

Acknowledgments

Contents

List of Figures

Chapter 1

Preliminaries and state of the art

In this chapter, some mathematical preliminaries that will be used throughout the document are presented. The first part is focused on the image processing. After it is explained the basic concepts of projective geometry, and it is made an analysis of the model geometry. Although this work considers that the intrinsic and extrinsic parameters are known, an introduction of the techniques of calibration of the camera is written in this part. In the second section of the chapter there is an overview of the mini-helicopter four-rotor UAVs, and in the same way the dynamic model is detailed.

1.1 Computer Vision

1.1.1 Projective geometry

1.1.2 Model geometry

1.1.3 Camera calibration

1.1.4 Formation of the image in projective geometry

1.2 UAVs: Mini-helicopter of four rotors

An Unmanned Aerial Vehicle (UAV) is an autonomous flying machine, this means that the vehicle can perform flights without a human pilot by the usage of an autonomous pilot algorithm. While the concept of UAV seems a novelty, its usage is not. In fact, the interest in the study of these type of vehicles has increased, and there are many works published especially related to control laws applied on the topic. The research on this subject is also possible due to the evolution of information and communication technologies and the power computation of the different devices.

It should be noted that the ballistic or semi-ballistic missiles, the cruise missiles and the artillery projectiles are not considered UAV systems.

An UAV system is composed mainly by two elements:

- Flight component: formed by the aerial vehicle.
- Ground component: formed by the ground station, which allows the communication between this and the aerial system, data processing, monitoring, control, etc.

In general the drones can be classified according with the size, the application or a combination of both, however, the size is the most used criterion to make the classification. The classification, depicted in Fig. ?? is resumed in the next items, ?:

- Micro-UAV: This kind of vehicles can be operated by just one person. The propulsion is electrically made and due to the low cost of the materials, the vehicles are primarily used for civilian applications. The average autonomy is about 30 minutes.
- Mini-UAV: The equipment necessary for these vehicles is more specialized. In general, these drones fly at speeds of $70km/h$ and medium altitudes of $3,5km$. Their size allows them to fly around 4 hours.
- MALE/HALE: The Medium Altitude High Endurance (MALE) drones are used to perform higher endurance flights and also at higher altitudes ($10 - 15km$). The High Altitude High Endurance (HALE) drones can fly at about $20km$ of altitude. These drones are able to perform missions with medium endurance of two days. The Predator (MALE) and the Global Hawk (HALE) are the most popular drones of this category and are mainly applied for military missions.
- UCAV: The Unmanned Combat Air Vehicle is mainly used for offensive missions. They are under research and only some prototypes are available, like the RQ-170 Centinel.

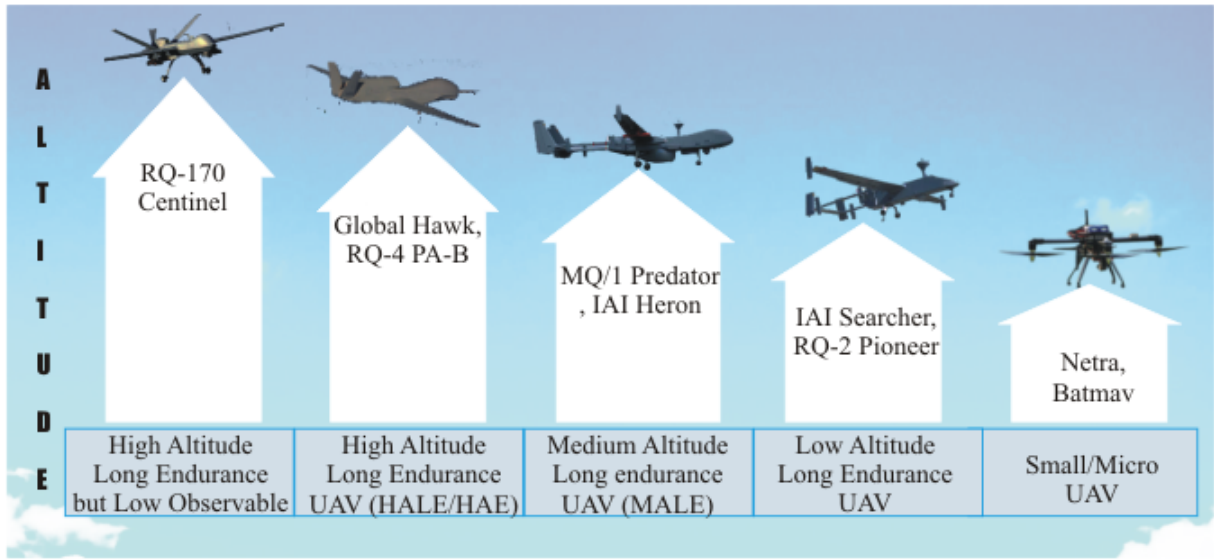


Figure 1.1 – The different categories of drones.

Due to their aerodynamical function, there is an alternative classification, where it is possible to find three families: fixed wings aircrafts, rotating wings aircrafts and the swing-wing aircrafts. A study shows that the research has mainly focused on the rotating wings aircrafts with VTOL (Vertical Take Off and Landing) configuration. These vehicles are capable of take off and land in a vertical way and due to this advantage, the range of applications is very extensive, being used on military and civilian applications, like surveillance, rescue, aerial mapping, etc.

1.2.1 Vertical Take Off and Landing (VTOL) vehicles

1.2.2 Rotorcraft

- **Helicopter:** is a type of rotorcraft in which lift and thrust are supplied by rotors. This allows the helicopter to take off and land vertically, to hover, and to fly forward and laterally. These attributes allow helicopters to be used in congested or isolated areas where fixed-wing aircrafts would usually not be able to take off or land. Fig. ?? shows an example of a helicopter.
- **Multirotor:** also called multicopter, is a rotorcraft with more than two rotors. An advantage of multirotor aircraft is the simpler rotor mechanics required for flight control. Unlike single and double-rotor helicopters, multirotors often use fixed-pitch and yaw blades; control of vehicle motion is achieved by varying the relative speed of each rotor to change the thrust and torque produced by each one. Due to their ease of both construction and control, multirotor aircraft are frequently used in radio control aircraft and UAV projects, in which the names **tricopter**, **quadcopter** see



Figure 1.2 – An Apache attack helicopter

Fig. ??, **hexacopter** and **octocopter** see Fig. ??, are frequently used to refer to 3, 4, 6 and 8-rotor helicopters, respectively.



(a)



(b)

Figure 1.3 – (a)Quadcopter and (b)octocopter configurations, courtesy of (?)

- Autogyro: is also known as gyroplane, gyrocopter or rotaplane. Is a type of rotorcraft that uses an unpowered rotor in autorotation to develop lift, and an engine-powered propeller, similar to that of a fixed-wing aircraft, to provide thrust. While similar to a helicopter rotor in appearance, the autogyro's rotor must have air flowing through the rotor disc to generate rotation. Fig. ?? shows an example of an autogyro.



Figure 1.4 – Autogyro MT-03 in flight

- Gyrodyne: is a type of VTOL aircraft with a helicopter rotor-like system that is driven by its engine for take off and landing and also includes one or more conventional propellers to provide forward thrust during cruising flight. Lift during forward flight is provided by a combination of the rotor, like the autogyro, as well as conventional wings.

