VISION-BASED POSITION ESTIMATION IN MULTIPLE QUADROTOR SYSTEMS WITH APPLICATION TO FAULT DETECTION AND RECONFIGURATION

MASTER THESIS, 2012-2013
SCHOOL OF ENGINEERS, UNIVERSITY OF SEVILLE

Author
Alejandro Suárez Fernández-Miranda

Supervising Teachers
Dr. Guillermo Heredia Benot
Dr. Aníbal Ollero Baturone
A strong man doesn’t need to read the future,
he makes his own.

Solid Snake - Metal Gear Solid
Table of Content

1. INTRODUCTION ........................................................................................................... 10
  1.1. Introduction ........................................................................................................... 10
  1.2. General description of the project ........................................................................ 11
  1.3. Related works ....................................................................................................... 12
      1.3.1. Vision-based position estimation ................................................................. 12
      1.3.2. Visual tracking algorithms ........................................................................... 12
      1.3.3. FDIR ........................................................................................................... 13
      1.3.4. Quadrotor dynamic modeling and control ................................................. 13
  1.4. Development time estimation ................................................................................ 14
2. Vision-based position estimation in multiple quadrotor systems .................................. 17
  2.1. Problem description .............................................................................................. 17
  2.2. Model of the system ............................................................................................. 19
  2.3. Position estimation algorithm .............................................................................. 21
  2.4. Visual tracking algorithm .................................................................................... 23
  2.5. Experimental results ............................................................................................ 25
      2.5.1. Software implementation ............................................................................ 25
      2.5.2. Description of the experiments ................................................................. 27
      2.5.3. Analysis of the results ................................................................................ 28
          2.5.3.1. Fixed quadrotor with parallel cameras ............................................. 29
          2.5.3.2. Fixed quadrotor with orthogonal camera configuration ................. 30
          2.5.3.3. Fixed quadrotor with moving cameras ............................................ 31
          2.5.3.4. Fixed quadrotor with orthogonal camera configuration and tracking loss 33
          2.5.3.5. Z-axis estimation with flying quadrotor and parallel camera configuration ...................................................................................................................... 34
          2.5.3.6. Depth and lateral motion in the XY plane ......................................... 37
          2.5.3.7. Quadrotor executing circular and random trajectories ...................... 39
  2.6. Accuracy of the estimation .................................................................................... 42
  2.7. Summary ............................................................................................................... 43
  2.8. Evolution in the development of the vision-based position estimation system .......... 44
3. Application of virtual sensor to Fault Detection and Identification ............................ 47
  3.1. Introduction .......................................................................................................... 47
  3.2. Additive positioning sensor fault .......................................................................... 47
  3.3. Lock-in-place fault ............................................................................................... 49
3.4. Criterion for virtual sensor rejection ............................................................... 50
3.5. Threshold function ......................................................................................... 51

4. Simulation of perturbations in the virtual sensor over quadrotor trajectory control .... 55
4.1. Introduction ..................................................................................................... 55
4.2. Quadrotor trajectory control ........................................................................ 56
  4.2.1. Dynamic model ........................................................................................ 56
  4.2.2. Attitude and height control ..................................................................... 58
  4.2.3. Velocity control ....................................................................................... 59
  4.2.4. Trajectory generation .............................................................................. 59
4.3. Model of perturbations .................................................................................. 60
  4.3.1. Sampling rate .......................................................................................... 60
  4.3.2. Delay ........................................................................................................ 60
  4.3.3. Noise ........................................................................................................ 60
  4.3.4. Outliers ................................................................................................... 61
  4.3.5. Packet loss .............................................................................................. 61
4.4. Simulation results .......................................................................................... 61
  4.4.1. Different speeds and delays ................................................................... 62
  4.4.2. Noise and outliers with fixed delay ........................................................ 63
  4.4.3. Delay, noise, outliers and packet loss ..................................................... 64

5. Conclusions ........................................................................................................ 68

REFERENCES ....................................................................................................... 71
List of Figures

Figure 1. Estimated percentage of the development time for each of the phases of the project ................................................................. 14
Figure 2. Gantt diagram with the evolution of the project................................................................. 15
Figure 3. Two quadrotors with cameras in their base tracking a third quadrotor whose position want to be estimated, represented by the green ball ................................................................. 17
Figure 4. Images taken during data acquisition experiments at the same time from both cameras, with two orange balls at the top of a Hummingbird quadrotor ................................................. 18
Figure 5. Relative position vectors between the cameras and the tracked quadrotor ................ 19
Figure 6. Pin-hole camera model ................................................................................................ 20
Figure 7. State machine implemented in the modified versión of the CAMShift algorithm ...... 25
Figure 8. Orange rubber hat at the top of the Hummingbird quadrotor used as visual marker 28
Figure 9. Camera configuration with parallel optical axes in the Y-axis of the global frame and fixed quadrotor ........................................................................................................ 29
Figure 10. Position estimation error in XYZ (blue, green, red) and distance between cameras and quadrotor (magenta, black) for fixed quadrotor and parallel optical axes. Blue marks “*” correspond to instants with tracking loss from one of the cameras ............................................. 30
Figure 11. Orthogonal configuration of the cameras .................................................................. 30
Figure 12. Position estimation error in XYZ (blue, green, red) and distance between cameras and quadrotor (magenta, black) for fixed quadrotor and orthogonal configuration of the cameras ........................................................................................................ 31
Figure 13. Estimation error and distance with cameras with fixed quadrotor and moving cameras, initially with parallel optical axes and finally with orthogonal configuration .......... 32
Figure 14. Evolution of the position estimation error with multiple tracking losses (marked by a “*” character) in one of the cameras ........................................................................................................ 32
Figure 15. Orthogonal camera configuration with tracked quadrotor out of the FoV for one of the cameras ........................................................................................................ 33
Figure 16. Position estimation error and distance with cameras with long duration tracking loss for one of the cameras (blue and green “*” characters) and both cameras (red “*” characters) ........................................................................................................ 34
Figure 17. Number of consecutive frames with tracking loss (blue) and threshold (red) ........ 34
Figure 18. Configuration of the cameras and the quadrotor for the Z-axis estimation experiment ........................................................................................................ 35
Figure 19. Vicon height measurement (red) and vision-based estimation (blue) ....................... 36
Figure 20. Position error estimation in XYZ (blue, green, red) and distance between each of the cameras and the tracked quadrotor (magenta, black) ................................................................. 36
Figure 21. Configuration of the cameras for the experiment with depth (YE axis) and lateral motion (XE axis) ........................................................................................................ 37
Figure 22. X-axis estimation with depth and lateral motion for the quadrotor ......................... 38
Figure 23. Y-axis estimation with depth and lateral motion for the quadrotor ....................... 38
Figure 24. Number of consecutive frames with tracking loss with depth and lateral motion for the quadrotor ........................................................................................................................................ 39
Figure 25. Configuration of the cameras with the quadrotor executing circular and random trajectories .................................................................................................................................. 40
Figure 26. X-axis estimation and real position when the quadrotor is executing circular and random trajectories ................................................................................................................... 40
Figure 27. Y-axis estimation and real position when the quadrotor is executing circular and random trajectories ................................................................................................................... 41
Figure 28. Number of consecutive frames with tracking loss when the quadrotor is executing circular and random trajectories .................................................................................................................................. 41
Figure 29. Simulation of GPS data with drift error between t = 20 s and t = 30 s ...................... 48
Figure 30. Distance between position given by GPS and vision-based estimation with GPS drift error .......................................................................................................................................... 48
Figure 31. GPS simulated data with fixed measurement from t = 110 s .................................... 49
Figure 32. Distance between vision-based position estimation and GPS with faulty data (black), and threshold (red) ........................................................................................................................................ 50
Figure 33. Number of consecutive frames with tracking loss and threshold for rejecting virtual sensor estimation .................................................................................................................................. 51
Figure 34. Angle and distance between the camera and the tracked object ............................. 52
Figure 35. Estimation error and distance with cameras with the cameras changing from parallel to orthogonal configuration .................................................................................................................................. 52
Figure 36. Distance between GPS simulated data and estimated position and threshold corresponding to Figure 35 ........................................................................................................................................ 53
Figure 37. Two quadrotors with cameras in their base tracking a third quadrotor whose position want to be estimated, represented by the green ball .................................................................................................................................. 55
Figure 38. Images taken during data acquisition experiments at the same time from both cameras, with two orange balls at the top of a Hummingbird quadrotor ................................................ 56
Figure 39. Reference path and trajectories followed in XY plane with different values of delay in XY position measurement, fixed delay of 100 ms in height measurement and V = 0,5 m·s⁻¹. 62
Figure 40. External estimation of XY position with Gaussian noise, outliers and a reference speed of V = 0,5 m·s⁻¹ ........................................................................................................................................ 64
Figure 41. Trajectories followed by the quadrotor with noise and outliers (blue) and without them (black) ........................................................................................................................................ 64
Figure 42. Quadrotor trajectories with simultaneous application of noise, delay, outliers and packet loss for V = 0,5 m·s⁻¹ (blue) and V = 0,75 m·s⁻¹ (green) .................................................................................................................................. 65
Figure 43. Reference and real value for height when XY position is affected by noise, delay, outliers and packet loss with a reference speed of V = 0,75 m·s⁻¹ ........................................................................................................................................ 66
Acknowledgments

This work was partially funded by the European Commission under the FP7 Integrated Project EC-SAFEMOBIL (FP7-288082) and the CLEAR Project (DPI2011-28937-C02-01) funded by the Ministerio de Ciencia e Innovacion of the Spanish Government.

The author wishes to acknowledge the support received by the CATEC during the experiments carried out in its testbed. Special thanks to Miguel Ángel Trujillo and Jonathan Ruiz from CATEC, and to professors José Ramiro Martinez de Dios and Begoña C. Arrúe Ullés from the University of Seville, for their help.

Finally, the author wants to remark all the help and advice provided by the supervisors Guillermo Heredia Benot and Aníbal Ollero Baturone.
Publications

Accepted papers


Awaiting acceptance papers

1. INTRODUCTION

1.1. Introduction

Position estimation is an important issue in many mobile robotics applications, where automatic position or trajectory control is required. This problem can be found in very different scenarios, including both terrestrial and aerial robots, with different specifications in accuracy, reliability, cost, weight, size, or computational resources. Two main approaches can be considered: position estimation based on odometry, beacons or any other internal sensors, or using a global positioning system such as GPS or Galileo. Each of these technologies has its advantages and disadvantages. The selection of a specific device will depend on the particular application, namely, the specifications in the operation conditions of the robot. For example, it is well known that GPS sensors only work with satellite visibility, so they cannot operate in indoors, but they are extensively used in fixed-wing UAVs and other outdoor exploration vehicles. Another drawback of these devices is their low accuracy, with position errors around two meters, although centimeter accuracies can be obtained with Differential GPS (DGPS). For small indoor wheeled robots, a simple and low cost solution is to use odometry methods, integrating speed or acceleration information obtained from optical encoders or Inertial Measurement Units (IMU). However, the lack of a position reference will cause a drift error along the time, so the estimation might become useless after a few seconds. In recent years, a great effort to solve the Simultaneous Localization and Mapping (SLAM) problem has been dedicated, making possible its application in real time, although it still carries high computational costs.

