Orientation estimation fusing a downward looking camera and inertial sensors for a hovering UAV

Gastón Araguás Claudio Paz David Gaydou Gonzalo Perez Paina

Research Center in Informatics for Engineering (CIII)
National Technological University, Córdoba Faculty, Argentina
http://ciii.frc.utn.edu.ar

16*th* International Conference on Advanced Robotics
School of Engineering
Universidad de la República del Uruguay
November 25-29

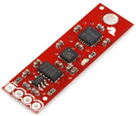
Outline

Introduction and objectives

Unmanned Aerial Vehicles (UAV) - Quadrotors

- Advantages: inexpensives, easy to build and to maintain, lightweight and easy to control.
- Disadvantages: very limited payload, and computational resources.

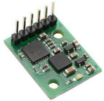
Some applications like inspection or indoor navigation require stationary fly or hovering $\longrightarrow$ orientation estimation.



Inertial sensors

Camera

Sonar



Magnetic compass

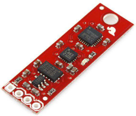
GPS

Introduction and objectives

Unmanned Aerial Vehicles (UAV) - Quadrotors

- Advantages: inexpensives, easy to build and to maintain, lightweight and easy to control.
- Disadvantages: very limited payload, and computational resources.

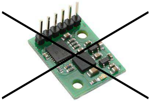
Some applications like inspection or indoor navigation require stationary fly or hovering $\longrightarrow$ orientation estimation.



Inertial sensors

Camera

Sonar



Magnetic compass

GPS

Introduction and objectives

Unmanned Aerial Vehicles (UAV) - Quadrotors

- Advantages: inexpensives, easy to build and to maintain, lightweight and easy to control.
- Disadvantages: very limited payload, and computational resources.

Some applications like inspection or indoor navigation require stationary fly or hovering ⟶ orientation estimation.

Inertial sensors | Camera | Sonar | Magnetic compass | GPS

Objectives

Orientation estimation algorithm fusing spectral features based visual yaw angle estimation, and inertial measurements with an EKF (double correction stage).

Visual yaw angle estimation - Rotation estimation

A 3D scene point $\boldsymbol{M}=\begin{bmatrix}X & Y & Z & 1\end{bmatrix}^T \in \mathbb{P}^3$ is projected to the point $\boldsymbol{m}=\begin{bmatrix}u & v & 1\end{bmatrix}^T \in \mathbb{P}^2$ on the image plane, as (pin-hole model)

$$s\boldsymbol{m} = P\boldsymbol{M} = K[R|\boldsymbol{t}]\boldsymbol{M}$$

where $(R, \boldsymbol{t})$ relates the WCS and the CCS.

Visual yaw angle estimation - Rotation estimation

A 3D scene point $\boldsymbol{M}=\left[X\ Y\ Z\ 1\right]^T \in \mathbb{P}^3$ is projected to the point $\boldsymbol{m}=\left[u\ v\ 1\right]^T \in \mathbb{P}^2$ on the image plane, as (pin-hole model)

$$s\boldsymbol{m} = P\boldsymbol{M} = K[R|\boldsymbol{t}]\boldsymbol{M}$$

where $(R, \boldsymbol{t})$ relates the WCS and the CCS. For a downward looking camera and planar scene $(Z = 0)$

$$s\boldsymbol{m} = K[R|\boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K\left[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}\right]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},$$

or

$$s\boldsymbol{m} = H\boldsymbol{M},$$

with

$$H = K\left[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}\right].$$

Visual yaw angle estimation - Rotation estimation

A 3D scene point $\boldsymbol{M} = \left[X \ Y \ Z \ 1\right]^T \in \mathbb{P}^3$ is projected to the point $\boldsymbol{m} = \left[u \ v \ 1\right]^T \in \mathbb{P}^2$ on the image plane, as (pin-hole model)

$$s\boldsymbol{m} = P\boldsymbol{M} = K[R|\boldsymbol{t}]\boldsymbol{M}$$

where $(R, \boldsymbol{t})$ relates the WCS and the CCS. For a downward looking camera and planar scene $(Z = 0)$

$$s\boldsymbol{m} = K[R|\boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K\left[\boldsymbol{r}_1 \ \boldsymbol{r}_2 \ \boldsymbol{t}\right]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},$$

or

$$s\boldsymbol{m} = H\boldsymbol{M},$$

with

$$H = K\left[\boldsymbol{r}_1 \ \boldsymbol{r}_2 \ \boldsymbol{t}\right].$$

For a moving camera, at time t_A

$$s_A\boldsymbol{m}_A = H_{WA}\boldsymbol{M},$$

for the next camera frame, at time t_B

$$s_B\boldsymbol{m}_B \approx H_{WB}\boldsymbol{M}.$$

For smooth motion $s_A \approx s_B$

$$\boldsymbol{m}_A \approx H_{BA}\boldsymbol{m}_B,$$

with $H_{BA} = (H_{WB})^{-1}H_{WA}$ the homography that relates corresponding points $\boldsymbol{m}_A \leftrightarrow \boldsymbol{m}_B$.