Fig. ?? shows an example of gyrodyne.



Figure 1.5 – Gyrodyne AM-X3 in flight

1.2.3 Powered lift

- Convertiplane: is an aircraft which uses rotor power for vertical take off and landing and converts to fixed-wing lift in normal flight. These vehicles may be divided into two broad classes, based on whether the rotor is fixed as in a helicopter or tilts to provide thrust in forward flight. Fig. ?? shows an example of a convertiplane.



Figure 1.6 – Convertiplane

- Tail-sitter: called also tailsitter, is a type of VTOL vehicle that takes off and lands on its tail, then tilts horizontally for forward flight. Fig. ?? shows an example of tail-sitter UAV's.
- Lift jets: is an auxiliary jet engine used to provide lift for VTOL operation, but may be shut down for normal wing-borne flight.



Figure 1.7 – Tail-sitter prototypes, as part of the Googleâ€™s project Wing for delivery

- Lift fans: is an aircraft configuration in which lifting fans are located in large holes in an otherwise conventional fixed wing or fuselage. It is used for V/STOL¹ operation. The aircraft takes off using the fans to provide lift, then transitions to fixed-wing lift in forward flight. Several experimental craft have been flown, but only the F-35 Lighting II entered into production, see Fig. ??



Figure 1.8 – F-35 Lightning II combat aircraft during take off

The present work is centered on the analysis and study of multirotors, particularly on the modelling and control of a quadcopter carrying a manipulator arm. However, during the development of the project, the hexacopter configuration was also used. It allowed the obtention of some interesting results.

1.3 Multi-rotors: state of the art

The most common vehicle with the capacity of take off and landing in vertical way is the standard helicopter, which is composed of a principal rotor and a rear rotor. However, the multirotors, where we can find the four rotor helicopter, quadrotor or quadcopter, the hexacopter or hexarotor and the octocopter or octorotor have been the center of interest for many works in the last years, see ?, ? and ?. Some of the advantages offered by these vehicles are its symmetry, which makes it easy to design and build. The usage of four,

1. Vertical and/or Short Take-Off and Landing

six or eight rotors provides improved stability on hover, because the distributed pushing forces are acting at the same distance from the center of mass instead of just one pushing force acting on the whole center of mass. Also, the propellers can be protected by the frame of the prototype, which makes its usage indoor safer, face to the non-protected propellers of the standard helicopter.

The modeling of this kind of vehicles has done possible the design of many control laws which allow the attitude and position stabilization with good results. Some of the most used control laws have been: backstepping, found on [1, 2 and 3]; sliding mode control has also been applied to these vehicles, and it can be found on [4, 5 and 6]; linear control laws (PD and PID), they can be seen at [7, 8 and 9].

Other interesting projects developed with multi-rotors can be found on the domains of: navigation, where embedded cameras and/or different sensors are used to know the relative position of the aerial system, see [10 and 11]; fault-tolerant control, where fault detection on the actuators is implemented and a control law is proposed for the aerial vehicle, see [12 and 13].

1.3.1 Characteristics and operation of the mini-UAV

This part of the chapter is devoted to the mathematical modeling of the quadrotor. First, a general description of the function of the system is presented. After that, the mathematical modeling is treated, and equally the relation between motors, propellers and the dynamics of the system.

The quadrotor or four rotor helicopter is a mechatronic system composed of a cross structure. At each end of the cross we find a propeller coupled to a motor, and at the center of the configuration all the electronic elements are found (power source, computer, etc). Compared to the classical helicopter, this system does not have main rotor and the control is performed by the angular velocity change on each rotor, [14].

The four rotors are composed of the propellers coupled to DC motors or DC Brushless motors (BLDC). Such a platform is represented in Fig. 1.1, where the front and rear motors (1 and 2) rotate clockwise, while the other two (3 and 4) rotate counterclockwise. In this way, gyroscopic effects and aerodynamic torques tend to cancel each other out in trimmed flight.

Each rotor produces a force f_i parallel to its rotation axe, as well as a drag torque Q_i , opposite to the direction of rotation. The total force or total thrust acting on the helicopter (parallel to the z_b axis) is the addition of the four forces generated by each rotor ($F_T = f_1 + f_2 + f_3 + f_4$). The combination of these forces and the drag torques allow the angular motions over the main axes of the helicopter. Consequently, three movements for the position are produced, see Fig. 1.2.

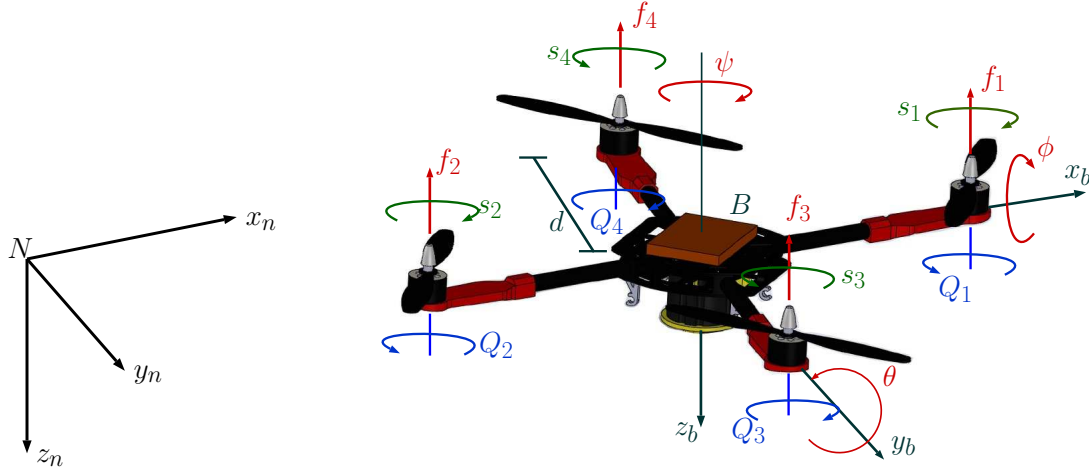


Figure 1.9 – Scheme of the quadrotor configuration: inertial reference frame $N(x_n, y_n, z_n)$, the reference body fixed frame $B(x_b, y_b, z_b)$, the f_i forces on each motor, angular velocity of the motors s_i and the reaction torques Q_i .

- **Roll (ϕ):** It is produced by the difference $f_3 - f_4$. To obtain this, the velocity of the right motor m_3 is increased/reduced, while the velocity of the motor m_4 is equally decreased/increased. This difference of forces produces a torque Γ_ϕ around the axis x_b .
- **Pitch (θ):** It is produced by the difference $f_1 - f_2$. It is obtained similarly using the front and rear motors m_1 and m_3 . This difference of forces produces a torque Γ_θ around the axis y_b .
- **Yaw (ψ):** It is the combination of all the reactive torques, $Q_1 + Q_2 - Q_3 - Q_4$. It is obtained by decreasing/increasing the speed of the front and rear motors while decreasing/increasing the speed of the lateral motors. In other words, if a difference of speed between the motors turning in the opposite direction is produced, the reactive torques produce a torque Γ_ψ around the axis z_n .
- **Vertical displacement on the x_n axis:** To go forward or back, the rotational speed of motor m_2 must be decreased/increased, while decreasing/increasing the rotational speed of the motor m_1 .
- **Lateral displacement on the y_n axis:** To go to the right or left, the rotational speed of the lateral motors m_4 and m_3 must be decreased/increased.
- **Displacement on the z_n axis:** To go up or down, the torque of all the rotors m_i must be decreased/increased. In the absence of disturbances, the aerial system can perform a hover at a certain height by having a zero translation speed. Then, the

total thrust F_T must balance the weight mg of the aerial system by pointing its direction in the axe z_b .