The current trend is to integrate multiple sources of information, fusing their data in order to obtain a better estimation for accuracy and reliability. The sensors can be both static at fixed positions or mounted over mobile robots. Multi-robot systems for instance are platforms were these techniques can be implemented naturally. This work is focused in multi-quadrotor systems with a camera mounted in the base of each UAV, so the position of a certain quadrotor will be obtained from the centroid of its projection over the image planes and the position and orientation of the cameras. Moreover, the Kalman filter used as estimator will also provide the velocity of the vehicle.

The external estimation obtained (position, orientation or velocity) can be used for controlling the vehicle. However, some aspects such as estimation errors, delays or estimation availability have to be considered carefully. The effects of new perturbations introduced in the control loop should be analyzed in simulation previous to its application in the real system, so potential accidents causing human or material damages can be avoided.
1.2. General description of the project

The goal of this work is the development of a system for obtaining a position estimation of a quadrotor being visually tracked by two cameras whose position and orientation are known. A simulation study based on data obtained from experiments will be carried for detecting failures on internal sensors, allowing the system reconfiguration to keep the system under control. If the vision-based position estimation provided by the virtual sensor is going to be used in position or trajectory control, it is convenient to study the effects of the associated perturbations (delays, tracking loss, noise, outliers) over the control. Before testing it in real conditions, with the associated risk of accidents and human or material damages, it is preferable to analyze the performance of the controller in simulation. So a simulator of the quadrotor dynamics and its trajectory control system, including the simulation of the identified perturbations, was built.

The position and velocity of the tracked object on the 3D space will be obtained from an Extended Kalman Filter (EKF), taking as input the centroid of the object on every image plane of the cameras, as well as their position and orientation. Two visual tracking algorithms were used in this project: the Tracking-Learning-Detection (TLD), and a modified version of the CAMShift algorithm. However, TLD algorithm was rejected due to high computational costs and its bad results applied to the quadrotor tracking, as it is based on template matching. On the other hand, CAMShift algorithm is a color-based tracking algorithm that uses the HSV color space for extracting color information (Hue component) in a single channel image, simplifying the object identification and making it robust to illumination and appearance changes.

In multi-UAV systems such as formation flight, cooperative surveillance and monitoring or aerial refueling, every robot might carry additional sensors, not for their own control, but for estimating part of the state of other vehicle, for example its position, velocity or orientation. This external estimation can be seen as a virtual sensor, in the sense that it provides a measurement of a certain signal computed from other sensors. In normal conditions, both internal and virtual sensors should provide similar measurements. However, consider a situation with a UAV approaching to an area without satellite visibility, so its GPS sensor is not able to provide position data but the IMU keeps integrating acceleration, increasing error with the time, and then the difference between both sources becomes significant. If a certain threshold is exceeded, the GPS can be considered as faulty, starting a reconfiguration process that handles this situation.

For the external position estimation, a communication network is necessary for the interchange of the information used on its computation (time stamp, position and orientation of the cameras, centroid of the tracked object in the image plane). Although this is beyond the scope of this work, communication delays and packet losses should be taken into account when the virtual sensor is going to be used for controlling the UAV.

A quadrotor simulator with its trajectory control system has been developed for studying the effects of a number of perturbations identified during experiments, including those related to communications. The simulator was implemented as a MATLAB-Simulink block diagram that includes quadrotor dynamics, attitude, position and trajectory controllers, and a way-point
generator. Graphical and numerical results are shown in different conditions, highlighting the most important aspects in each case. These results should be used as reference only, as the effects of perturbations over quadrotor performance will depend on the control scheme being used.

Finally, all position estimation experiments were performed hand holding the cameras: they were not mounted in the base of any quadrotor. What is more, both cameras used were connected to the same computer through a five-meter cable, what limited the movements around the tracked UAV. In the next step of the project (not considered here), the cameras will be mounted on the quadrotors, and image processing will be done onboard or in a ground control station. The onboard image acquisition and processing introduces additional problems such as vibrations, weight limitations, or available bandwidth.

1.3. Related works

The main contribution of this work is the application of the visual position estimation to the Fault Detection and Identification (FDI). However, a number of issues have also been treated, including visual tracking algorithms, quadrotor dynamic modeling and quadrotor control.

1.3.1. Vision-based position estimation

The problem of multi-UAV position estimation in the context of forest fire detection has been treated in [1], estimating motion from multiple planar homographies. Accelerometer, gyroscope and visual sensor measurements are combined in [2] using a non-linear complementary filter for estimating pose and linear velocity in an aerial robot. Simultaneously Localization and Mapping (SLAM) problem has been applied to small UAVs in outdoors, in partially structured environments [3]. Quadrotor control using onboard or ground cameras is described in [4] and [5]. Here both position and orientation measurements are computed from the images provided by a pair of cameras. Homography techniques are combined with a Kalman filter in [6] for obtaining UAV position estimation when building mosaics. Other applications where vision-based position estimation can be employed include formation flight and aerial refueling [7], [8], [9], [10].

1.3.2. Visual tracking algorithms

Visual tracking algorithms with application to position estimation of moving objects have to be fast enough to provide an accurate estimation. As commented earlier, TLD algorithm [11] was tested in the first place due to its ability to adapt to changes in the appearance of the object. However, as this algorithm is based on template matching and the surfaces of the quadrotors
are not big enough, most of the time the tracking was lost. On the other hand, the execution time was too high due to the high number of operations involved in the correlations with the template list. Color-based tracking algorithms such as CAMShift [12] present good properties for this purpose, including simplicity, low computation time, invariance to changes in the illumination, rotation and position, or noise rejection. A color marker in contrast with the background has to be disposed in the object to be tracked. The problem of this algorithm appears when an object with similar color is in the image, although it can be solved considering additional features. The basic CAMShift assumes that tracked object is always visible on the image, so it cannot handle tracking losses. Some modifications have been done to CAMShift to make possible tracking recovery when object is temporarily occluded, when it changes its appearance or when similarly colored objects are contained in the scene [13]. A Kalman filter is used in [14] for handling occlusions, while a multidimensional color histogram in combination with motion information solve the problem of distinguishing color.

1.3.3. FDIR

Reliability and fault tolerance has always been an important issue in UAVs [20], where Fault Detection and Identification (FDI) techniques play an important role in the efforts to increase the reliability of the systems. It is even more important when teams of aerial vehicles cooperate closely between them and the environment, as it is the case in formation flight and heterogeneous UAV teams, because collisions between them or between the vehicles and objects of the environment may arise.

In a team of cooperating autonomous vehicles, FDI of individual vehicles in which they use their own sensors for FDI can be regarded as Component Level (CL) FDI. Most CL-FDI applications to UAVs that appear in the literature use model-based methods, which try to diagnose faults using the redundancy of some mathematical description of the system dynamics. Model-based CL-FDI has been applied to unmanned aircraft, either fixed wing UAVs [21] or helicopter UAVs [22][23][24].

The Team Level (TL) FDI exploits the team information for detection of faults. Most published works rely on transmission of the state of the vehicles through the Communications channel for TL-FDI [25]. What has not been thoroughly explored is the use of the sensors onboard the other vehicles of the team for detection of faults in an autonomous vehicle, which requires sensing the state of a vehicle from the other team components.

1.3.4. Quadrotor dynamic modeling and control

Quadrotor modeling and control has been extensively treated in literature. The derivation of the dynamic model is described in detail in [15]. Some control methods applied in simulation and in real conditions can be found here too. PID, LQ and Backstepping controllers have been tested in [16] and [17] over an indoor micro quadrotor. Mathematical modeling and experimental results in quadrotor trajectory generation and control can be found in [18]. Ref.
[19] addresses the same problem but the trajectory generation allows the execution of aggressive maneuvers.

1.4. Development time estimation

The development of this project can be divided into three phases:

- Development of the vision-based position estimation system
- Development of the quadrotor trajectory control simulator
- Documentation (papers, reports, memories)

The estimation of the percentage of time dedicated to each of these phases has been represented in Figure 1. The Gantt diagram with the identified tasks and their start date and end date can be seen in Figure 2. The project started in November 2012, with the technical part being finished in June 2013. Since then, two papers have been sent to ROBOT 2013 congress (accepted) and ICRA 2014 (awaiting acceptance), and the project report has been written.

![Percentage of the Development Time](image.png)

**Figure 1. Estimated percentage of the development time for each of the phases of the project**
<table>
<thead>
<tr>
<th>Nombre de la tarea</th>
<th>P4</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Oct</td>
<td>Nov</td>
<td>Dic</td>
<td>Ene</td>
</tr>
<tr>
<td>Position estimation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation in Windows with TLD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptation to Ubuntu 12.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data acquisition experiments and analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation of modified CAMShift</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data acquisition and analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quadrotor simulator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quadrotor dynamics block diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trajectory controller block diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controller tuning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAFEMOBIL Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper to ROBOT2013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper to ICRA 2014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.** Gantt diagram with the evolution of the project
2. Vision-based position estimation in multiple quadrotor systems

2.1. Problem description

Consider a situation with three quadrotors A, B and C. Two of them, A and B, have cameras mounted in their base with known position and orientation referred to a global frame. Images taken from cameras are sent along with their position and orientation to a ground station. Both cameras will try to stay focused on the third quadrotor, C, so a tracking algorithm will be applied to obtain the centroid of the object on every received image. An external position estimator executed in the ground station will use this data to obtain an estimation of quadrotor C position that can be used for position or trajectory control in the case C does not have this kind of sensors, they are damaged, or they are temporarily unavailable. The situation described above has been shown in Figure 3. Here the cones represent the field of view of the cameras, the orange quadrotor is the one being tracked and the green ball corresponds to its position estimation.

