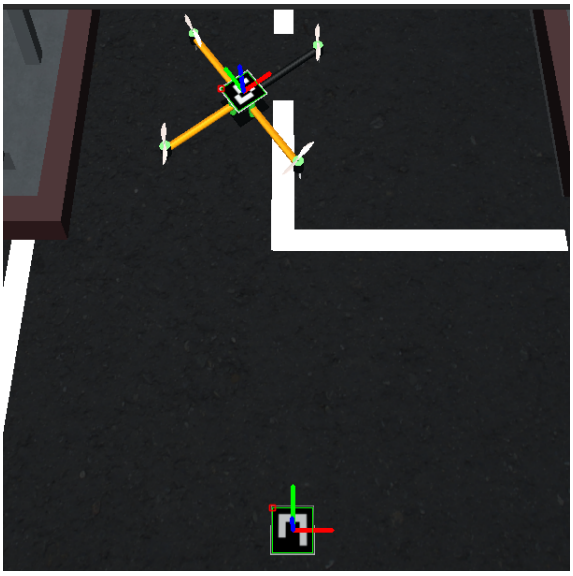

Figure 6. Graphical representation of the simulation environment.

$$x_{est} = 0.868t_x - 2.52$$
$$y_{est} = 0.978t_y - 0.0563$$
$$z_{est} = 0.932t_z + 0.725$$

where t_x, t_y and t_z are the measured quadrotor position provided by the camera, x_{est}, y_{est} and z_{est} are the adjusted quadrotor position in inertial frame.

On the other hand, the attitude is obtained through MEKF using camera and MEMS sensors data. The observation vectors considered are

$$\mathbf{y}_a = \frac{\mathbf{a}_m}{||\mathbf{a}_m||}$$
$$\mathbf{y}_c = \mathbf{R}_{MF}\mathbf{c}_x$$

where $\mathbf{y}_a$ is the normalized accelerometer vector, as described in (8). The camera observation vector, $\mathbf{y}_c$, is determined by the product between the unit vector $\mathbf{c}_x = [1\ 0\ 0]^T$ and the rotation matrix $\mathbf{R}_{MF}$, which is determined using the Algorithm 1, note that $\mathbf{R}_{MF} = \mathbf{R}_{FM}^T$.

The estimated observation vectors are computed as described in (9), where the references vectors are defined as follow,

$$\mathbf{r}_a = [0\ 0\ -1]^T$$
$$\mathbf{r}_c = [1\ 0\ 0]^T$$

where $\mathbf{r}_a$ is the accelerometer reference vector and $\mathbf{r}_c$ is the camera reference vector.

The numerical simulations consider four cases in which the movement of the quadrotor starts in an initial state and is controlled to keep the UAV hovering $5m$ above the ground w.r.t the inertial frame origin. The adopted control strategy is the Linear Quadratic Regulator (LQR) as in Costa Fernandes et al. (2020). The initial states are defined by

Table 2. The parameters of the estimation system

Accel. Standard Deviation	σ_a	$3.7 \times 10^{-3}(G)$
Camera Standard Deviation	σ_c	$2.6 \times 10^{-3}(unit)$
Gyroscope Standard Deviation	σ_g	$3.5 \times 10^{-2}(rad/s)$
Bias Standard Deviation	σ_u	$1.5 \times 10^{-4}(rad/s^2)$
MEMS Sensor Frequency	f_{MEMS}	$100(Hz)$
Cameras Frequency	f_{cam}	$10(Hz)$
Simulation Time	T_s	$6(s)$

$$\mathbf{x}_0 = [X\ Y\ Z\ q_s\ q_x\ q_y\ q_z]^T$$

where X, Y e Z are the quadrotor position states in inertial frame.

The configuration of each case is described as follow:

- **Case 1:** Attitude estimation using camera and MEMS sensors, with:
  $\mathbf{x}_0 = [-0.3\ 1.0\ 4.0\ 0.775\ -0.342\ -0.092\ 0.525]^T$
- **Case 2:** Attitude estimation using only MEMS sensors. The initial states are the same as in Case 1.
- **Case 3:** Attitude estimation using camera and MEMS sensors, with:
  $\mathbf{x}_0 = [0.5\ 0.3\ 4.5\ 0.831\ -0.212\ 0.227\ -0.462]^T$
- **Case 4:** Attitude estimation using only MEMS sensors. The initial states are the same as in Case 3.