The first three movements are considered for the body fixed frame, and the last three for the inertial reference frame.

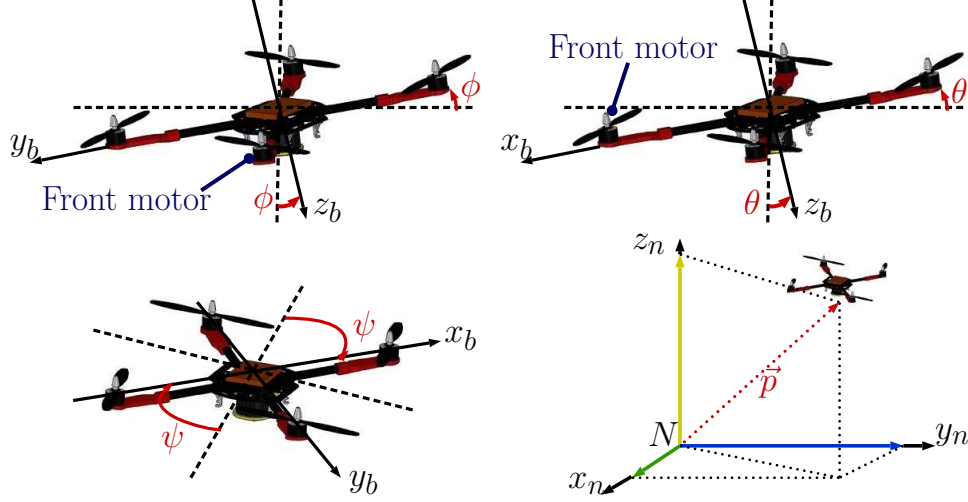


Figure 1.10 – Roll (ϕ), pitch (θ), yaw (ψ) and space displacement

From the description of the different angular movements and the vertical and horizontal displacements, it is visible that the position of the system depends on its attitude. In this way, the displacement of the quadcopter can be controlled from the attitude control.

1.4 Position representation

Robot tasks are often defined by the use of Cartesian coordinates. Let consider the scheme in Fig. ???. It is possible to specify the coordinates of the point p with respect to either frame $o_0x_0y_0$ or frame $o_1x_1y_1$.

Geometrically, a point corresponds to a specific location in space, and a vector specifies a direction and a magnitude. Vectors can be used, for example, to represent displacements or forces. Therefore, while the point p is not equivalent to the vector v_1 , the displacement from the origin o_0 to the point p is given by the vector v_1 . Under this convention, it is clear that points and vectors are not equivalent, since points refer to specific locations in space, but a vector can be moved to any location in space. Then, two vectors are said to be equal if they have the same direction and the same magnitude.

When assigning coordinates to vectors, the same notational convention is used as when assigning coordinates to points. Thus, v_1 and v_2 are geometric entities that are invariant with respect to the choice of coordinate systems, but the representation by coordinates of these vectors depend directly on the choice of reference coordinate frame.

Using this convention, an expression of the form $v_1^1 + v_2^2$ where v_1^1 and v_2^2 are as in Fig. ??, is not defined since the frames $o_0x_0y_0$ and $o_1x_1y_1$ are not parallel. Thus, a clear need appears, not only for a representation system that allows points to be expressed with respect to various coordinate systems, but also for a mechanism that allows to transform the coordinates of points that are expressed in one coordinate system into the appropriate coordinates with respect to some other coordinate frame.

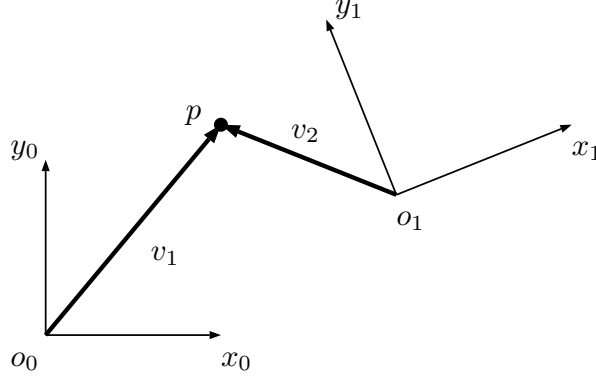


Figure 1.11 – Two coordinate frames, a point p , and two vectors v_1 and v_2 , (?).

1.5 Rigid body attitude representation

Consider two orthogonal right-handed coordinate frames: the body coordinate frame, $B(x_b, y_b, z_b)$ located at the center of mass of the rigid body and the inertial coordinate frame, $N(x_n, y_n, z_n)$, located at some point in the space (for instance, the earth NED frame). This coordinate system is showed in Fig. ??. In general, the origin of B is chosen to coincide with the gravity center (CG) of the body.

The body attitude in the space can be represented in many ways, each one with their advantages and disadvantages, depending mainly on the application.

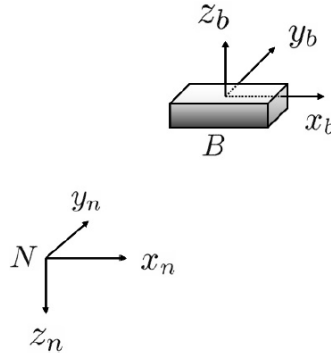


Figure 1.12 – Inertial and mobile frame of a rigid body.

Definition 1.5.1 *The movement of a rigid body with reference frame B relative to a rigid body or reference frame N is called a simple rotation of B in N , if there is a line L , called rotation axis, where the orientation relative to B and N keeps the same between the start and the end of the movement.*

Theorem 1.5.2 (*Rotation Euler Theorem*). *Every relative change in orientation between two rigid bodies with the two coordinate systems B and N can be produced by one simple rotation of B over N .*

Let \vec{b} and \vec{r} be the coordinates of a vector \vec{X} in B and N respectively. Vector \vec{b} can be written in terms of vector \vec{r} .