![Figure 3. Two quadrotors with cameras in their base tracking a third quadrotor whose position want to be estimated, represented by the green ball](image)

One of the main issues in vision-based position estimation applied to trajectory or position control is the presence of delays in the control loop, which should not be too high to prevent the system of becoming unstable. The following sources of delay can be identified:

- Image acquisition delay
- Image transmission through radio link
- Image processing for tracking algorithm
- Position estimation and its transmission

The first two are imposed by hardware and available bandwidth. The last one is negligible in comparison with the others. On the other hand, image processing is very dependent on the
computation cost required by the tracking algorithm. In this work, the external position estimation system was developed and tested with real data, obtaining position and orientation of cameras and tracked quadrotor from a Vicon Motion Capture System in the CATEC testbed. The visual tracking algorithm used was a modified version of CAMShift algorithm. This color-based tracking algorithm uses Hue channel in the HSV image representation for building a model of the object and detecting it, applying Mean-Shift for computing the centroid of the probability distribution. As this algorithm is based only in color information, a small orange ball was disposed at the top of the tracked quadrotor, in contrast with the blue floor of the testbed. Figure 4 shows two images captured by the cameras during data acquisition phase.

![Figure 4. Images taken during data acquisition experiments at the same time from both cameras, with two orange balls at the top of a Hummingbird quadrotor](image)

Although in practical application the external position estimation process will run in real time, here the computations were done off-line in order to make easier the development and debug of this system, so the estimation was carried out in two phases:

1) The data acquisition phase, where images and the measurements of the position and orientation of both cameras and the tracked object were captured along with the time stamp and saved into a file and a directory containing all images.

2) The position estimation phase, corresponding to the execution of the extended Kalman filter that makes use of the captured data to provide an off-line estimation of the quadrotor position at every instant indicated by the time stamp.

As normal cameras do not provide depth information (unless other constraints are considered, such as tracked object size), two or more cameras are needed in order to obtain the position of the quadrotor in the three-dimensional space. Even with one camera, if it changes its position and orientation and the tracked object movement is reduced, the position can be estimated. One of the main advantages of using Kalman filter is its ability to integrate multiple sources of information, in the sense that it will try to provide the best estimation independently on the number of observations available at a certain instant. The results of the experiments presented here were obtained with two cameras, although in some cases the tracked object was occluded or out of the field of view (FoV) for one or both cameras. The
extended Kalman filter equations described later were obtained for two cameras, but they can be easily modified to consider an arbitrary number of cameras.

### 2.2. Model of the system

The system for the vision-based position estimation of a moving object using two cameras is represented in Figure 5. It is assumed for the cameras to be mounted on the quadrotors, but for clarity, they have not been drawn.

![Figure 5. Relative position vectors between the cameras and the tracked quadrotor](image)

The position and orientation of cameras and tracked quadrotor will be referred to fixed frame \( \{E\} = \{X_E, Y_E, Z_E\} \). For this problem, \( P_{\text{CAM1}}, P_{\text{CAM2}} \) and rotation matrixes \( R_{E\text{CAM1}} \) and \( R_{E\text{CAM2}} \) are known. The following relationships between position vectors are derived:

\[
\begin{align*}
P_{E\text{obj}} &= R_{E\text{CAM1}} \cdot P_{E\text{obj}}^{\text{CAM1}} + P_{\text{CAM1}} \\
P_{E\text{obj}} &= R_{E\text{CAM2}} \cdot P_{E\text{obj}}^{\text{CAM2}} + P_{\text{CAM2}}
\end{align*}
\]

The tracking algorithm will provide the centroid of the tracked object. The pin-hole camera model relates the position of the object referred to the camera coordinate system in 3D space with its projection in the image plane. Assuming that the optical axis is \( X \), then:

\[
\begin{align*}
x^{IMn} &= f_x \cdot \frac{y^{CAMn}}{x^{CAMn}_{\text{obj}}} \\
y^{IMn} &= f_y \cdot \frac{y^{CAMn}}{x^{CAMn}_{\text{obj}}}
\end{align*}
\]

where \( f_x \) and \( f_y \) are focal length in both axes of the cameras, assumed to be equal for all cameras. Figure 6 represents the pin-hole camera model with the indicated variables.
It also will be needed a model of camera lens for compensating typical radial and tangential distortion. A calibration process using chessboard pattern or circles pattern is required for obtaining distortion coefficients. There are two ways for compensate this kind of perturbation:

A) Backward compensation: given the centroid of the object in the image plane, the distortion is undone so the ideal projection of the point is obtained. However, it might require numerical approximations if equations are not invertible.

B) Forward compensation: position estimator will obtain an estimation of the object centroid, and is here the model of distortion is applied directly. The drawback of this solution is that distortion equations should be considered when computing jacobian matrix, otherwise a slight error must be accepted.

The distorted point on the image plane is computed as follows:

\[
\begin{bmatrix}
    x_d(1) \\
    x_d(2)
\end{bmatrix} = (1 + k_c(1) \cdot r^2 + k_c(2) \cdot r^4 + k_c(5) \cdot r^6) \cdot x_n + dx
\]  

(3)

Here \( x_n = [x, y]^T \) is the normalized image projection (without distortion), \( r^2 = x^2 + y^2 \) and and \( dx \) is the tangential distortion vector:

\[
\begin{bmatrix}
2 \cdot k_c(3) \cdot x \cdot y + k_c(4) \cdot (r^2 + 2 \cdot x^2) \\
 k_c(3) \cdot (r^2 + 2 \cdot y^2) + 2 \cdot k_c(4) \cdot x \cdot y
\end{bmatrix}
\]

(4)

The vector with the distortion coefficients, \( k_c \), as well as the focal length and the principal point of the cameras were obtained with the MATLAB camera calibration toolbox.
2.3. Position estimation algorithm

An Extended Kalman Filter (EKF) was used for the position estimation of the tracked object from its centroid in both images and the position and orientation of the cameras. The “extended” version of the algorithm is used because of the presence of nonlinearities in the rotation matrix and in pin-hole camera model. For EKF application, a nonlinear state space description of the system is considered in the following way:

\[
x_k = f(x_{k-1}) + w_{k-1} \\
z_k = h(x_k) + v_k
\]

where \(x_k\) is the state vector, \(f(\cdot)\) is the state evolution function, \(z_k\) is the measurement vector, \(h(\cdot)\) is the output function, and \(w_k\) and \(v_k\) are Gaussian noise processes.

State vector will contain position and velocity of the tracked UAV referred to fixed frame \(\{E\}\), while measurement vector will contain the centroid of the object in both images given by the tracking algorithm in current instant, but also in previous one (this is done for taking into account velocity when updating estimation). These two vectors are then given by:

\[
x_k = [x_k, y_k, z_k, v_{x_k}^k, v_{y_k}^k, v_{z_k}^k]^T \\
z_k = [x_k^{M1}, y_k^{M1}, z_k^{M1}, x_k^{M2}, y_k^{M2}, z_k^{M2}]^T
\]

If no other information source can be used, a linear motion model is assumed, so system evolution function will be:

\[
x_k = \begin{bmatrix} x_k \\ y_k \\ z_k \\ v_{x_k}^k \\ v_{y_k}^k \\ v_{z_k}^k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \Delta t \cdot v_{x_{k-1}}^k \\ y_{k-1} + \Delta t \cdot v_{y_{k-1}}^k \\ z_{k-1} + \Delta t \cdot v_{z_{k-1}}^k \\ v_{x_{k-1}}^k + \Delta t \cdot a_{x_{k-1}}^k \\ v_{y_{k-1}}^k + \Delta t \cdot a_{y_{k-1}}^k \\ v_{z_{k-1}}^k + \Delta t \cdot a_{z_{k-1}}^k \end{bmatrix}
\]

Here \(\Delta t\) is the elapsed time between consecutives updates. If acceleration of the tracked quadrotor can be obtained from its internal sensors or computed from orientation, this information would be integrated in the last three terms of the system evolution function:

\[
x_k = \begin{bmatrix} x_k \\ y_k \\ z_k \\ v_{x_k}^k \\ v_{y_k}^k \\ v_{z_k}^k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \Delta t \cdot v_{x_{k-1}}^k \\ y_{k-1} + \Delta t \cdot v_{y_{k-1}}^k \\ z_{k-1} + \Delta t \cdot v_{z_{k-1}}^k \\ v_{x_{k-1}}^k + \Delta t \cdot a_{x_{k-1}}^k \\ v_{y_{k-1}}^k + \Delta t \cdot a_{y_{k-1}}^k \\ v_{z_{k-1}}^k + \Delta t \cdot a_{z_{k-1}}^k \end{bmatrix}
\]
On the other hand, it is necessary to relate the measurable variables (centroid of the tracked object on every image plane) with the state variables. This can be done through the equations of the model:

\[
\begin{bmatrix}
    x_{k}^{Mn} \\
    y_{k}^{Mn} \\
    z_{k}^{Mn}
\end{bmatrix}
= \begin{bmatrix}
    f_x \cdot r_{12}^n \cdot (x_k - x_k^{CAMn}) + r_{22}^n \cdot (y_k - y_k^{CAMn}) + r_{32}^n \cdot (z_k - z_k^{CAMn}) \\
    f_y \cdot r_{13}^n \cdot (x_k - x_k^{CAMn}) + r_{23}^n \cdot (y_k - y_k^{CAMn}) + r_{33}^n \cdot (z_k - z_k^{CAMn}) \\
    f_z \cdot r_{11}^n \cdot (x_k - x_k^{CAMn}) + r_{21}^n \cdot (y_k - y_k^{CAMn}) + r_{31}^n \cdot (z_k - z_k^{CAMn})
\end{bmatrix}
\]

(9)

Here \( r_{ij}^n \) is the \( ij \) element of the rotation matrix for the \( n \)-th camera. For computing at instant \( k \) the centroid of the object at instant \( k-1 \), we make use of the following equation:

\[
\begin{bmatrix}
    x_{k-1} \\
    y_{k-1} \\
    z_{k-1}
\end{bmatrix} = \begin{bmatrix}
    x_k - v_k^x \cdot \Delta t \\
    y_k - v_k^y \cdot \Delta t \\
    z_k - v_k^z \cdot \Delta t
\end{bmatrix}
\]

(10)

Then, an equivalent expression to (9) is obtained.

Jacobian matrixes \( J_f \) and \( J_h \) used in EKF equations can be easily obtaining from these two expressions. However, only the matrix corresponding to the state evolution function is reproduced here due to space limitations:

\[
J_f = \begin{bmatrix}
    1 & 0 & 0 & \Delta t & 0 & 0 \\
    0 & 1 & 0 & 0 & \Delta t & 0 \\
    0 & 0 & 1 & 0 & 0 & \Delta t \\
    0 & 0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

(11)

Now let consider the general position estimation problem with \( N \) cameras. The state vector is the same as defined in (7), however, for practical reasons, the measurement vector will only contain the centroid of the tracked object at instants \( k \) and \( k-1 \) for one camera each time:

\[
z_k^n = [x_k^{Mn}, y_k^{Mn}, x_{k-1}^{Mn}, y_{k-1}^{Mn}]^T
\]

(12)

The vector \( z_k^n \) represents the measurements of camera \( n \)-th at iteration \( k \). As the number of cameras increases, the assumption of simultaneous image acquisition might not be valid, and a model with different image acquisition times is preferred instead. This requires a synchronization process with a global time stamp indicating when the image was taken. Let define the following variables:

- \( t_p \): time instant of last estimation update
- \( t_{aq}^n \): time instant when image of camera \( n \)-th was captured
If $t_{acq}^m$ is the time stamp of the last image captured, then:

$$\Delta t = t_{acq}^m - t_0$$
$$t_0 = t_{acq}^m$$

(13)

The rest of computations are the same as described for the case with two cameras.