The initial covariance matrix considered is,

$$\mathbf{P}_0 = \mathbf{I}_{6x6} \times 10^3$$

3. RESULTS

In this section, the results obtained through numerical simulations in the different scenarios, cases 1 to 4, are presented. The simulations parameters are shown in Table 2. The camera data acquisition rate is 10 Hz, since such value is a reasonable update rate for real applications proposes.

Table 3 shows the Root Mean Square Error (RMSE) of the position estimation of the quadrotor. It can be seen that the measurements error is on the order of centimeters. This result is satisfactory for simple navigation scenarios which do not require high position accuracy. Note that the RMSE of the measures Y and Z showed higher values in Cases 1 and 2. This result can be explained by the fact that the quadrotor is hovering at a lower height than in cases 3 and 4, then the marker fixed on the UAV become smaller on the image plane. Because of that, the detection presents less accuracy causing fluctuations in measurements.

Table 3. RMSE Camera Position Estimation

Case	Position (cm)		
	X	Y	Z
1	1.749	2.877	4.045
2	1.745	2.877	4.045
3	2.469	2.638	2.345
4	2.469	2.639	2.345

Figure 7 shows the trajectory of the real and estimated position states of the quadrotor. Only the curves of cases 1 and 3 are plotted since there are no significant changes concerning the others cases, as exposed in Table 3.

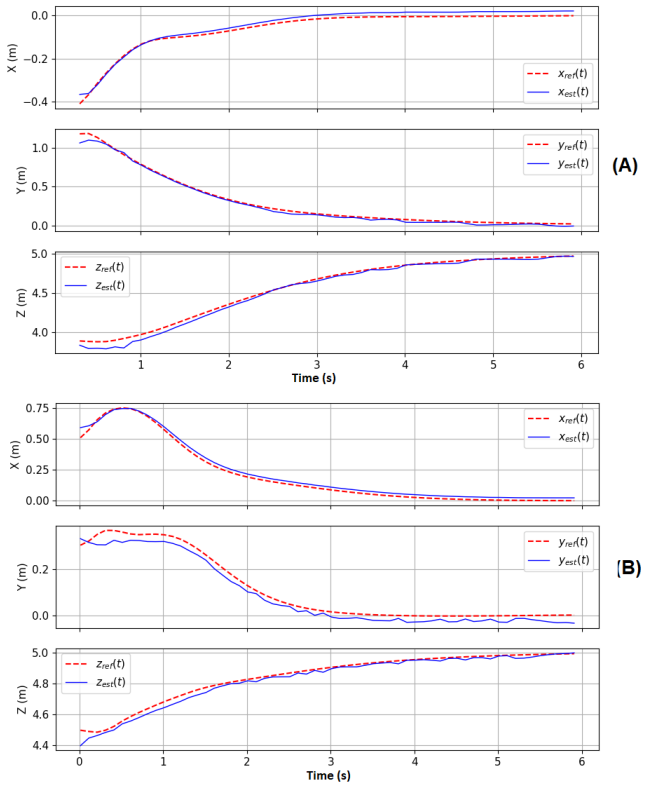

Figure 7. Quadrotor Position - (A) Case 1: with camera ; (B) Case 3: without camera