Let $\vec{e} = [e_1 \ e_2 \ e_3]$ be a unit vector collinear to the rotation axis L around which B is rotated by an angle β . Consequently, \vec{b} is obtained by

$$\vec{b} = \cos \beta \vec{r} + (1 - \cos \beta) \vec{e} \vec{e}^T \vec{r} - \sin \beta \vec{e} \times \vec{r} \quad (1.1)$$

In fact, the coordinates \vec{b} and \vec{r} are linked by means of the following linear transformation:

$$\vec{b} = C \vec{r} \quad (1.2)$$

Matrix C can be taken as an operator which takes a fixed vector \vec{r} expressed in N and is expressed in B . From (??)

$$C = \cos \beta I_3 + (1 - \cos \beta) \vec{e} \vec{e}^T - \sin \beta [\vec{e}^\times] \quad (1.3)$$

where I_3 represents the identity matrix of dimension three and $[\xi^\times]$ represents the skew-symmetric matrix, given by:

$$[\xi^\times] = \left(\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} \right)^\times = \begin{pmatrix} 0 & \xi_3 & \xi_2 \\ -\xi_3 & 0 & \xi_1 \\ \xi_2 & -\xi_1 & 0 \end{pmatrix} \quad (1.4)$$

Matrix $C \in \mathbb{R}^{3 \times 3}$ identifies the orientation of the moving frame B with respect to the inertial frame N and it allows the coordinate transformation of a vector system into another one. This matrix is known as the direction cosine matrix (DCM), rotation matrix or attitude matrix.

Rotation matrix

Rotation matrix C belongs to the subspace of orthogonal matrices of dimension three, called special orthogonal group, denoted by $S0(3)$, and defined by

$$S0(3) = \{C \mid C \in \mathbb{R}^{3 \times 3}, C^T C = I_3, \det(C) = 1\} \quad (1.5)$$

In a rotation matrix C , each element c_{ij} is a direct cosine, given by:

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \quad (1.6)$$

where

$$c_1 = \begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \end{pmatrix} \quad c_2 = \begin{pmatrix} c_{12} \\ c_{22} \\ c_{32} \end{pmatrix} \quad c_3 = \begin{pmatrix} c_{13} \\ c_{23} \\ c_{33} \end{pmatrix} \quad (1.7)$$

Consequently,

$$C = \begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix} \quad (1.8)$$

where

$$c_i^T c_i = 1 \quad \text{and} \quad c_i^T c_j = 0 \quad \forall i \neq j \quad (1.9)$$

Rotational velocity

Suppose that a rotation matrix C is time varying, so that $C = C(t) \in S0(3)$ for every $t \in \mathbb{R}$. Assuming that $C(t)$ is continuously differentiable as a function of t . An argument identical to the one in the previous section shows that the time derivative $\dot{C}(t)$ of $C(t)$ is given by

$$\dot{C}(t) = [\vec{\omega} \times] C(t) \quad (1.10)$$

where $\vec{\omega}$ is the angular velocity of the rotating frame with respect to the fixed frame at time t .

Euler angles and Roll, Pitch and Yaw angles

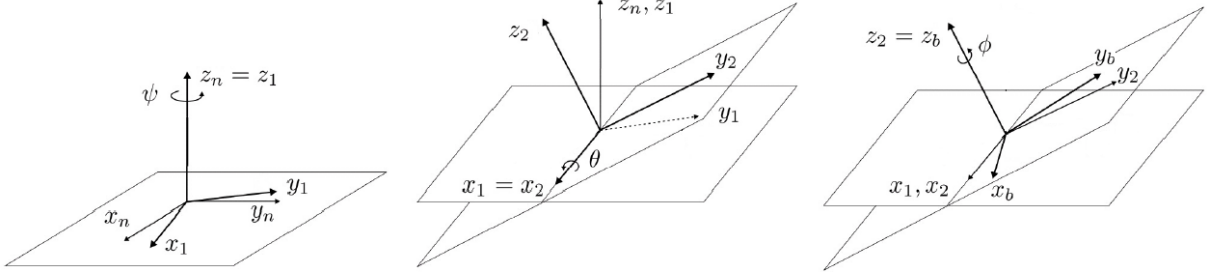


Figure 1.13 – Euler angles.

A common method of specifying a rotation matrix in terms of three independent quantities is to use the so-called Euler Angles (ψ, θ, φ) . Consider again the fixed coordinate frame $N(x_n, y_n, z_n)$ and the rotated frame (x_b, y_b, z_b) , shown in Fig. ??.

It is possible to specify the orientation of the frame (x_1, y_1, z_1) relative to the frame $N(x_n, y_n, z_n)$ by the three angles, and it can be obtained by three successive rotations as follows: first rotate about z -axis by the angle ϕ , next, rotate about the current y -axis by the angle θ . Finally, rotate about the current z -axis by the angle ψ . In terms of the basic rotation matrices the resulting rotational transformation C_1^0 can be generated as the product

$$\begin{aligned}
 C_1^0 = C_{z,\phi}, C_{y,\theta}, C_{z,\psi} &= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} \quad (1.11)
 \end{aligned}$$

A rotation matrix C can also be described as a product of successive rotations about the principal coordinate axes x_n , y_n and z_n taken in a specific order. These rotations define the roll, pitch and yaw angles. The order of rotations is specified as $z - y - x$, in other words, first a yaw about z_n by an angle ϕ , then pitch about the y_n by an angle θ , and finally roll about the x_n by an angle ψ . Since the successive rotations are relative to

the fixed frame, the resulting transformation matrix is given by

$$\begin{aligned}
 C_1^0 = C_{z,\phi} C_{y,\theta} C_{x,\psi} &= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \\
 &= \begin{bmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}
 \end{aligned} \tag{1.12}$$

Now, let $\vec{\omega} = [\omega_x \ \omega_y \ \omega_z]$ be the angular velocity of the body in the reference frame B with respect to the reference frame N . Then, the kinematic equation is given by ?:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan \theta \sin \phi & \tan \theta \sin \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{1.13}$$

Quaternions

The drawbacks of the angle/axis representation can be overcome by a different four-parameter representation; namely, the unit *quaternion* or Euler parameters, defined by:

$$q := \begin{pmatrix} \cos \frac{\beta}{2} \\ \vec{e} \sin \frac{\beta}{2} \end{pmatrix} = \begin{pmatrix} q_0 \\ q_v \end{pmatrix} \in \mathbb{H} \tag{1.14}$$

where

$$\mathbb{H} = q|q_0^2 + \vec{q}^T \vec{q} = 1, \quad q = \begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix}, \quad q_0 \in \mathbb{R}, \quad \vec{q} \in \mathbb{R}^3 \tag{1.15}$$

where $\vec{q} = (q_1 \ q_2 \ q_3)^T \in \mathbb{R}^3$ and $q_0 \in \mathbb{R}$ are known as the vector and scalar parts of the quaternion respectively. The identity quaternion and the conjugated quaternion are given by:

$$q_{id} = [1 \ 0^T]^T \quad \bar{q} = [q_0 \ -\vec{q}^T]^T \tag{1.16}$$

Since the quaternion is unitary, $q^{-1} = \bar{q}$. The product of two quaternions $q_1 = [q_{1_0} \ \vec{q}_1^T]^T$ and $q_2 = [q_{2_0} \ \vec{q}_2^T]^T$ is defined by:

$$q_1 \otimes q_2 = \begin{pmatrix} q_{1_0} & -\vec{q}_1^T \\ \vec{q}_1 & I_3 q_{1_0} + [\vec{q}^\times] \end{pmatrix} \begin{pmatrix} q_{2_0} \\ \vec{q}_2 \end{pmatrix} \tag{1.17}$$