### 2.4. Visual tracking algorithm

Visual tracking applied in the position estimation and control imposes hard restrictions in computational time, in the sense that delays in position measurements affect significantly the performance of the trajectory control, limiting the speed of the vehicle to prevent it from becoming unstable. However, vision-based tracking algorithms must include other properties like:

- Robustness to light conditions
- Noise immunity
- Ability to support changes in the orientation of the tracked object
- Low memory requirements, what usually also implies low computation time
- Capable to recover from temporal losses (occlusions, object out of FoV)
- Support image blurring due to camera motion
- Applicable with moving cameras

It must be taken into account that quadrotors have a small surface, and its projection in the image can be very changing due to its “X” shape. On the other hand, using color markers do not affect the quadrotor control, but they simplify the visual detection task.

In this research, we tested two tracking algorithms: TLD (Tracking-Learning-Detection) and a modified version of CAM-Shift. The first one builds a model of the tracked object while it is on execution, adding a new template to the list when a significant difference between current observation and model is detected. For smooth operation, TLD needs the tracked object to have significant edges and contours since detection is made through a correlation between templates and image patches with different position and scales. This makes its use for quadrotor tracking difficult due to the small surface and uniformity in the quadrotor shape. Experimental results show the following problems in the application of this algorithm:

- Significant error in centroid estimation
- Computation time is relatively high
- Too many false-positives are found

23
• The bounding box around the object tends to diverge
• Tracked object is usually lost when it comes far from camera
• Tracked object is lost when background is not uniform

Better results were obtained with a modified version of CAMShift algorithm (Continuously Adaptive Mean-Shift). CAMShift is a color-based tracking algorithm, so a color marker is required to be placed in a visible part of the quadrotor in contrast with the background color. In tests, we put two small orange balls at the top of the quadrotor, while the floor of the testbed was blue. The tracked object (the two orange balls, not the quadrotor) is represented by a histogram of the hue component containing the color distribution of the object. Here the HSV (Hue-Saturation-Value) image representation is used instead of RGB color space. This representation allows the extraction of color information and its treatment as a one-dimensional magnitude, so histogram based techniques can be applied. Saturation (color density) and Value (brightness) are limited in order to reject noise and other perturbations. For every image received, the CAMShift algorithm computes a probability image weighting Hue component of every pixel with the color distribution histogram of the tracked object, so the pixels with color closer to the object will have higher probabilities. Mean-Shift algorithm is then applied to obtain the maximum of the image probability in an iterative process. The process computes the centroid of the probability distribution within a window that will slide in the direction of the maximum until its center converges. Denoting the probability image by $P(x,y)$, then the centroid of the distribution inside searching window will be given by:

$$x_c = \frac{M_{10}}{M_{00}} ; \quad y_c = \frac{M_{01}}{M_{00}}$$

(14)

where $M_{00}$, $M_{10}$ and $M_{01}$ are zero and first order moments of image probability computed as follows:

$$M_{00} = \sum_{x} \sum_{y} P(x,y) \quad ; \quad M_{10} = \sum_{x} \sum_{y} x \cdot P(x,y) \quad ; \quad M_{01} = \sum_{x} \sum_{y} y \cdot P(x,y)$$

(15)

CAMShift algorithm will return an orientated ellipse around the tracked object, whose dimensions and orientation will be obtained from second order moments.

The basic implementation of CAMShift assumes that there is a nonzero probability in the image at all times. However, if tracked object is temporarily lost from image due to occlusions or because it is out of the field of view, the algorithm must be able to detect tracked object loss and redetect it, so tracking can be reset once object is visible again. This can be seen as a two state machine, as shown in Figure 7:
For detecting object lost and for object redetection, a number of criterions can be used:

- The zero order moment, as an estimation of the object size
- The size, dimensions or aspect ratio of the bounding box around the object
- One or more templates of the near surrounding of the color marker that include the quadrotor
- A vector of features obtained from SURF, SIFT or similar algorithms

For detection process, the incoming image will be divided into a set of patches, and for every patch, a measurement of the probability for the tracked object to be contained there is computed, and then, the detector will return the position of the most probable patch. If this path is a false positive, it will be rejected by the object loss detector, and a new detection process will begin. Otherwise, CAMShift will use this patch as initial search window.

Experimental results made us conclude:

- CAMShift is about 5-10 times faster than TLD
- Modified version of CAMShift can recover in short time from object lost when this is visible again
- False-positive are rejected by the object loss detector
- CAMShift can follow objects further than TLD
- CAMShift requires much less computation resources than TLD

2.5. Experimental results

This section presents graphical and numerical results of vision-based position estimation using the algorithms explained above. Previously, it is described the software developed specifically for this experiments, as well as the conditions, equipment and personal involved.

2.5.1. Software implementation
Three software modules were developed to support the experiments of vision-based position estimation. These experiments were divided into two phases: the data acquisition phase, and the data analysis phase. Real-time estimation experiments have not been done.

- **Data acquisition module**: this program was written in C++ for both Ubuntu 12.04 and Windows 8 Operative Systems using Eclipse Juno IDE and Microsoft Visual Studio Express 2012, respectively. It makes use of Open CV (Open Computer Vision) library, as well as Vicon Data Stream SDK library. The program contains a main loop where images of the two cameras are captured and saved as individual files along with the measurements of the position and orientation of both cameras and the tracked object given by a Vicon Motion Capture System. These measurements are saved in a text file with their corresponding time stamp. Images and measurements are assumed to be captured at the same time for the estimation process, although in practical, these data are obtained sequentially, so there is a slight delay. Previously to the data acquisition loop, the user must specify the resolution of the cameras and the folder name where images and Vicon measurements are stored.

- **Tracking and position estimation module**: this program was also implemented in C++ for Ubuntu 12.04 and Windows 8, using Open CV and ROS (Robot Operating System) libraries. It was designed to accept data in real time but also data captured by the acquisition program. It has not been tested in real time. Until now, it has only been used for the off-line position estimation. The program contains a main loop with the execution of the tracking algorithm and the extended Kalman filter. It also performs the same functions as the data acquisition program, taking sequentially images from cameras as well as position and orientation measurements from Vicon. The modified version of the CAMShift algorithm returns the centroid of the tracked quadrotor for every image in both cameras. Then, the position estimation is updated with this information and visualized with the *rviz* application from ROS. It was found that using ROS and Vicon Data Stream libraries simultaneously causes an execution error that was reported by other users. The tracking and position estimation program is not complete jet. The selection of the tracked object is done manually drawing a rectangle around it. The modified CAMShift provides good results taking into account the fast movement of the quadrotor and the blurring of the images in some experiments, but in a number of situations it returns false positives when the tracked object is out of the field of view and there is an object with a similar color within the image. On the other hand, Kalman filter tuning takes too much time when performed with the tracking algorithm. However, the position estimation is computed from the position and orientation measurements and from the centroid of the tracked object on the image plane for both cameras, so images are no longer necessary if the centroids have already been obtained by the tracking algorithm.

- **Position estimation module**: the position estimation algorithm was implemented in a MATLAB script in order to make easier and faster the Kalman filter setting. It takes as input the position and orientation measurements of the cameras and the quadrotor
(used as ground truth), the time stamp, the centroid of the tracked object given by CAMShift, and a flag indicating for every camera if the tracking is lost in current frame. As output, the estimator provides the position and velocity of the quadrotor in the global coordinate system. In the data acquisition phase, the real position of the quadrotor was also recorded, making possible the computation of the estimation error. This magnitude, distance between tracked quadrotor and cameras, tracking loss flag and other signals are represented graphically for better analysis of the results.

2.5.2. Description of the experiments

The data acquisition experiments were carried out in the CATEC testbed using its Vicon Motion Capture System for obtaining position and orientation of the cameras and the tracked quadrotor. The acquisition program was executed in a workstation provided by CATEC or in a laptop provided by the University of Seville. Two USB cameras Logitech C525 were connected to the computer through five-meter USB cables. This limited the mobility of the cameras during the experiments when following the quadrotor. The cameras were mounted over independent bases whose position was measured by Vicon. The optical axis of the cameras corresponded to the X axis of the base. It is important for the estimation that both axes are parallel. Otherwise an estimation error proportional to the distance between the cameras and the quadrotor is derived.

The tracked object was a Hummingbird quadrotor. Two orange balls or a little rubber hat were disposed at the top of the UAV as visual marker in contrast with the blue floor of the testbed, as shown in Figure 8. The cameras tried to stay focused in this marker. One important aspect referred to the cameras is the autofocus. For data acquisition experiments, two webcams models were used: the Genius eFace 2025 (manually adjustable focus) and the Logitech C525 (autofocus). For applications with moving objects, cameras with fixed focus or manually adjustable are not recommended. On the other hand, the image quality in the case of the Logitech C525 was much better that with the Genius eFace 2025.
The experiments were carried out by three or four persons:

- The pilot of the quadrotor
- The person in charge of the data acquisition program
- Two persons for handing the cameras

At the beginning of each experiment, the coordinator indicates to the pilot and the responsible for the cameras the position and motion pattern to be executed, according to the planning defined previously. Then, the resolution of the images and the name of the folder where Vicon data and acquired images will be saved are specified. Each experiment took between 2 and 5 minutes. The total number of images acquired was around 40,000. The initial set up and the execution of the experiments were carried out in four hours.

2.5.3. Analysis of the results

The position estimation results are presented here in different conditions explained separately. The experiments were designed to consider a wide range of situations and configurations, with different resolution of the cameras. Graphics corresponding to estimation error also represent the distance between each of the cameras and the quadrotor for magnitude comparison. Typically, the estimation error in position is around 0.15 m for a mean distance of 5 m from cameras, although, as it will be seen later, the error will strongly depend on the relative position between cameras and quadrotor. The effect of the tracking loss has been represented with a blue or green character “*” for one of the cameras, and with a red character “**” if tracking is lost for both cameras.
2.5.3.1. **Fixed quadrotor with parallel cameras**

In this experiment, the quadrotor was fixed at the floor. The optical axes of the cameras were in parallel, with a base line around 1.5 meters. The situation is the one described in Figure 9. A resolution of 640x480 was selected. Figure 10 shows the estimation error in XYZ, as well as the distance between each of the cameras and the quadrotor. As seen, the position estimation error in the X and Z axes is around 15 cm, however, it reach the 3 m in the Y axis when the distance from cameras is maximum. In general, the most parallel the optical axes of the cameras are, the higher the error in depth estimation is.