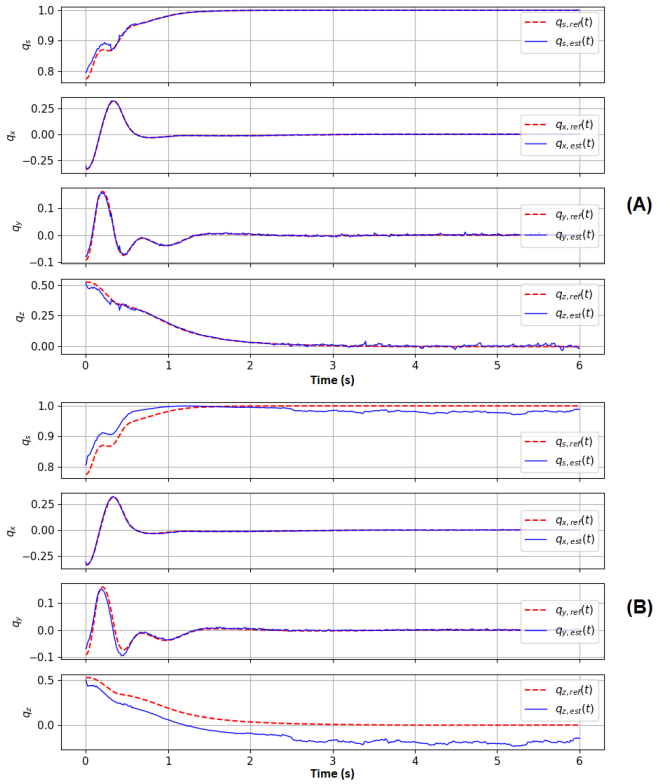

Figure 8. Quadrotor Attitude - (A) Case 1: with camera; (B) Case 2: without camera

The real and estimated attitude curves of the quadrotor are depicted in Figures 8 and 9. The attitude estimation results are better when were used both camera and MEMS sensors in MEKF. This estimation case precision is improved because accelerometer can estimate only rotations around the x and y axes, therefore the yaw angle suffers drift, as shown in the Figures 8B and 9B.

From the obtained results it is possible to observe that the camera acted as a satisfactory reference sensor, therefore, it is an adequate alternative to the magnetometer.

Table 4. RMSE Results of MKEF Attitude Estimation

Case	Quaternion (10^{-3})			
	q_s	q_x	q_y	q_z
1	5.064	2.604	3.624	12.34
2	20.59	3.627	7.720	168.2
3	12.26	2.627	3.453	20.71
4	26.33	4.437	4.377	205.8

The RMSE attitude values found are consistent with the data shown in the curves of Figures 8 and 9 since the RMSE of the quaternions q_s and, mainly, q_z, obtained from Case 2, present high values concerning the others (See Table 4). This is due to the drift angle of the yaw angle of the quadrotor, described above.

4. CONCLUSION

The use of computer vision proved to be an alternative for estimating the translational and rotational states of the UAV. The position measured by the camera provide an adequate precision for simple navigation of a quadrotor in indoor environment. Besides, the attitude estimation performed by MEKF demonstrated that computer vision prevented the drift of the yaw angle. Therefore, the use of computer vision proved to be an alternative to the use of GPS to obtain the position, and also to the magnetometer for observation of the yaw angle.

Further works will include simulations in ROS/Gazebo environment, which is widely used by the scientific community. Moreover, experimental implementations and tests can be done to verify other aspects of the proposed solution, such as observation noises, processing time, and data communication in a real-time system.

ACKNOWLEDGMENTS

This study was partly financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001

REFERENCES

Botta, A. and Quaglia, G. (2020). Performance Analysis of Low-Cost Tracking System for Mobile Robots. *Machines*, 8(2), 29. doi:10.3390/machines8020029.

Bradski, G. (2000). The OpenCV Library.

Chang, L., Hu, B., and Li, K. (2016). Iterated multiplicative extended kalman filter for attitude estimation using vector observations. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4), 2053–2060. doi:10.1109/TAES.2016.150237.

Costa Fernandes, R., Silva, P.S., Ieda Fazanaro, F., and Ferruzzo Correa, D.P. (2020). Attitude and Position Estimation in UAVs using Artificial Landmarks and