Visual yaw angle estimation - Rotation estimation

A 3D scene point $\boldsymbol{M} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \in \mathbb{P}^3$ is projected to the point $\boldsymbol{m} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T \in \mathbb{P}^2$ on the image plane, as (pin-hole model)

$$s\boldsymbol{m} = P\boldsymbol{M} = K[R|\boldsymbol{t}]\boldsymbol{M}$$

where $(R, \boldsymbol{t})$ relates the WCS and the CCS. For a downward looking camera and planar scene $(Z = 0)$

$$s\boldsymbol{m} = K[R|\boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K\begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},$$

or

$$s\boldsymbol{m} = H\boldsymbol{M},$$

with

$$H = K\begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{bmatrix}.$$

For a moving camera, at time t_A

$$s_A\boldsymbol{m}_A = H_{WA}\boldsymbol{M},$$

for the next camera frame, at time t_B

$$s_B\boldsymbol{m}_B \approx H_{WB}\boldsymbol{M}.$$

For smooth motion $s_A \approx s_B$

$$\boldsymbol{m}_A \approx H_{BA}\boldsymbol{m}_B,$$

with $H_{BA} = (H_{WB})^{-1}H_{WA}$ the homography that relates corresponding points $\boldsymbol{m}_A \leftrightarrow \boldsymbol{m}_B$.
Neglecting the roll and pitch angle

$$\boldsymbol{m}_A = H_{BA}\boldsymbol{m}_B \approx \begin{bmatrix} R_z & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}\boldsymbol{m}_B$$

$\longrightarrow$ Euclidean transformation.

Visual yaw angle estimation - Spectral features

Phase Correlation Method

Given two images i_A and i_B differing only in a displacement (u, v), such as

$$i_A(x, y) = i_B(x - u, y - v),$$

their Fourier transform are related by

$$I_a(\omega_x, \omega_y) = e^{-j(u\omega_x + v\omega_y)} I_b(\omega_x, \omega_y).$$

The cross power spectrum (CPS) is defined as

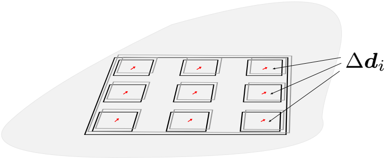
$$\frac{F(\omega_x, \omega_y) G^*(\omega_x, \omega_y)}{|F(\omega_x, \omega_y)||G^*(\omega_x, \omega_y)|},$$

where G^* is the complex conjugate of G.

$$Q(\omega_x, \omega_y) = \frac{I_a(\omega_x, \omega_y) I_b^*(\omega_x, \omega_y)}{|I_a(\omega_x, \omega_y)||I_b^*(\omega_x, \omega_y)|} = e^{-j(u\omega_x + v\omega_y)},$$

where $Q(\omega_x, \omega_y)$ is the correlation phase matrix, and the inverse transform is an impulse located in (u, v)

$$\mathcal{F}^{-1}[Q(\omega_x, \omega_y)] = q(x, y) = \delta(x - u, y - v).$$



Orientation estimation - Orientation representation

The orientation estimation is performed using the EKF and quaterion orientation representation $\boldsymbol{q} = \left[q_0 \ q_1 \ q_2 \ q_3\right]^T$ with $||\boldsymbol{q}|| = 1$.

Orientation estimation - Orientation representation

The orientation estimation is performed using the EKF and quaterion orientation representation $\boldsymbol{q} = \left[q_0 \ q_1 \ q_2 \ q_3\right]^T$ with $||\boldsymbol{q}|| = 1$.

Given the angular velocity vector $\boldsymbol{\omega} = \left[\omega_x \ \omega_y \ \omega_z\right]^T$, changes in orientation can be expressed by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \times \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix},$$

which represents the time derivative of the orientation quaternion, also written as

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

Orientation estimation - Orientation representation

The orientation estimation is performed using the EKF and quaterion orientation representation $\boldsymbol{q} = \begin{bmatrix} q_0 \ q_1 \ q_2 \ q_3 \end{bmatrix}^T$ with $||\boldsymbol{q}|| = 1$.