![Camera configuration with parallel optical axes](image)

*Figure 9. Camera configuration with parallel optical axes in the Y-axis of the global frame and fixed quadrotor*
2.5.3.2. **Fixed quadrotor with orthogonal camera configuration**

Now the configuration is the one shown in Figure 11. The optical axes of both cameras are orthogonal, corresponding to the best case for the depth estimation. This fact is confirmed by the results shown in Figure 12, where it can be seen that the estimation error has been reduced considerably.
**2.5.3.3. Fixed quadrotor with moving cameras**

This experiment is a combination of the above two. At the beginning, the cameras are in parallel with a short base line. That is why the estimation error shown in Figure 13 is initially high. Then, the cameras are moved until they reach the orthogonal configuration (in $t = 17$ s), reducing at the same time the error. Figure 14 represent in more detail the evolution of the estimation error when tracking loss occurs from $t = 32.5$ s until $t = 34.5$ s. In two seconds, the estimation error in the Y-axis change in 50 cm due to the integration of the speed in the position estimation. In this case, the estimation is computed using monocular images from a single camera.
Figure 13. Estimation error and distance with cameras with fixed quadrotor and moving cameras, initially with parallel optical axes and finally with orthogonal configuration.

Figure 14. Evolution of the position estimation error with multiple tracking losses (marked by a ‘*’ character) in one of the cameras.
2.5.3.4. Fixed quadrotor with orthogonal camera configuration and tracking loss

The goal of this experiment is to study the effect of long-term tracking loss from one or both cameras over the position estimation. The quadrotor was in a fixed position with the cameras in an orthogonal configuration, as shown in Figure 15. Here, the quadrotor is out of the Field of View (FoV) for the right camera. The estimation error results have been represented in Figure 16. The green and blue characters “*” represent tracking loss from left or right camera, while red character “**” correspond to tracking loss from both cameras simultaneously. The distance between each of the cameras to the tracked quadrotor has also been plotted in magenta and black. As it can be seen, the error grows rapidly when the vision-based estimation becomes monocular. The error is increased in 1 meter in around 4 seconds.

The number of consecutive frames with tracking loss can be used as a criterion for rejecting the position estimation, defining a maximum threshold. This idea is shown in Figure 17, where it has been represented the number of consecutive frames with tracking loss and a threshold of 15 frames.

![Figure 15. Orthogonal camera configuration with tracked quadrotor out of the FoV for one of the cameras](image)
2.5.3.5. **Z-axis estimation with flying quadrotor and parallel camera configuration**

In this experiment the quadrotor height is estimated with the cameras being in the configuration indicated in Figure 18 and a resolution of 1280x720 pixels. The pilot of the quadrotor was asked to perform movements along the Z-axis with two meters amplitude.
Figure 19 shows the Z-axis measurement given by Vicon (taken as ground truth) and the corresponding estimation obtained from visual tracking. The highest estimation errors are associated with changes in the sign of the height speed. Tracking loss marks are not represented here. The estimation error in depth (corresponding to $Y_e$ axis in the global frame) is around 0.5 m, as shown in Figure 20, with a mean distance from cameras of 4.5 m.

The model of the system considered for the extended Kalman filter assumes that tracked object moves following a linear trajectory. This is why changes in the direction or in the sign of the movement have associated high errors in estimation. However, integrating acceleration information provided by the inertial sensors (IMU) in the EKF should reduce these errors. In the case of the quadrotors, the acceleration in the XY plane can be approximated from the roll and pitch angles.

![Figure 18. Configuration of the cameras and the quadrotor for the Z-axis estimation experiment](image)
Figure 19. Vicon height measurement (red) and vision-based estimation (blue)

Figure 20. Position error estimation in XYZ (blue, green, red) and distance between each of the cameras and the tracked quadrotor (magenta, black)
2.5.3.6. Depth and lateral motion in the XY plane

Here the quadrotor will move along the $Y_e$ (depth) and $X_e$ (lateral) axes maintaining a constant height. The configuration of the cameras and the quadrotor is the one represented in Figure 21, with an image resolution of 1280x720 pixels and a mean distance of 5 m between the cameras and the tracked quadrotor. A high resolution in the images implies high acquisition and computation time, and then, the number of estimation updates per second might be insufficient. On the other hand, if the quadrotor speed is too high to be tracked or it is out of the FoV of the cameras for a long time, the position estimation will result useless. Figure 22 and Figure 23 represent the real position of the quadrotor taken from Vicon and the estimated one when the amplitude of the depth motion is about 4 m, and the amplitude in the lateral motion is about 5 m. There are some high amplitude errors in estimation for both X and Y axes due to consecutive tracking losses, as shown in Figure 24. However, once the tracking is recovered, the estimation error is rapidly reduced to normal values around 0,15 m. As mentioned, it is convenient to define a maximum number of consecutive tracking losses, so the vision-based position estimation is rejected if this threshold is exceeded.

![Figure 21. Configuration of the cameras for the experiment with depth (YE axis) and lateral motion (XE axis)](image-url)
Figure 22. X-axis estimation with depth and lateral motion for the quadrotor

Figure 23. Y-axis estimation with depth and lateral motion for the quadrotor
2.5.3.7. Quadrotor executing circular and random trajectories

In this experiment the cameras stayed in the fixed positions indicated in Figure 25, changing their orientation to track the quadrotor. The pilot was asked to perform two types of movements with the UAV: circular trajectory with a radius about 3 or 4 m, changing to a random trajectory from $t = 80$ s. The image resolution was set to 1280x720 pixels. The mean distance between the cameras and the quadrotor was 4 m.

The circular and random trajectories can be considered as the two worst cases for the extended Kalman filter as the model of the systems assumes linear motion. Here, a reduced number of frames with tracking loss may have a considerable influence over the error in the position estimation, as shown in Figure 26 and Figure 27. In the case of the Y-axis estimation (Figure 27), the error increases when the distance to the cameras becomes higher, although this was already expected, as it corresponds to the direction of depth. Finally, Figure 28 shows the consecutive number of frames with tracking loss, what causes the high-amplitude errors in the X and Y position estimation.

Figure 24. Number of consecutive frames with tracking loss with depth and lateral motion for the quadrotor
Figure 25. Configuration of the cameras with the quadrotor executing circular and random trajectories

Figure 26. X-axis estimation and real position when the quadrotor is executing circular and random trajectories
Figure 27. Y-axis estimation and real position when the quadrotor is executing circular and random trajectories

Figure 28. Number of consecutive frames with tracking loss when the quadrotor is executing circular and random trajectories
2.6. Accuracy of the estimation

The accuracy of the vision-based position estimation will mainly depend on the number of available cameras without tracking loss, and their relative position between them. As shown previously in some experiments, the best configuration for the cameras is the orthogonal one, while the worst case corresponds to the optical axes being parallel. On the other hand, the image resolution is not so relevant for the accuracy. Moreover, as the estimation is computed from the centroid of the color marker, it is convenient that the projection of this marker has the smallest possible area, so a low resolution is preferable. However, if the color markers have a small area, the tracking algorithm will probably lose it when it moves away from the camera.

Table 1 contains some results obtained from the first two experiments presented above. The image resolution for all cases is 640x480.

<table>
<thead>
<tr>
<th>Camera configuration</th>
<th>Distance from cameras [m]</th>
<th>Depth axis error [m]</th>
<th>Transversal axes error [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>4.75</td>
<td>1.5</td>
<td>0.15</td>
</tr>
<tr>
<td>Parallel</td>
<td>6.75</td>
<td>3</td>
<td>0.3</td>
</tr>
<tr>
<td>Orthogonal</td>
<td>4.6</td>
<td>0.25</td>
<td>0.15</td>
</tr>
<tr>
<td>Orthogonal</td>
<td>6</td>
<td>0.35</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 1. Estimation error in depth axis error ($Y_E$) and transversal axes ($X_E$, $Z_E$) for different configurations and distances

As commented earlier, there will be an offset error in the position estimation associated to an offset in the camera orientation measurement. The estimation error will increase proportionally with the distance between the camera and the tracked object, although it could be partially corrected by a calibration process. On the other hand, a small amplitude noise can be identified in graphical results due to a number of causes including noise in the position and orientation measurements of the cameras, errors in the centroid given by CAMShift or computation delays. However, this noise is negligible in comparison with the offset or the error due to tracking loss.

The visual estimation is rapidly degraded when the tracking is lost from one or both cameras, with an error approximately quadratic with time. Experimental results show that the error increases up to one meter in two seconds of monocular tracking with the quadrotor being in a fixed position.
2.7. Summary

As summary, graphical and numerical results of the vision-based position estimation have been shown in different conditions and configurations. The highlights are as follows:

- The error in the position estimation strongly depends on the angle between the cameras and the tracked object so that the best configuration is the orthogonal one and the worst case corresponds to the cameras being parallel.
- Even in the parallel configuration, there is an offset error in the position estimation due to alignment error between the optical axis of the cameras and the corresponding axis of the base whose orientation is measured. This error will depend on the distance between the cameras and the UAV, but it could be corrected by calibration.
- The tracking loss from one of the cameras makes the estimation error increases up to one meter in around two seconds. The position estimation should be rejected if the number of consecutive losses exceeds a certain threshold.
- The position estimation might be enhanced if the extended Kalman filter integrates information of acceleration from the internal IMU of the UAV being tracked.

It must be taken into account that the EKF provides the estimated position of the color marker, not the estimation of the own UAV. The estimation is computed from the projection of a single pixel of the marker, assuming its centroid on the image. It is desirable for the marker to have a spherical shape so its projection over the image plane is independent from the viewpoint.

If this estimation is going to be used for controlling the UAV, replacing its internal position sensors, it is important to consider the sources of delay that will disturb the control. The delay here is defined as the elapsed time since the images are going to be captured until the estimation is received by the quadrotor. The following delays can be identified:

a) Image acquisition delay, depending on the resolution being used and the speed of the sensor, it is equal to the inverse of the number of frames per second. A typical value of 30 ms can be considered for reference.

b) Image transmission delay, affected by the image resolution and compression algorithm (if digital) and available bandwidth.

c) Tracking algorithm delay, in the case of the CAMShift algorithm, it is around 5-10 ms for a 640x480 resolution, although it will depend on the processor speed and the image resolution.

d) Estimation update delay, the computation time for an iteration of the EKF, negligible in comparison with the others.