Now, let b_q and r_q be the quaternions associated to vectors \vec{b} and \vec{r} , defined by:

$$b_q = [0 \quad \vec{b}^T]^T \quad r_q = [0 \quad \vec{r}^T]^T \quad (1.18)$$

These two quaternions are linked by the next relation:

$$b_q = q \otimes r_q \otimes q^{-1} = q \otimes r_q \otimes \bar{q} \quad (1.19)$$

Rotation matrix C can be expressed in terms of quaternions by:

$$C = C(q) = (q_0^2 - \vec{q}^T \vec{q}) I_3 + 2(\vec{q} \vec{q}^T - q_0 [\vec{q}^\times]) \quad (1.20)$$

from where

$$C(q) = \begin{pmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_0 q_1 + q_2 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_2 q_3 - q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{pmatrix} \quad (1.21)$$

The relation between the vectors \vec{b} and \vec{r} is:

$$\vec{b} = C(q) \vec{r} \quad (1.22)$$

Denoting by $w = (\omega_1 \quad \omega_2 \quad \omega_3)^T$ the angular velocity vector of the body coordinate frame, B relative to the inertial coordinate frame N expressed in B , the kinematics equation is given by

$$\begin{pmatrix} \dot{q}_0 \\ \dot{\vec{q}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\dot{\vec{q}}^T \\ I_3 q_0 + [\vec{q}^\times] \end{pmatrix} w = \frac{1}{2} \Xi(q) w \quad (1.23)$$

Rigid motions

Definition 1.5.3 A rigid motion is an ordered pair (d, C) where $d \in \mathbb{R}^3$ and $C \in SO(3)$. The group of all rigid motions is known as the **Special Euclidean Group** and is denoted by $SE(3)$. It is denoted that $SE(3) = \mathbb{R}^3 \times SO(3)$.

A rigid motion is a pure translation together with a pure rotation.

Homogeneous transformations

The combination of position and orientation representations allows the formulation of homogeneous transformations, used in the modeling of arm manipulators. Then, the trans-

formation matrices with the form:

$$H = \begin{bmatrix} C & d \\ 0 & 1 \end{bmatrix} \quad (1.24)$$

are called **homogeneous transformations**. Then, a homogeneous transformation is a representation of a rigid motion where the system $S0(3)$ is used interchangeably to represent the set of rigid motions and the set of all 4×4 matrices given in (??).

A set of basic homogeneous transformations generating $S0(3)$ is given by

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_a & -s_a & 0 \\ 0 & s_a & c_a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.25)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{y,b} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.26)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,a} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.27)$$

for translation and rotation about the x, y, z axis respectively.

The most general homogeneous transformation that is considered is given by

$$H_1^0 = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28)$$

In the above equation $n = (n_x, n_y, n_z)^T$ is a vector representing the direction of X_1 in the $o_0x_0y_0$ system, $s = (s_x, s_y, s_z)^T$ represents the direction of y_1 , and $a = (a_x, a_y, a_z)$ represents the direction of z_1 . $d = (d_x, d_y, d_z)^T$ represents the vector from the origin o_0 to the origin o_1 expressed in the frame $o_0x_0y_0z_0$.

Attitude error

Two orientations of a rigid body are considered, described by rotation matrices C_1 and C_2 respectively. Then, the relative attitude between these two orientations is computed by:

$$C_r = C_1^{-1}C_2 \quad (1.29)$$

In fact, C_r represents an operator of orientation which rotates C_2 about C_1 . From here, the relative orientation is used in the estimation and in the orientation control as *attitude error*. With this, $C_d = C_1$ is the desired attitude of a rigid body and $C = C_2$ is the real attitude of the body. Consequently, the attitude error is computed by:

$$C_e = C_d^{-1}C \quad (1.30)$$

If the attitude error is zero, then, $C_e = I_3$. When the unitary quaternion is used to represent the attitude of the body, the relative orientation between q_1 and q_2 is expressed by:

$$q_r = q_1^{-1} \otimes q_2 = \begin{pmatrix} q_{1_0} & \vec{q}_1^T \\ -\vec{q}_1 & I_3 q_{1_0} - [\vec{q} \times] \end{pmatrix} \begin{pmatrix} q_{2_0} \\ \vec{q}_2 \end{pmatrix} = \bar{q}_1 \otimes q_2 \quad (1.31)$$

The geodesic metric is given by the next expression:

$$\beta_r = 2|\arccos(q_{r_0})| \quad (1.32)$$

This metric represents the smallest angle of rotation between attitude q_1 and the attitude q_2 . The Euclidean distance between the two unitary quaternions gives an approximation of the geodesic metric:

$$\frac{2}{\pi}\beta_r \leq 2 \| q_1 - q_2 \|_2 \leq \beta_r \quad (1.33)$$

Also, when β_r is enough small, it is possible to do the next approximation:

$$\beta_r \approx 2 \| q_1 - q_2 \|_2 \quad (1.34)$$

For the case of the attitude control law, if $q_d = q_1$ is the desired attitude of the body and $q = q_2$ is the real attitude of the body, the attitude error is given by:

$$q_e = q_d^{-1} \otimes q = \bar{q}_d \otimes q \quad (1.35)$$

When the attitude error is zero, the error quaternion has two possible values:

$$q_e = [\pm 1 \ 0]^T \quad (1.36)$$

This is due to the non bijection quaternion with the group $SO(3)$.

1.6 Conclusions

«««< current

Chapter 2

Pose estimation using a single camera

UAV autonomous navigation into corridors or tunnels is a hard task due mainly to the localization problem. GPS measurements are usually dropped noised or unavailable. Cameras become a popular sensors for pose estimation even if sometimes is hard to compute on-board the algorithm. This chapter is centered on the estimation of the relative position of the vehicle through a single image in real-time execution. This information is useful since it serves to make an indoor navigation (in a corridor). The algorithm designed is intended to use the minimum of resources of the computer. Although this first stage focuses on using tools such as matlab, the algorithms can be easily implemented in an embedded system. So the program can be resume as follow: firstly the virtual image is obtained. Then the edges of the image are extracted. The rotation matrix of the camera is estimated using infinite vanishing points and finite vanishing points. A new sub-classification of lines is made from the finite vanishing point. This subset is done in four quadrants. Using the finite vanishing point and the information of each quadrant, we can obtain the principal line from each corner. The collinearity property is used with pairs of major lines. If there is a collinearity between each pair of lines the system is centered. Otherwise, the result is a percentage of displacement in the y -axis, z -axis. The scale is a percentage of displacement where the center of the corridor indicates a zero percentage of displacement. A numerical validation is made in order to verify this algorithm.

2.1 Pose estimation algorithm

The proposed vision algorithm takes into account the perspective projection, the vanish point identification and the collinearities of the principal lines in the image to estimate the pose of an aerial vehicle. Ideas from ?? and ? have inspired our algorithm.

The general scheme of the estimation pose algorithm can be seen in Figure ?? . As can be observed in this figure, once the image is acquired a line detector algorithm is applied. For making a subset of vertical, horizontal and diagonal lines a histogram is then used. After that, the rotation matrix is acquired and the line classification algorithm is again applied for extracting principal lines in the image. This information is used with the collinearity approach for estimating the pose of the camera. Therefore it is relatively easy compute the y and z -axis in the image plane.