Given the angular velocity vector $\boldsymbol{\omega} = \begin{bmatrix} \omega_x \ \omega_y \ \omega_z \end{bmatrix}^T$, changes in orientation can be expressed by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \times \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix},$$

which represents the time derivative of the orientation quaternion, also written as

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

where $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is the skew-symmetric matrix associated to the vector $\boldsymbol{\omega}$, such that

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \boldsymbol{q}.$$

Orientation estimation - Camera/IMU fusion

Double correction stage EKF

- Prediction: gyroscope measurements.
- Correction:
  - accelerometers measurements,
  - visual yaw angle estimation.

Orientation estimation - Camera/IMU fusion

Double correction stage EKF

- Prediction: gyroscope measurements.
- Correction:
  - accelerometers measurements,
  - visual yaw angle estimation.

Extended Kalman Filter

Prediction:

$$\hat{\boldsymbol{x}}_k^- = F_{k-1}\hat{\boldsymbol{x}}_{k-1}$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

Correction:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^- + K_k(\boldsymbol{z}_k - h_k(\hat{\boldsymbol{x}}_k^-))$$

$$P_k = (I - K_k H_k)P_k^-$$

- Prediction:

$$F_k = I + \frac{\Delta t}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})$$

Orientation estimation - Camera/IMU fusion

Double correction stage EKF

- Prediction: gyroscope measurements.
- Correction:
  - accelerometers measurements,
  - visual yaw angle estimation.

Extended Kalman Filter

Prediction:

$$\hat{\boldsymbol{x}}_k^- = F_{k-1}\hat{\boldsymbol{x}}_{k-1}$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

Correction:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^- + K_k(\boldsymbol{z}_k - h_k(\hat{\boldsymbol{x}}_k^-))$$

$$P_k = (I - K_k H_k)P_k^-$$

- Prediction:

$$F_k = I + \frac{\Delta t}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})$$

- First correction stage:

$$h_{a\,k}(\boldsymbol{q}) = g \begin{bmatrix} 2q_1q_3 - 2q_2q_0 \\ 2q_2q_3 + 2q_1q_0 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

with $\boldsymbol{z}_a = \begin{bmatrix} a_x \; a_y \; a_z \end{bmatrix}^T$.

Orientation estimation - Camera/IMU fusion

Double correction stage EKF

- Prediction: gyroscope measurements.
- Correction:
  - accelerometers measurements,
  - visual yaw angle estimation.

Extended Kalman Filter

Prediction:

$$\hat{\boldsymbol{x}}_k^- = F_{k-1}\hat{\boldsymbol{x}}_{k-1}$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

Correction:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^- + K_k(\boldsymbol{z}_k - h_k(\hat{\boldsymbol{x}}_k^-))$$

$$P_k = (I - K_k H_k)P_k^-$$

- Prediction:

$$F_k = I + \frac{\Delta t}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})$$

- First correction stage:

$$h_{a\,k}(\boldsymbol{q}) = g \begin{bmatrix} 2q_1q_3 - 2q_2q_0 \\ 2q_2q_3 + 2q_1q_0 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

with $\boldsymbol{z}_a = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$.

- Second correction stage:

$$\psi = arctan\,(\gamma)$$

where $\gamma = \dfrac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}$,

Orientation estimation - Camera/IMU fusion

Double correction stage EKF

- Prediction: gyroscope measurements.
- Correction:
  - accelerometers measurements,
  - visual yaw angle estimation.

Extended Kalman Filter

Prediction:

$$\hat{\boldsymbol{x}}_k^- = F_{k-1}\hat{\boldsymbol{x}}_{k-1}$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

Correction:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^- + K_k(\boldsymbol{z}_k - h_k(\hat{\boldsymbol{x}}_k^-))$$

$$P_k = (I - K_k H_k)P_k^-$$

- Prediction:

$$F_k = I + \frac{\Delta t}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})$$

- First correction stage:

$$h_{a\,k}(\boldsymbol{q}) = g\begin{bmatrix} 2q_1q_3 - 2q_2q_0 \\ 2q_2q_3 + 2q_1q_0 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

with $\boldsymbol{z}_a = \left[a_x \; a_y \; a_z\right]^T$.