<table>
<thead>
<tr>
<th></th>
<th>Estimation transmission delay, namely, the elapsed time between the transmission and reception of the position estimation to the UAV through the communication network. It can also be considered negligible, but it might be affected by packet loss.</th>
</tr>
</thead>
</table>

In practice, the relative velocity between the UAV and the cameras should be limited (otherwise the tracking algorithm would be unable to track the object), and it should
convenient that the trajectories followed by the tracked object are as linear as possible so the model of motion considered in the EKF can be applied.

The definition of the variable threshold for the Fault Detection and Identification (FDI) depending on the distance from the furthest camera, allows the automatic rejection of the vision-based position estimation when there is a high number of consecutive frames with tracking loss. In a system with only two cameras, if one of them is unable to track the object, then the estimation will be monocular with the corresponding error integration. However, the variable threshold will increase in the same manner as the estimated distance to the furthest camera.

Execution times Two execution times were measured:

- Image acquisition: depending on the image resolution, the data interface (USB, FireWire IEEE 1394) and the USB hub internal to the computer. With the USB cable, the acquisition time for each of the cameras at 640x480 was around 15-20 ms, and it increases to 40-50 ms for 1280x720 pixels.
- Image processing: corresponding to the tracking algorithm. The time consumed by CAMShift with an image resolution of 640x480 pixels is about 5-10 ms.

The time consumed every iteration by the Extended Kalman Filter was not measured, but it is negligible in comparison with these two.

2.8. Evolution in the development of the vision-based position estimation system

A first version of the tracking and estimation program was completed in a month (November). It was programmed in C++ with the Visual Studio environment in Windows OS. The tracking algorithm used by the time was the TLD (Tracking-Learning-Detection). For the first tests, the cameras had to be in fixed positions, as position and orientation measurements were not available. The estimation program was executed in real time, with a virtual reality representation of the estimation generated with VTK (Visualization Tool Kit) library.

After a meeting with CATEC, it was decided to change the platform from Windows to Ubuntu. The installation of the 12.04 LTS version along with the Open CV library, the Eclipse Juno IDE and other software used took another month. The Open CV version used in Windows was the 2.4.2, however, the 2.4.3 version was released by the time. Detecting compatibility problems with the latest version cost two weeks.

In the next month (January), the estimation program was adapted to Ubuntu. The first version was written in C, changing to C++ language. Some specific functions, such as timing functions, had to be modified because they are different in Windows and Linux. The CATEC provided a library for accessing the position and orientation measurements of its Vicon Motion Capture System, and then, the data acquisition program was developed. The first experiments were
carried out in these days. The resolution of the images was limited to 320x240 pixels due to some kind of problem with Ubuntu version, because Windows supported a resolution of 640x480 with no problem. On the other hand, the lack of planning for the execution of the experiments made that the images captured resulted useless.

The second set of experiments was carried out in February. The analysis of the data allowed the identification of two errors in the implementation of the extended Kalman filter. At the beginning, the optical axis was assumed to be the $Y$, but the orientation measurements were taken in such a way that it corresponded to the $X$ axis. The other error was in the sign of the jacobian matrix $J_h$. Identifying these two problems took three or four weeks. The TLD tracking algorithm was found to be inadequate for the position estimation due to the reasons explained previously. The CAMShift algorithm was then adapted. The modified version was completed and tested with good results in March. The EKF was also adapted to MATLAB in order to reduce debugging time.

In July the data acquisition program was adapted to Windows to avoid the limitations in the resolution of the images, what allowed the acquisition at 1280x720 pixels for the two cameras at the same time. The Genius eFace 2025 webcams were replaced by the Logitech C525. The estimation results presented in this report correspond to the last experiments done.
3. Application of virtual sensor to Fault Detection and Identification

3.1. Introduction

The vision-based position estimation can be used for detecting fault or loss on the internal sensors of a certain tracked quadrotor and replace them to keep the vehicle under control. In normal conditions, the position given by both sources should be quite similar. However, if a GPS is being used and the UAV approaches or enters in covered zones without satellite visibility, the device will no longer provide position measurements and the estimation process will continue integrating acceleration data, with the corresponding position drift along time due to integration error. In such situations, the external estimation computed from the images captured by other UAV’s can replace the GPS, making possible the position or trajectory control.

Before testing the fault detection and recovery system in real conditions, a simulation study is presented here based on the previous results in the position estimation. Let call \((x_{GPS}, y_{GPS}, z_{GPS})\) to the position given by the GPS, \((x_e, y_e, z_e)\) to the estimated one, and \(d_1\) and \(d_2\) to the estimated distance between the quadrotor and each of the cameras. Then, if vision-based position estimation is assumed to be reliable, the GPS can be considered faulty if the following condition is satisfied:

\[
\|r_{GPS} - r_e\| = \sqrt{(x_{GPS} - x_e)^2 + (y_{GPS} - y_e)^2 + (z_{GPS} - z_e)^2} \\
\geq d_0 + K \cdot \max\{d_1, d_2\}
\]  \hspace{1cm} (16)

Here, \(K \cdot \max\{d_1, d_2\}\) corresponds to a threshold distance used as criterion for rejecting GPS data. As known, the larger the distance between the tracked object and the cameras, the less accurate the estimation is. Then, if one of the two cameras is far away from the quadrotor, the threshold distance should be tolerant enough to avoid considering the GPS as faulty when it is not (false positive). In the general case with \(N\) cameras, the maximum distance should be replaced by other kind of metric, such as the mean distance.

3.2. Additive positioning sensor fault

Figure 29 shows the simulated GPS data obtained from the experiment with fixed quadrotor and orthogonal camera configuration. An additive Gaussian white noise (AWGN) with 8 dB signal-to-noise ratio is added to the Vicon data, as well as a quadratic error between \(t = 20\) s and \(t = 30\) s, which simulates the temporal loss of the GPS signal. On the other hand, Figure 30 represents the distance between the positions given by the GPS and the position estimator. The dashed red line is a threshold distance for rejecting GPS data.
The threshold used here was the following:

\[ d_{th} = 2 + 0.1 \cdot \max\{d_1, d_2\} \]  \hfill (17)
3.3. Lock-in-place fault

Let consider now that the GPS signal is lost or the device is damaged, so the position given to the controller is constant, for example, equal to the last measurement. This is represented in Figure 31, where the simulated GPS data becomes constant from $t = 110$ s. Figure 32 represents the distance between the position estimated by the virtual sensor and the one given by the GPS, as well as the threshold for detecting sensor fault. As seen, the distance exceeds the threshold in most of the time. In order to avoid considering the GPS as failed due to outliers, it is convenient to define a counter or a timer for monitoring how much the fault lasts.
3.4. Criterion for virtual sensor rejection

In normal conditions, with the object being tracked with two or more cameras, the external position estimation will provide good results if cameras are not aligned (the optical axes are not parallel). In this case, it is possible to detect fault in the GPS due to signal loss or damages in the device. However, it is also possible that the estimation provided by visual techniques is not correct. The reliability of the vision-based position estimation will depend on the number of consecutive frames with tracking loss and the number of cameras available. Even with a single camera, if the tracked object is in a fixed position and the camera moves around it fast enough, it could be possible to obtain an acceptable estimation. One way to reject the virtual sensor estimation when using two cameras is to consider a counter that increases in one or two units depending on the number of cameras with tracking loss. This counter will be reseted once the tracking is recovered for both cameras. Therefore, a threshold representing the maximum number of consecutive frames with tracking loss allowed can be defined. This idea has been illustrated in Figure 33.
3.5. Threshold function

The threshold defined previously for detecting GPS fault will depend on the following three variables:

1) Distance to the furthest camera
2) Number of consecutive frames with tracking loss
3) Angle between the cameras and the tracked object

The threshold should increase as the virtual sensor becomes less reliable. In the first place, the error in the position estimation will be proportional to the distance from cameras due to alignment errors between the camera and the base where it is mounted. On the other hand, tracking loss in consecutive frames makes the error increases rapidly, so in around two seconds the virtual sensor have to be rejected, but until then, the threshold will increase to handle this situation, avoiding the rejection of the GPS data. Finally, the angle between the cameras and the tracked object makes the error in the depth axis increase significantly, and then, the threshold should be more tolerant. All these effects are contained in the following formula:

\[
    d_{th} = d_0 + K_d \cdot \max(d_1,d_2) + K_{TL} \cdot N_{TL} + K_\alpha \cdot |\cos(\alpha)| \quad ; \quad K_d,K_{TL},K_\alpha \geq 0
\]  \hspace{1cm} (18)

Here \(d_0\) is an offset distance, \(N_{TL}\) corresponds to the number of consecutive frames with tracking loss, and \(\alpha\) is the angle between the cameras and the tracked object, as shown in Figure 34. Angle and distance between the camera and the tracked object.
Let apply this function to the data represented in Figure 35. At the beginning the cameras are in parallel configuration, but in $t = 17$ s they reach the orthogonal configuration. Some tracking losses can also be observed (they are marked with blue and green dots). Figure 36 represents the distance between the simulated GPS data with an AWGN of 15 dB SNR and the estimated position, as well as the following threshold:

$$d_{th} = 0.25 + 0.1 \cdot \max\{d_1, d_2\} + 0.04 \cdot N_{FL} + 1 \cdot |\cos(\alpha)|$$  \hspace{1cm} (19)

![Figure 34. Angle and distance between the camera and the tracked object](image)

![Figure 35. Estimation error and distance with cameras with the cameras changing from parallel to orthogonal configuration](image)
As it can be seen, the threshold initially takes a high value because the cameras are in parallel from the beginning, but it decreases as they separate. The peak in $t = 35$ s corresponds to an instant with a high number of frames with tracking loss. The offset value for this threshold contains a constant term for rejecting noise and another term proportional to the distance to the cameras.
4. Simulation of perturbations in the virtual sensor over quadrotor trajectory control

4.1. Introduction

Consider a situation with three quadrotors A, B and C. Two of them, A and B, have cameras mounted in their base with known position and orientation referred to a global frame. Images taken from cameras are sent along with their position and orientation to a ground station. Both cameras will try to stay focused on the third quadrotor, C, so a tracking algorithm will be applied to obtain the centroid of the object on every received image. An external position estimator executed in the ground station will use this data to obtain an estimation of quadrotor C position that can be used for position or trajectory control in the case C does not have this kind of sensors, they are damaged, or they are temporarily unavailable. The situation described above has been illustrated in Figure 37. Here the cones represent the field of view of the cameras, the orange quadrotor is the one being tracked and the green ball corresponds to its position estimation.