Figure ?? shows the general procedure, and this depicts too the vanishing point evolution. For obtaining the pose estimation it is necessary to know the camera model, next subsections will introduce this concept and each block in the general scheme algorithm.

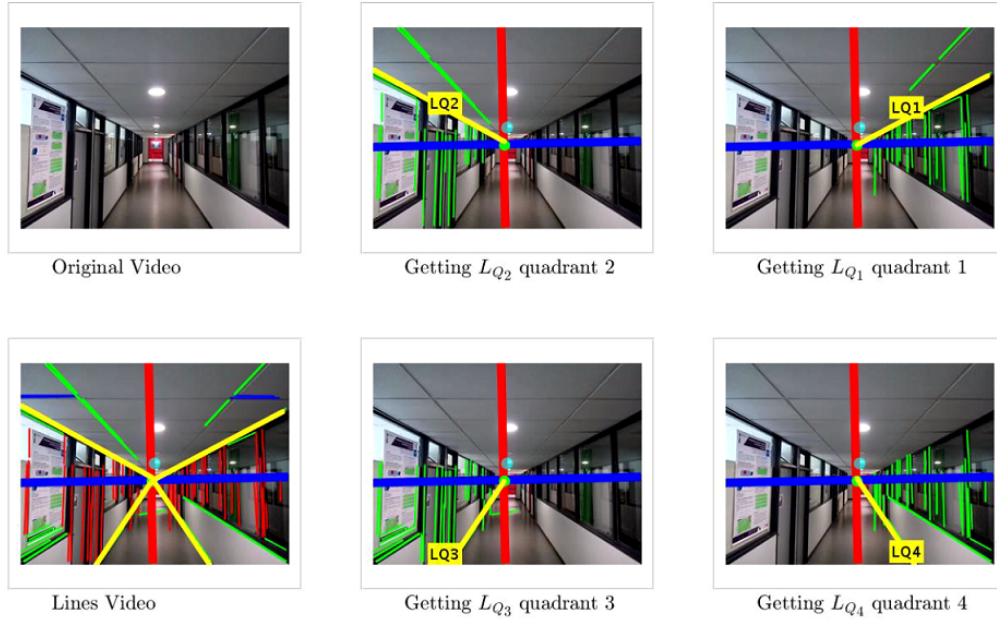


Figure 2.1 – These images introduce the applied methodology. The scene is a corridor and an image is taken from the video to extract a set of lines that could describe geometrically the environment and the rotation matrix. Four new subsets of lines are made taking into account the vanishing point (quadrant 1, 2, 3 and 4). The information of each quadrant is used to obtain the principal lines (L_{Q_1} , L_{Q_2} , L_{Q_3} , L_{Q_4}).

2.1.1 Vanishing point algorithm

Perspective projection of parallel lines in world space creates vanishing lines in image space that intersect in a point called vanishing point. If the parallel lines are parallel to the image plane, the vanishing lines are also parallel to this plane and the vanishing point is at an infinite distance from the image center, hence is called infinite vanishing point. Otherwise, it is a finite vanishing point. For our case the camera calibration is based on

the following assumption: the main vanishing lines in the input image correspond to three orthogonal directions in world space as can be seen in Figure ??.

2.1.2 Dominant lines detection

In man-made indoor environments, it is possible to extract line segments, texture, colors and so on. In this work, the linear time line segment detector called EDlines ? is used giving accurate results and with high time process (all the segments for an image 240×320 are computed in 3 *ms*). This algorithm no requires parameter tuning and run ups faster than others methods as the Hough transform ?.

2.1.3 Lines classification

RANSAC (Random Sample Consensus) has become a simple and powerful method to provide a partition of parallel straight lines into clusters by pruning outliers. The algorithm begins selecting randomly two lines for generating a VP hypothesis. These lines are grouped together in order to optimize the estimated VP and for obtaining a dominant VP. When a dominant VP is obtained, the process finishes. Nevertheless, several finite vanishing points can be extracted from the environment with this algorithm.

In the context of indoor navigation, the main orthogonal directions in environments consist generally in a vertical one (often associated with an infinite VP) and two horizontal ones (associated with finite or infinite VP). In this work, the heuristic strategy is taken to extract a limited number of reliable VP while enforcing the orthogonality constraint, in conjunction with RANSAC.

For a robust selection of VPs, an histogram is made in order to classify all the set of lines in a subset of vertical lines L_v , a subset of horizontal lines L_h and a subset of diagonal lines L_d . For clustering lines by RANSAC, the consensus score is quite different depending on each subset and is computing with (??). Unlike the finite VP whose coordinates may be determined in the image plan (lines L_d), the infinite VPs are generally represented with a direction (lines L_v , L_h). For finite VP, the consensus score is based on a distance between the candidate straight line and the intersecting point and represented as (??). For infinite VP, it is obtained with (??) using the angular distance between the direction of the candidate straight line and the direction representing the infinite VP, more details see ?.

$$score = \sum_{i=0}^n \Upsilon(v, l_i) \quad (2.1)$$

$$\Upsilon(v, l_i) = \begin{cases} 1 & d(v, l_i) < \delta \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

$$\Upsilon(v, l_i) = \begin{cases} 1 & \widehat{(\vec{v}, \vec{l}_i)}, (\vec{l}_i, \vec{v}) < \delta \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

where n is the number of dominant lines and $d(v, l_i)$ denotes the Euclidian distance from the finite VP candidate v to the line l_i . The lines whose distance is below a fixed threshold δ are considered as participants. (\vec{v}, \vec{l}_i) describes the angle between the infinite VP direction from the image center and the line l_i to test in image space.

2.1.4 Rotation matrix estimation

It is known that when using the VPs intrinsic and extrinsic camera parameters can be estimated. We consider that the intrinsic parameters are known thus we will focus on compute the others ones, in particularly the Rotation matrix. The main point p of the camera is set to the center of the image plane. The rotation matrix $(\vec{u}, \vec{v}, \vec{w})$ transforms points from real world to the image plane. Its columns are the vectors of the world coordinates frame expressed in the camera space. The directions of the three vanishing points from the optical center of the camera are assumed to be orthogonal. Thus without loss of generality, the following relations hold for the final calibrated.

$$f > 0 \quad (2.4a)$$

$$\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{w} = \vec{w} \cdot \vec{u} = 0 \quad (2.4b)$$

$$\vec{u} = \vec{v} = \vec{w} \quad (2.4c)$$

The obtained structured lines from the corridor image provide enough information to estimate the camera rotation matrix. Normally, it is supposed that the camera has a vertical position, and as the corridor is plenty of vertical and horizontal lines, then it is possible to find a finite vanishing point and two infinite vanishing points.