- Second correction stage:

$$\psi = arctan\,(\gamma)$$

where $\gamma = \frac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}$,

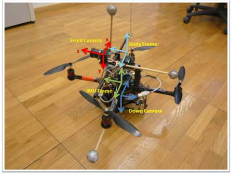
with $\boldsymbol{z}_h = \left[\cos\psi \; \sin\psi\right]^T$,

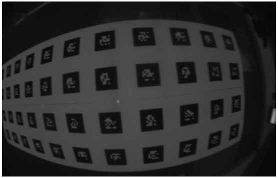
and

$$h_{h\,k}(\boldsymbol{q}) = \begin{bmatrix} \frac{1}{\sqrt{\gamma^2+1}} \\ \frac{\gamma}{\sqrt{\gamma^2+1}} \end{bmatrix}$$

Implementation and results

- OpenCV library for image processing and computer vision algorithm.
- MAV dataset of the sFly project [1], containing
  - Image sequence obtained by a forward and a downward looking camera.
  - Measurements from an inertial Measurement Unit (IMU).
  - Ground thruth information given by a Vicon system.

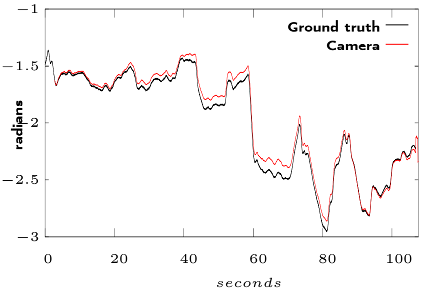




[1] http://www.sfly.org

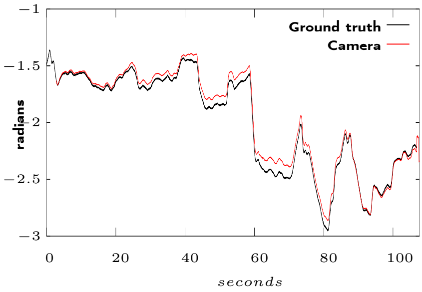
Implementation and results - Visual yaw angle estimation

Estimation for a short fly

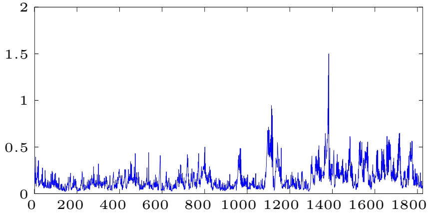


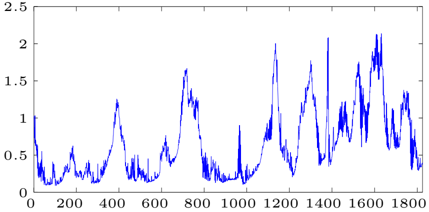
Implementation and results - Visual yaw angle estimation

Estimation for a short fly

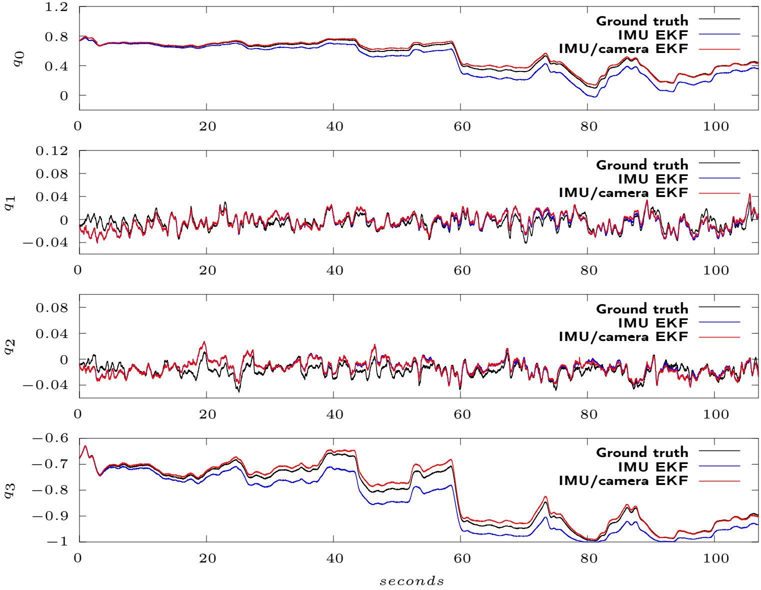


Mean accumulated error ε_{10} and ε_{100} (degrees vs. frame num.)