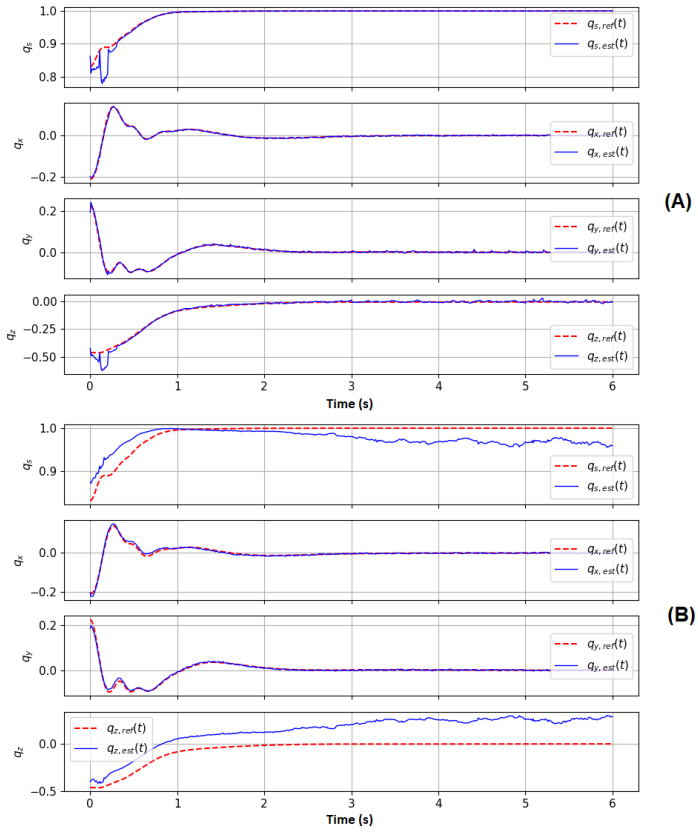

Figure 9. Quadrotor Attitude - (A) Case 3: with camera; (B) Case 4: without camera

MEMS Sensors in a Virtual Environment. doi:10.48011/asba.v2i1.1656.

Crassidis, J.L. and Junkins, J. (2012). *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, second edition.

Eivazi Adli, S., Shoaran, M., and Sayyed Noorani, S.M. (2020). GSPnP: simple and geometric solution for PnP problem. *The Visual Computer*, 36(8), 1549–1557. doi:10.1007/s00371-019-01747-x.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., and Marín-Jiménez, M.J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2280–2292. doi:10.1016/j.patcog.2014.01.005.

Gonzalez, R.C. and Woods, R.E. (2007). *Digital Image Processing*. Pearson, third edition.

Lee, D., Ryan, T., and Kim, H.J. (2012). Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In *Proceedings - IEEE International Conference on Robotics and Automation*, 971–976. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICRA.2012.6224828.

Li, J., Besada, J.A., Bernardos, A.M., Tarrío, P., and Casar, J.R. (2017). A novel system for object pose estimation using fused vision and inertial data. *Information Fusion*, 33, 15–28. doi:10.1016/j.inffus.2016.04.006.

Markley, F.L. (2003). Attitude Error Representations for Kalman Filtering. *Journal of Guidance, Control, and Dynamics*, 26(2), 311–317. doi:10.2514/2.5048.

Nazarahari, M. and Rouhani, H. (2021). 40 years of sensor fusion for orientation tracking via magnetic and inertial measurement units: Methods, lessons learned, and future challenges. *Information Fusion*, 68, 67–84. doi:10.1016/j.inffus.2020.10.018.

Oliveira, A.C.F., Altuna, J.A.T., and Correa, D.P.F. (2019). Dynamic Modelling And Control Of Unmanned Aerial Vehicle Of The Quadrotor Type. In *Proceedings of the 25th International Congress of Mechanical Engineering*. ABCM, Uberlândia, MG. doi:10.26678/ABCM.COBEM2019.COB2019-0695.

Piña, E. (2011). Rotations with Rodrigues' vector. *European Journal of Physics*, 32(5), 1171–1178. doi:10.1088/0143-0807/32/5/005.

Sabatini, A.M. (2011). Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation. *Sensors*, 11(10), 9182–9206. doi:10.3390/s111009182.

Santos, D.A. and Gonçalves, P.F.S.M. (2017). Attitude Determination of Multirotor Aerial Vehicles Using Camera Vector Measurements. *Journal of Intelligent Robotic Systems*, 86(1), 139–149. doi:10.1007/s10846-016-0418-0.

Suwandi, B., Kitasuka, T., and Aritsugi, M. (2017). Low-cost IMU and GPS fusion strategy for apron vehicle positioning. In *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, volume 2017-Decem, 449–454. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/TENCON.2017.8227906.

Wilson, S., Eberle, H., Hayashi, Y., Madgwick, S.O., McGregor, A., Jing, X., and Vaidyanathan, R. (2019). Formulation of a new gradient descent MARG orientation algorithm: Case study on robot teleoperation. *Mechanical Systems and Signal Processing*, 130, 183–200. doi:10.1016/j.ymssp.2019.04.064.