A finite vanishing point, two infinite vanishing points

This case occurs when two axes of the world frame are parallel to the image plane. Fig ?? introduces the terms involved for the camera calibration using one finite vanishing point

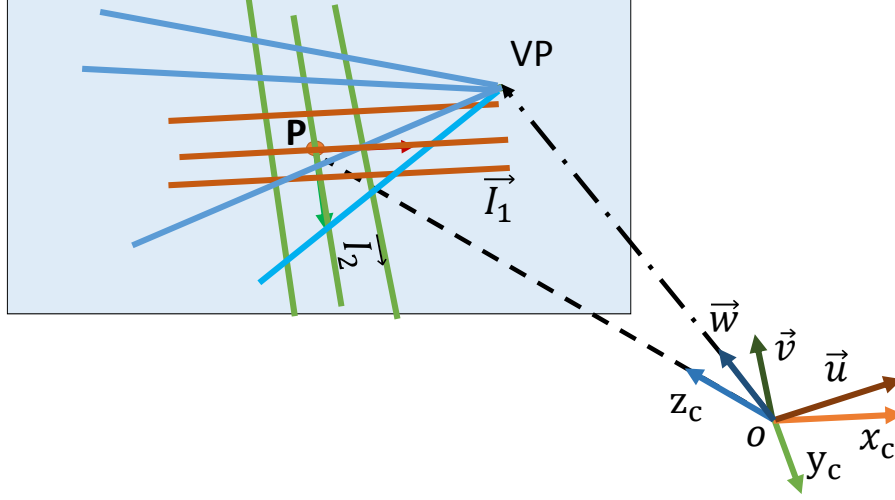


Figure 2.2 – Camera calibration with one finite and two infinite vanishing points. (x_c, y_c, z_c) is the camera coordinate frame, $(\vec{u}, \vec{v}, \vec{w})$ the world coordinate frame, VP the finite vanishing point, \vec{I}_1 and \vec{I}_2 the infinite vanishing point directions. P is the main point.

$\overrightarrow{OV\vec{P}} = (vp_x, vp_y, -f)^T$ and two infinite vanishing points with directions $\vec{I}_1 = (I_{1x}, I_{1y}, 0)^T$ and $\vec{I}_2 = (I_{2x}, I_{2y}, 0)^T$.

The vector \vec{w}' defines a non-normalized form of \vec{w} , and is computed from the finite vanishing point as

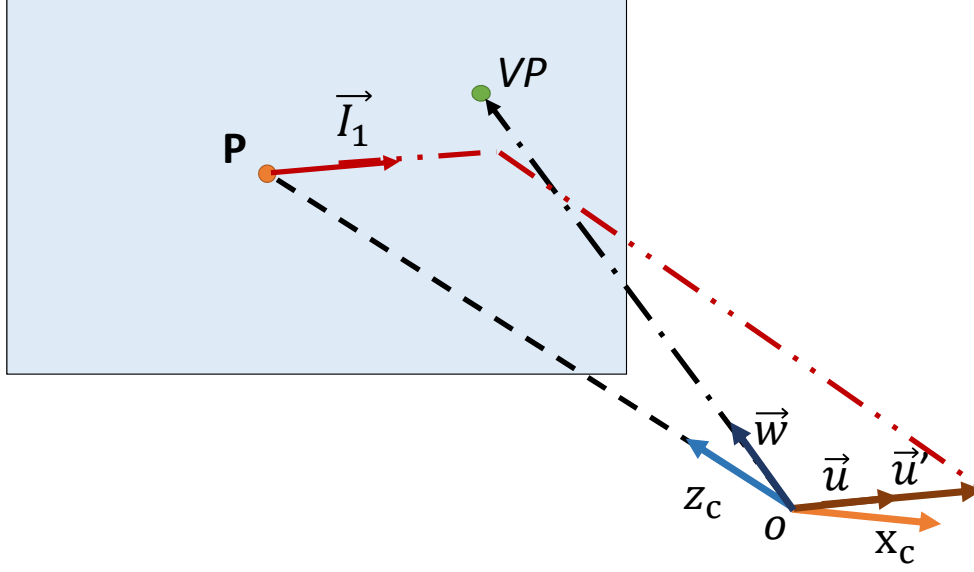
$$\vec{w}' = (w'_x, w'_y, w'_z) = \overrightarrow{OV\vec{P}} = (vp_x, vp_y, -f) \quad (2.5)$$

The coordinate axis \vec{u} , as indicated in Figure ??, lies on the plane defined by the points O , P and the direction $\vec{I}_1 = (I_{1x}, I_{1y}, 0)^T$. Then \vec{u}' , the non-normalized version of \vec{u} , can be expressed as $\vec{u}' = (I_{1x}, I_{1y}, u'_z)$. The goal is that \vec{u} and \vec{w} belong into an orthogonal coordinate frame, then

$$\vec{u}' \cdot \vec{w}' = I_{1x}w'_x + I_{1y}w'_y + u'_zw'_z = 0 \quad (2.6)$$

therefore

$$u'_z = \frac{I_{1x}w'_x + I_{1y}w'_y}{f} \quad (2.7)$$


 Figure 2.3 – \vec{u} estimation from the infinite vanishing points.

Similar procedures can be used for $\vec{v}' = (I_{2x}, I_{2y}, v'_z)$ (see Figure ??), thus

$$v'_z = \frac{I_{2x}w'_x + I_{2y}w'_y}{f} \quad (2.8)$$

Typically the vertical lines are plenty and more precise in many images, so it retains \vec{v}' and \vec{w}' and compute \vec{u}' using a cross product, $\vec{u}' = \vec{v}' \times \vec{w}'$. The rotation matrix of the camera $R = (\vec{u}, \vec{v}, \vec{w})$ from Eq. (??) which meets the conditions from Eq. (??) is obtained by normalizing \vec{u}' , \vec{v}' and \vec{w}' .

2.1.5 Principal lines estimation

colorblue

Once the camera rotation matrix obtained, fusing this information with L_v , L_h and L_d thus, the principal lines of the corridor (corners of the the corridor) could be estimated. The algorithm results in four new subsets (quadrants 1,2,3,4) relatives with to the vanishing point and the rotation matrix. Therefore, the information in each quadrant is useful to find the main line of each subset. Each subset of lines contains the beginning and end point of each line as well as its inclination and its length.

The principal lines which are in the border of the right wall-ceiling L_{Q_1} , right wall-floor

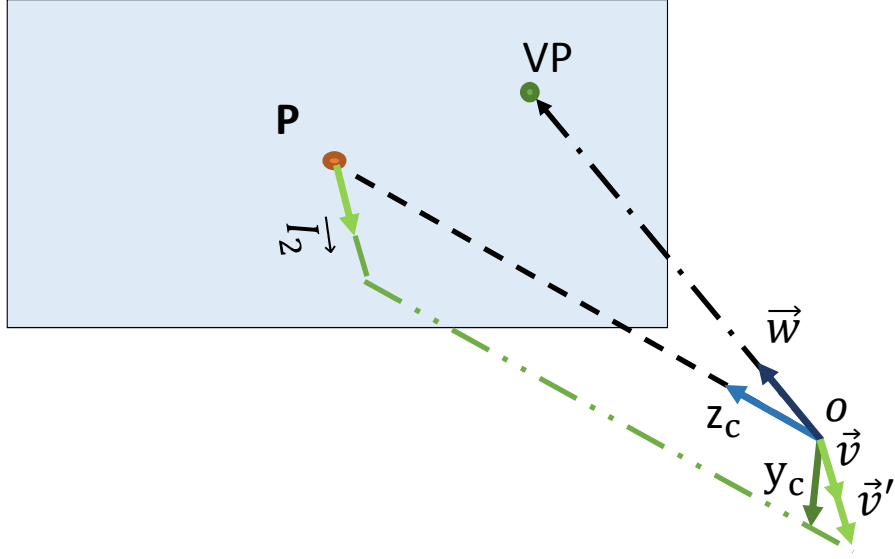


Figure 2.4 – \vec{v} estimation from the infinite vanishing points.