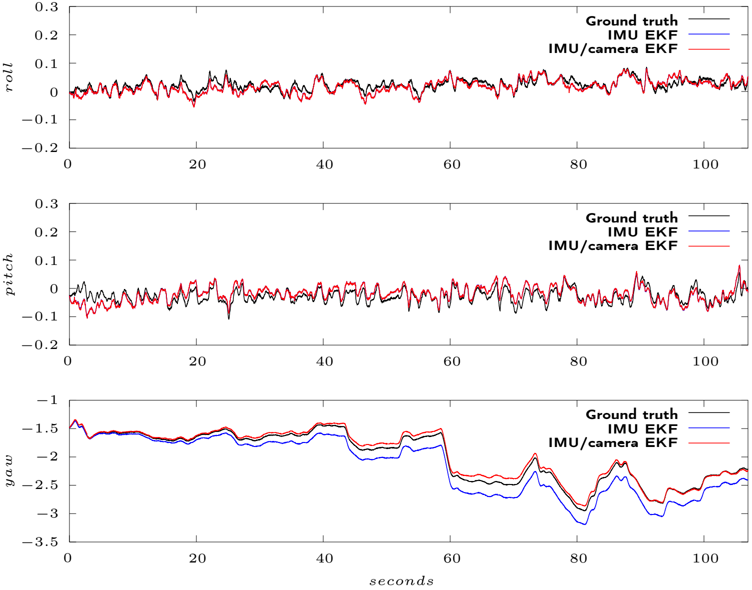




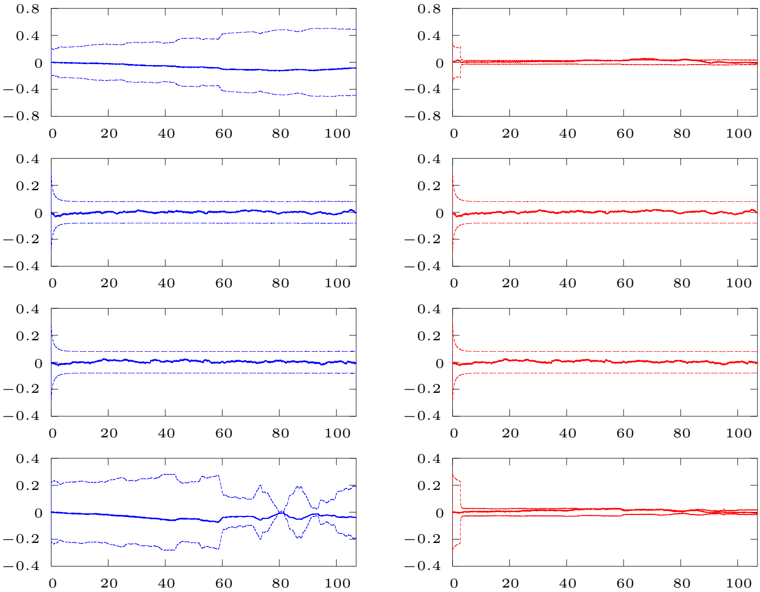
Implementation and results - Estimated quaternion



Implementation and results - Estimated Euler angles



Implementation and results - IMU only vs. Camera/IMU

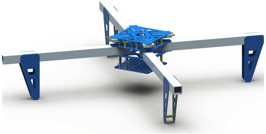


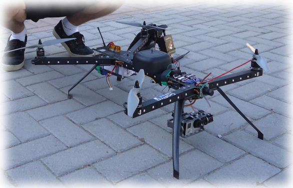
Conclusion and future work

- New approach for quadrotor orientation estimation fusing inertial measurements with a downward looking camera.
  - Inertial measurements are mainly for roll and pitch angles estimation,
  - and yaw angle is estimated by the camera using spectral features.
- Measurement fusing is based on a double correction stage EKF.
- Experimental results have been obtained using a public dataset of a hovering UAV.
- Even thought the visual yaw angle estimation has the typically accumulated error, it can be used to reduce the IMU drift.
- Camera-IMU orientation fusion presents a significantly reduction in both the bias and drift compared with the IMU only orientation estimation.
- Future work includes the estimation of the position (pose), and the implementation in a real setup.

Orientation estimation fusing a downward looking camera and inertial sensors for a hovering UAV

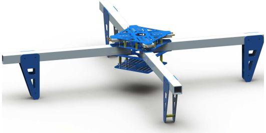
Gastón Araguás
Gonzalo F. Perez Paina
{garaguas,gperez}@scdt.frc.utn.edu.ar

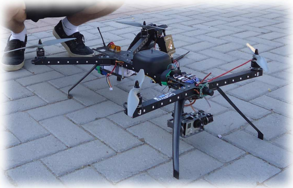




Orientation estimation fusing a downward looking camera and inertial sensors for a hovering UAV

Gastón Araguás
Gonzalo F. Perez Paina
{garaguas,gperez}@scdt.frc.utn.edu.ar





Thanks for your attention.