One of the main issues in vision-based external position estimation applied to position control is the presence of delays in the control loop, which should not be too high to prevent the system of becoming unstable. The following sources of delay can be identified:

- Image acquisition delay
- Image transmission through radio link
- Image processing for tracking algorithm
- Position estimation and its transmission

The first two are imposed by hardware and available bandwidth. The last one is negligible in comparison with the others. On the other hand, image processing is very dependent on the computation cost required by the tracking algorithm. Before this simulation study, the external position estimation system was developed and tested with real data, obtaining position and
orientation of cameras and tracked quadrotor from a Vicon Motion Capture System in the
CATEC testbed. The visual tracking algorithm used was a modified version of CAMShift
algorithm [12], [13]. This color-based tracking algorithm uses Hue channel in the HSV image
representation for building a model of the object and detecting it, applying Mean-Shift for
computing the centroid of the probability distribution. As this algorithm is based only in color
information, a small orange ball was disposed at the top of the tracked quadrotor, in contrast
with the blue floor of the testbed. Figure 38 shows two images captured by the cameras during
data acquisition phase.

![Figure 38. Images taken during data acquisition experiments at the same time from both cameras, with two orange balls at the top of a Hummingbird quadrotor](image)

With the captured data, the estimation of the quadrotor position was obtained offline with an
extended Kalman filter. Graphical results showed three types of errors in the estimation: an
offset error about 15 cm between the real position (taken from Vicon) and the estimated one
due to bad calibration of the cameras, a small amplitude noise typical in any estimation
process, and the presence of outliers derived from faulty data in orientation measurement
caused by occlusions. These perturbations will be simulated so their effects over the trajectory
control of quadrotor can be analyzed.

### 4.2. Quadrotor trajectory control

Since this study is carried out by simulation, a dynamic model of the quadrotor and its control
system is required. Otherwise, using a real system for testing perturbation effects over
trajectory control would imply high risk of accidents and damages on the UAV.

This section presents the dynamic equations of the quadrotor, relating the three-dimensional
position and orientation variables with the control signals, as well as the control scheme that
allows the quadrotor to follow a path defined by a set of way-points.

#### 4.2.1. Dynamic model

As any other mechanical system, quadrotor dynamics can be derived from two ways [15]:
using Newton-Euler formulation or Euler-Lagrange formulation, although the first one requires
less computations and adding dynamic terms to the model is easier. In this case, the
relationship between translational and rotational accelerations of the rigid body and external forces and torques is given by:

\[
\begin{bmatrix}
    m I_{3\times3} & 0 \\
    0 & I
\end{bmatrix}
\begin{bmatrix}
    \dot{V} \\
    \dot{\omega}
\end{bmatrix}
+ \begin{bmatrix}
    \omega \times mV \\
    \omega \times I\omega
\end{bmatrix} = 
\begin{bmatrix}
    F \\
    \tau
\end{bmatrix}
\]

(20)

where \( m \) is the total mass of the quadrotor, \( I \) is the inertia matrix, assumed to be diagonal, \( V \) and \( \omega \) are the translational and rotational velocities, and \( F \) and \( \tau \) are the external forces and torques applied to the system. Only two kinds of external forces are considered here: gravity and the thrust of the motors denoted by \( T_i \), with \( i = 1, 2, 3, 4 \). Then, the rotational acceleration due to motor forces and gyroscopic terms is:

\[
\begin{align*}
    I_{xx}\ddot{\phi} &= \theta \dot{\psi}(l_{yy} - l_{zz}) + l(-T_2 + T_3) \\
    I_{yy}\ddot{\theta} &= \phi \dot{\psi}(l_{xx} - l_{zz}) + l(T_1 - T_3) \\
    I_{zz}\ddot{\psi} &= \phi \theta (l_{xx} - l_{yy}) + \sum_{i=1}^{4} (-1)^i Q_i
\end{align*}
\]

(21)

where \( \phi, \theta, \) and \( \psi \) are roll, pitch and yaw angles, \( I_{xx}, I_{yy} \) and \( I_{zz} \) are the inertia moments along the three axes (the inertia products are zero if the mass distribution is symmetric), \( l \) is the length of the arms and \( Q_i \) is the counter-torque generated by the \( i \)-th motor. On the other hand, the translational accelerations can be computed as follows:

\[
\begin{aligned}
    m \ddot{z} &= mg - (c\phi \ c\theta) \sum_{i=1}^{4} T_i \\
    m \ddot{x} &= (s\psi \ s\phi + c\psi \ s\theta \ c\phi) \sum_{i=1}^{4} T_i \\
    m \ddot{y} &= (-c\psi \ s\phi + s\psi \ s\theta \ c\phi) \sum_{i=1}^{4} T_i
\end{aligned}
\]

(22)

There are three important considerations here: 1) quadrotor rotation is independent from position, while position depends on rotation, 2) quadrotor moves due to projection of total thrust of the motors over the horizontal plane, and 3) quadrotor is an under-actuated system with 6 DOF and only four control signals (the thrust of each motor).

Finally, it is assumed that motor thrust \( T_i \) and counter-torque \( Q_i \) are proportional to the square of its speed:

\[
T_i = b \cdot \Omega_i^2 \quad ; \quad Q_i = d \cdot \Omega_i^2
\]

(23)

57
4.2.2. Attitude and height control

Although in practice controllers actuate over motors speed, four control signals are defined in the following way:

\[
U = \begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} = \begin{bmatrix}
b \cdot (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
b \cdot (\Omega_2^2 + \Omega_3^2) \\
b \cdot (\Omega_1^2 - \Omega_4^2) \\
d \cdot (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)
\end{bmatrix}
\]

(24)

These signals allow to control quadrotor height, roll, pitch and yaw angles, respectively, in terms of rotor speeds, decoupling control problem despite de gyroscopic term in (21).

Looking at (21), if these terms are rejected, orientation dynamic can be approximated by a double integrator. The same assumption can be done in (22) for small angles. Gyroscopes and accelerometers are combined for having a good estimation of rotational position, but they usually also provide rate information in roll, pitch and yaw angles. That is why a position and velocity feedback scheme [26], [27] can be applied, so references in angular speed and their corresponding control signals will be given by:

\[
\begin{bmatrix}
\phi_d \\
\theta_d \\
\psi_d
\end{bmatrix} = \begin{bmatrix}
K_p^\phi \cdot e_\phi \\
K_p^\theta \cdot e_\theta \\
K_p^\psi \cdot e_\psi
\end{bmatrix}; \quad \begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix} = \begin{bmatrix}
K_p^\phi \cdot (\phi_d - \phi) \\
K_p^\theta \cdot (\theta_d - \theta) \\
K_p^\psi \cdot (\psi_d - \psi)
\end{bmatrix}
\]

(25)

This technique however cannot be applied for height control because in general, for small UAVs, speed on Z axis is not an available signal. Instead, height control signal can be decomposed in the sum of a constant term for compensating quadrotor weight, and a correction term that considers height error:

\[
U_1 = mg + u_h(t)
\]

(26)

It was found that designing a discrete time controller able to handle discontinuities in height references, like steps, was a hard task. Moreover, continuous time controllers that provided stable response became unstable when they were discretized. Ref. [28] proposes an optimal digital controller for double integrator system with a sample time of 0.1 seconds, suitable for height control, which will be used in this work. This controller was tested in simulation with good results.
4.2.3. Velocity control

The velocity control in the XY plane is now considered, maintaining a null reference for yaw angle. Let \((x_d, y_d)\) be the desired point and \((x, y)\) the current position of the quadrotor. Then, the direction of the velocity in the global frame can be computed as follows:

\[
\Psi = \tan^{-1}\left(\frac{y_d - y}{x_d - x}\right)
\]  \hspace{1cm} (27)

The quadrotor will try to maintain a constant speed \(V\), although accepting speed transitions when the direction changes between waypoints. Then the velocity error on XY axes is computed as follows:

\[
e_{Vx} = V \cdot \cos(\Psi) - \dot{x}
\]
\[
e_{Vy} = V \cdot \sin(\Psi) - \dot{y}
\]  \hspace{1cm} (28)

Assuming a null reference for yaw angle, then speed on X axis will depend on pitch angle, while speed on Y axis will do in roll angle. Simulation experiments show that a simple proportional controller provides good results, so roll and pitch references are obtained in the following way:

\[
\phi_d = K_p^\Psi \cdot e_{Vy}; \quad \theta_d = K_p^x \cdot e_{Vx}
\]  \hspace{1cm} (29)

4.2.4. Trajectory generation

A trajectory \(\Gamma\) will be defined as a sequence of N way-points, where each way-point specifies the XYZ desired coordinates that the quadrotor should reach:

\[
\Gamma = \{wp_1, wp_2, \ldots, wp_N\}; \quad wp_n = [x_n^d, y_n^d, z_n^d]^T
\]  \hspace{1cm} (30)

The criterion used for jumping from current way point to the next one was to consider a threshold distance \(d_{th}\), so the way-point counter will be incremented if the following condition is satisfied:

\[
\sqrt{(x_n^d - x)^2 + (y_n^d - y)^2 + (z_n^d - z)^2} < d_{th}
\]  \hspace{1cm} (31)

No interpolation polynomials were used in this work, so quadrotor will follow segments of linear trajectories.
4.3. Model of perturbations

This section presents the five perturbations considered in the external position estimation that is sent to the quadrotor for its trajectory control.

4.3.1. Sampling rate

As any digital system, the external estimator will receive external observations at discrete time instants. Two working modes can be considered: periodic sample rate or variable sampling time. In general, quadrotor position control system will work at higher frequencies that the external estimator, particularly if this functional block makes use of computer vision techniques.

Because quadrotor control system requires position data at fixed rate, a sampling rate conversion process is then needed. The simplest solution is the use of Zero Order Holder, so last update obtained from estimator will be provided until the next update is received. Previous experiments to this work show that the sampling rate for visual-based external position estimation is around 5-20 Hz, depending on image resolution and available bandwidth for its transmission, while quadrotor sample rate for position control is typically between 1-10 Hz for GPS and 100-200 Hz for a Vicon Motion Capture System.

4.3.2. Delay

Here the delay is defined as the elapsed time between the reception in the quadrotor of the estimation of its position and the moment when data used for this estimation was captured. In the case of image based estimation, the delay includes image acquisition and transmission, visual tracking algorithm application, estimation computation and its transmission to the UAV.

Height is a critical magnitude in quadrotor control, particularly sensitive to delays. In many cases, small size UAVs have ultrasonic devices mounted in their base, so it is not too hard assumption to consider its internal measurement. In practical, ultrasonic devices used with quadrotors usually have a delay less than 100 ms. In most cases, delay can be considered to be equal to the sampling time.