L_{Q_2} , left wall-ceiling L_{Q_3} and left wall-floor L_{Q_4} are depicted in Fig ??.

$$C_m \equiv \begin{bmatrix} \omega_n \sum x_n^2 & \omega_n \sum x_n y_n \\ \omega_n \sum x_n y_n & \omega_n \sum y_n^2 \end{bmatrix}$$

(x_n, y_n) are the initial and final points from each line, C_m represents the covariance matrix, m represents the frames and

$$\omega_n = e^{\frac{-0.5\theta^2}{\sigma^2}}$$

$$\theta = \arctan \frac{x_n a_{m-1} + y_n b_{m-1}}{x_n a_{m-1} - y_n b_{m-1}}$$

Where ω_n is the weight that supply the response variance to a constant value, a_{m-1}, b_{m-1} represents initial line hypothesis and the .

$$y = LQ_k = -\frac{b_m}{a_m}x$$

where the vector proper $\alpha = [a_m, b_m]^T$ extracted from C_m represents the principal line of each quadrant with $k = 1, 2, 3, 4$.

Sigo sin entender esta ultima parte que me enviaste podrias explicarlo en espanol?

2.1.6 Pose camera estimation

Position camera estimation is obtained using the previous results and the relation with the borders of the corridor. Hence, when the camera is in the center of the corridor, there is a collinearity between the lines L_{Q_1}, L_{Q_3} and L_{Q_2}, L_{Q_4} that can be expressed as

$$CP_{1,3} = L_{Q_1} \wedge L_{Q_3} \quad (2.9)$$

$$CP_{2,4} = L_{Q_2} \wedge L_{Q_4} \quad (2.10)$$

The third component of $CP_{1,3}$ and $CP_{2,4}$ supplies the information to obtain the relative camera position and it is given as

$$y_r = CP_{1,3} - CP_{2,4} \quad (2.11)$$

$$z_r = CP_{1,3} + CP_{2,4} \quad (2.12)$$

where y_r and z_r are the relative position with respect to the center of the corridor.

Las figuras que pusiste y como las pusiste no se ven, tienden a confundir y hace mas complicado el asunto para entender, te voy a poner un bosquejo de como quedaria mejor.. para esto necesito una secuencia de estas figuras separadas (cada figura separada..)

2.2 Numerical validation

2.3 Conclusions

=====

»»»> before discard

Chapter 3

Position control law and velocity estimation of an UAV

3.1 Position control law

3.2 Velocity estimation

3.2.1 Observer

3.2.2 Kalman filter

3.3 Conclusions

Chapter 4

Platform and experimental validation

This chapter is devoted to the presentation of the experimental platform, including the MOCA room at GIPSA-lab, ground station, position estimation system and developed aerial platforms, as well as the presentation of the experimental results and their consequent analysis.

First, to implement the position control law, the linear position of the system must be known, for this, the MOCA room was used.

This one is composed by 12 cameras and a ground station, which allows the computation of the control inputs through MATLAB/ Simulink and sends them to the system through radio signals. More details about this system are equally given in this chapter.

During the developmen of the work, different platforms were used in order to prove the algorithms comprising the different stages of the project.

First, a Flexbot micro quadcopter was used in order to test the attitude and position control laws. The general performance of this model was good, however, when the camera system was added to test the position algorithm, the platform was not able to take off due to its reduced specifications. As a result of this, it was decided to move to a bigger platform.

The second platform was the Flexbot micro-hexacopter. The technical specifications of flight controller board and motors were similar to the first one, but this model also offered two extra actuators and a bigger battery, in order to obtain more carrying capacity and autonomy.

With this new model, new experimental tests were performed and the behaviour of the system was improved. The motors used until this point of the project were DC motors and their operational life is not so long, therefore, the performance of these ones were reduced drastically with each test. Finally, and due to the precedent problems, all the

system was changed. Two different frames were totally designed and 3D printed. The flight controller board was changed, with better specifications in terms of processor and general performance. The DC motors were changed to brushless motors, to increase the power and carrying capacity. Consequently, the use of speed controllers for the motors was needed. The propellers were enlarged according to the specifications of the motors as well as the size of the battery. All the used platforms are described in the next subsections, some characteristics of each one are given, but only the last two, will be deeply detailed. After that, the experimental results are presented.

4.1 Moca Room and ground station

In order to test and to compare the proposed algorithm, it was made the platform integrated with the Vicon system and a terrestrial station . The Vicon system is used to obtain the orientation and the real position of the system. The ground station is concerned with calculating the position control and the velocity estimation algorithms. The image processing, which give us the pose of the system using the environment geometry, is done too in this part of the platform. The complete platform is useful to compare the results obtained with the proposed algorithm. In continuation it will be discussed in detail each part of this:

4.1.1 Moca Room

The motion acquisition is made through infrared cameras with emitters and receivers of infrared light and also through reflecting markers attached to the moving objects or individuals. The MOCA room is composed of 12 VICON© cameras (T40 series), attached to a metal structure in high and pointing their vision towards a common area. There are also 8 digital cameras pointing to the same area, but these ones are used for objects reconstruction or motion capture by image processing. With this system it is possible to compute the position and attitude up to 100Hz. Fig. 5.1 shows an image of the MOCA room and the reflecting markers.

A VICON camera is an infrared camera, which emits and receives infrared rays. A set of cameras pointing towards a common area is able to detect a reflective marker. The markers are little balls of retro-reflecting materials going from 0.5 to 2cm. of diameter. The cameras emit a very special light which makes the receivers sensitive only to this one, when a marker is placed in the area covered by the cameras, it creates a single point in the plane of each one of the cameras (if the area is well covered). Then, the information is collected in a computer running the VICON© tracker software. Fig. 5.2 shows an image of the used VICON cameras and the VICON tracker environment.

4.1.2 Ground station

The ground station is composed by three computers: the first one is under the real time MATLAB/Simulink© environment, the second one a target PC, which is under the xPC target tool- box, as well as a radio-frequency emitter and The estimated states (attitude and position) are sent to MATLAB/Simulink through a UDP frame every 2ms. From these data, the position control algorithm is computed and implemented in real-time at 200Hz on the target PC, which uses the xPC target toolbox. xPC Target also manages communications between the host and target PC, as well as the different inputs/outputs of the real-time application. The control variables are finally sent back to the system through a GIPSA-lab's built-in bridge that converts UDP frames to DSM2 protocol. For this, the radio-frequency emitter is used. Fig. 5.3 shows an overview of the computing process.

4.2 Experimental platforms

4.3 Experimental results

4.4 Conclusions

Chapter 5

Conclusions and Future work

5.1 Conclusions

5.2 Future Work

Appendix A

Vanishng point

Appendix B

RANSAC

Appendix C

Linear least squares regression