4.3.3. Noise

Position estimation process will be affected by a number of perturbations that cause the estimation to be slightly different from real value. When the position estimation is computed using a pair of images and the position and orientation data of the cameras themselves, the main contributions to the error are distortion in non-well calibrated lenses, noise in the measurement of position and orientation of the cameras and errors in the centroid of the
tracked object. On the other hand, an offset error in the measurement of the orientation of the cameras causes an estimation error in the position proportional to the distance between the cameras and the tracked object.

In this work, noise will be simulated as an additive random variable, with normal or uniform distribution function, characterized by its mean and variance or maximum and minimum. The offset error in the position estimation obtained after the experiments was around 0,15 m, with a mean distance of 4 m between the cameras and the quadrotor. The standard deviation of the noise will strongly depend on the number of consecutive frames with tracking loss in one or both cameras. In normal conditions, a value of 0,05 m can be assumed.

4.3.4. Outliers

An outlier can be easily identified as a point that is located far away from the rest of estimation points. It has short duration in time, and it will appear at each sample period with a certain probability. Therefore two parameters can be associated: the amplitude of the outlier, and its probability. This kind of perturbation was identified during analysis of the results of image based position estimation, after the application of extended Kalman filter, and it was caused by outliers in cameras orientation measurement and inconsistencies in centroids in the pair of images. As an extended Kalman filter was used for estimation, the outliers will be partially filtered, but amplitudes around 25 m can be considered, while an acceptable value for the outlier probability is 0,05.

4.3.5. Packet loss

Position estimation is externally estimated by an agent that receives images from the cameras as well as the position and orientation of the cameras, and will send its estimation via radio link to the tracked object. In practical, it should be considered the possibility of the loss of one or more packets containing this estimation. For this simulation, we consider a probability of packet loss at each sample time, and a uniform random variable that will represent the number of packets lost. Values of 0,01 for packet loss probability and a uniformly distributed random number between 1 and 10 in the number of packets lost can be used for reference.

4.4. Simulation results

Graphical and numerical results of simulations in different conditions are presented here, showing separately the effects of perturbations indicated in previous section so it will be easier to interpret their effects on the performance of trajectory control. These results are only for reference. Effects of perturbations will obviously depend on the control structure being used, although the one presented in Section 3 is a conventional control scheme.
4.4.1. Different speeds and delays

Delay in position measurement and quadrotor speed will be increased until instability is reached, without considering any other perturbation. The sample time for quadrotor control is 20 ms, except in height control, where it was used 100 ms period. Figure 39 shows the reference path and the trajectories followed for different values of delay in XY position measurement and fixed delay of 100 ms in height when the speed of the quadrotor is 0,5 m·s$^{-1}$. A threshold distance of $d_{th} = 0,25$ m was specified for jumping between way-points.

![XY plane trajectory with XY position measurement delay](image)

Figure 39. Reference path and trajectories followed in XY plane with different values of delay in XY position measurement, fixed delay of 100 ms in height measurement and $V = 0,5$ m·s$^{-1}$

Table 2 relates different speeds of the quadrotor with maximum delays allowed in XY position and in height measurements. In the first case, instability is associated with the existence of a loop that catch the quadrotor, preventing it to follow the trajectory, while the criterion used for height was an overshoot greater than 50%. These results were obtained for the reference path shown in Figure 39. The effect of delay in height control is independent from quadrotor speed, but in the first three cases, height limits the maximum delay allowed when a full XYZ external position estimation is considered.

Table 2. Maximum values of delays in XY position and height measurement for different references of speed

<table>
<thead>
<tr>
<th>Speed [m·s$^{-1}$]</th>
<th>XY plane max. delay [s]</th>
<th>Height max. delay [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,25</td>
<td>2,4</td>
<td>0,2</td>
</tr>
<tr>
<td>0,5</td>
<td>0,94</td>
<td>0,2</td>
</tr>
<tr>
<td>0,75</td>
<td>0,44</td>
<td>0,2</td>
</tr>
<tr>
<td>1</td>
<td>0,18</td>
<td>0,2</td>
</tr>
</tbody>
</table>
The threshold distance $d_{th}$ between the current quadrotor position and the next way-point can be used to compensate delay. Then, the higher the threshold is, the lower the effect of the delay over the trajectory will be, although this value is limited by the minimum distance between way-points. Quadrotor speed will also be limited by the maximum error allowed in the trajectory tracking or in the position control for a certain application where reducing delay is not possible. However, the speed of the UAV might be adjusted depending on the distance to the next way-point, specifying higher speeds when the distance is large enough, and decreasing it as the quadrotor approaches to the way-point.

4.4.2. Noise and outliers with fixed delay

In this simulation it will be applied an additive Gaussian noise with 0,2 m² variance and outliers with uniformly distributed amplitude in the range ±10 m and a probability of 0,1 over position measured in XY plane. It will be maintained a delay of 0,25 s in this measurement, and 0,1 s on height. Figure 40 shows the external XY position estimation used by the trajectory controller. The reference path was the same.

In Figure 41 it can be seen the effects of noise and outliers over trajectory control in the XY plane. It has also been represented the trajectory followed without this two perturbations, considering only the delay. There is an important change on the behavior when noise is applied over estimation. Outliers instead are filtered by the low-pass dynamics of the quadrotor, so their effect is not very significant in comparison due in part to the lack of a derivative term in the velocity controller, as seen in (29). Trajectory control can be therefore enhanced if position estimation is filtered before being used by controller.
4.4.3. Delay, noise, outliers and packet loss

Finally, all perturbations presented in Section 4 will be considered together, introducing now a packet loss probability of 0.02 with a uniform random number of packets lost between 1 and 10. These perturbations are only applied over XY position estimation because, as mentioned above, the height is assumed to be measured with internal sensors. Figure 42 represents the
trajectory followed by the quadrotor for speeds of 0.5 and 0.75 m·s⁻¹. As quadrotor trajectory controller needs position measurements every sample period, when a packet is lost, the system will use the last measurement received. This implies serious problems if the last measurement was an outlier and the number of packets lost are large enough, causing strange behaviors. The sudden change in the direction of the quadrotor in Figure 42 for $V = 0.5$ m·s⁻¹ around $(x, y) = (1.7, -1.1)$ may be due to the coincidence of both perturbations.

On the other hand, height control is not affected by these perturbations despite the coupling between XY position and height through the control signals, as motors thrust is used for controlling these three variables. In Figure 43 the Z-axis reference and the real height are shown in the conditions described above and $V = 0.75$ m·s⁻¹.
The simultaneous application of all perturbations considered in this study has an unacceptable effect over the trajectory control. Delay and packet loss have a similar influence as the second one can be considered as a random delay. Outliers, if they are not consecutive, are filtered by the low-pass quadrotor dynamics, and they can be easily detected too. On the other hand, noise is the main cause of the bad results in the trajectory tracking shown in Figure 42. There are two ways to decrease its effect: improving the position estimation, which depends on the tracking algorithm and on the position and orientation measurements, or considering control methods more robust to noise.
5. Conclusions

In this work, a vision-based position estimation system with applications to multi quadrotor systems has been developed and tested, obtaining graphical and numerical results in different conditions. The system considered here contains two cameras, whose position and orientation referred to a global frame are known, that are visually tracking a moving object, a quadrotor in this case, whose position is externally estimated using an Extended Kalman Filter. This estimation, what can be considered as provided by a virtual sensor, is used for a Fault Detection, Identification and Recovery application. In normal conditions, the visual estimation should be quite similar to the one given by sensors internal to the UAV, such as a GPS. However, if in a certain situation the internal sensor fails due to temporal loss of the signal or permanent damage of the device, the distance between both estimations will start to increase. In such situations, if the visual estimation is considered to be reliable (what can be known taken into account the number of available cameras without tracking loss), then the error can be detected simply defining a threshold distance. Once this threshold is exceeded, a reconfiguration process in the control scheme makes possible to keep the UAV under control, using the virtual sensor as position sensor. However, the provided estimation is affected by a number of perturbations, including delays, noise, outliers and packet loss, whose effects should be analyzed by simulation previous to test the system in real conditions due to the high risk of accidents and damages in the vehicle or in persons.

The study presented here can be divided into three parts:

- The position estimation. A number of data acquisition experiments were carried out in the CATEC testbed using their Vicon Motion Capture System for measuring the position and orientation of both cameras and the tracked quadrotor (the position given by Vicon is used as ground truth and for simulate GPS data). The images captured by the cameras were synchronized with the Vicon data through a time stamp that indicates the time instant when they were acquired. Then, an off-line estimation process is applied, making possible the debug of the estimation system more comfortably. In a first stage, the images are passed to the tracking algorithm in order to obtain the centroid of the tracked object for both cameras. Then, the centroid along with the position and orientation of the cameras are used by the Extended Kalman Filter for obtaining the estimation of the quadrotor position. The estimation error can be computed, as the real position was also obtained from Vicon. A large set of experiments were done considering different situations, making possible the identification of the main causes that contribute to increase the estimation error. As result, it was found that the estimation error depends strongly on the relative position of the cameras (if the optical axes are in parallel the error in the depth axis is high) and the number of frames with tracking loss when the estimation became monocular or there were no available cameras.
• The Fault Detection and Identification. As commented, the real position of the quadrotor given by Vicon was also used to simulate a GPS signal internal to the UAV. Typical GPS errors such as additive error or lock-in-place fault were simulated in order to verify that a fault detection system based on variable threshold is able to detect internal failures, what requires to considering the visual estimation as reliable. Experiments show that the number of consecutive frames with tracking loss can be used for rejecting the visual estimation simply defining a constant threshold. However, the FDI threshold, defined as the minimum distance between the position estimation given by visual methods and internal sensors such that if it is exceeded then the internal sensors are considered to be failed, should take into account the variable accuracy of the visual estimation. As known, the estimation error will increase as the optical axes of the cameras become parallel. On the other hand, the error is proportional to the mean distance between the cameras and the quadrotor due to errors in the orientation measurement. Finally, it also has to consider temporal tracking losses, offset errors and noise in the visual position estimation.

• The analysis of perturbations of the visual estimation over the quadrotor trajectory control. As next step of this project, the visual position estimation will be used to control the UAV. However, a simulation study of the perturbations introduced by this virtual sensor should be done in order to avoid potential accidents. In this sense, delays play an important role as image acquisition, transmission and processing can be significant. Simulation results have been obtained in multiple conditions, considering different delays in the measurement of the position, speeds of the tracked UAV, outlier probability, amplitude of the noise or packet loss. Although the effects of all these perturbations will depend on the control structure being used, the results obtained can be used as reference for future work.
REFERENCES


