

TRAJECTORY CONTROL AND MODELLING FOR WIND TURBINE MAINTENANCE BY USING A RPAS

Paulo ANTUNES^{1*}, Kouamana BOUSSON², Juan Antonio GARCÍA-MANRIQUE³

^{1, 2} Department of Aerospace Sciences, Universidade da Beira Interior, Calçada Fonte do Lameiro, Covilhã, Castelo Branco 6201-001, Portugal

³ Department of Mechanical Engineering and Materials, Universitat Politècnica de Valencia Ringgold, Camino Vera, s/n 5E, Valencia, Valencia 46022, Spain

Received 16 November 2017; accepted 19 January 2018

Abstract. One of the most demanded assets nowadays is energy. Over many options to generate it, humankind must seek sustainable ways; therefore, renewable energies must be empowered. Moreover, wind provides great benefits, granting uncalculated power at our disposition. Since this task is executed by big structures, their maintenance represents a difficult task for human efforts to achieve, because the height requires much support, effort and time to accomplish.

A Remotely Piloted Aircraft System (RPAS) can be adapted to perform surveillance on different types of surfaces, presenting a comfortable way to execute it in dangerous and difficult to access spaces, providing safer methods, and bringing experience of qualified workers and technology together.

This paper will provide methods for generating a trajectory over Wind Turbine blades, relying on the specification of a Phantom 4. The main objective is to establish a path over this structure, based on the measurements of a specific model and incorporating all of the blades surface. The results were satisfying once the precision was inside the allowed deviation. Nevertheless, some issues might be improved such as the velocity between waypoints and the polynomial selected to define the trajectory.

Keywords: maintenance, RPAS, Phantom 4, wind turbines, simulation, composite materials.

Introduction

The quadcopter is a type of Remotely Piloted Aircraft System (RPAS) that can perform Vertical Take Off and Landing (VTOL). This provides the maneuverability that most of RPAS lack, not only for VTOL, but also for its dimensions that allow incorporating onboard sensors.

This is an under-actuated system with four inputs (roll, pitch, yaw and throttle) and six outputs. These systems have given proof that they are able to perform complex missions accurately and effectively (Gupte, 2012).

With this, is possible to operate in constrained and difficult to access spaces. With the technological development of such machines, comes an interest in adapting them for our everyday needs. Thus, machines come to give the help that we demand. This type of task is not only expensive, but also dangerous since most of them are either outdated or inadequate, like the rope descendent technique.

With the use of remotely controlled, semi-autonomous or autonomous operations, the monitoring and inspection

of big structures like wind turbine blades, telecommunication antennas, buildings, and others, can be brought to a new level of quality and economy (Morgenthal & Hallermann, 2016). This article presents an efficient simulation, of what might be a solution for upcoming years.



Figure 1. Inspection of a wind turbine conducted using the rope descendent technique (Nurm, 2017)

*Corresponding author. E-mail: opauloantunes@gmail.com

There is no doubt that renewable energy sources are the road that humankind must pursue. Choosing them over fossil ones is surely a greater step towards sustainability. Among other renewable energy sources, wind is used in large-scale electricity generation. The use of this technology has been increasing in recent years (IRENA, 2012). Aspects such as providing a mature technology, plus commercial aspects have enhanced its large-scale implementation. Facts shown by Global Wind Statistics, which shows that just in the year 2016, a power capacity of 13.926 MW was installed in Europe (Fried, 2017). The wind turbines in case are composed by a main tower, and three blades that contain airfoils produced over an irregular shape, as seen at the tip. To conduct surveillance over these structures, is a complex task.

More common than ever, these blades are produced from composite materials, which may be difficult to predict with accuracy its durability. (Esu, 2016) Moreover, structures with defects can decrease the energy harvesting by nearly 30% (Staffell, 2014).

This paper summarizes a model adapted to describe the RPAS' equations of motion. Based on it, two types of paths to be described for the RPAS were idealized, enabling to capture a series of photos that, with a high-definition camera, would allow performing an inspection using a software that use photogrammetry and computer vision algorithms to transform both RGB and multispectral images into 3D maps and models. Based on this idea, a simulation in MATLAB environment was created. Since the cameras need to be steady to capture high quality photos, decreasing the error associated with surveillance is a challenge, so only small angles were considered.

RPAS can have diverse applications such as: precision farming, archeology, photography and robot first response (Kovacs, 2016).

In this paper, a trajectory generation methodology developed by (Kumar, 2016) on which it is possible to plan a trajectory from an initial point to a desired one, is adapted. The main goals to be achieved in this study are:

- the trajectories must present an approach for real-world surveillance;
- the trajectories must obey the restrictions posed by



Figure 2. Damaged leading edge of a wind turbine blade (Motion, 2017)

- the dynamics and input of the quadcopter;
- the control law applied to this system must be able to re-plan the trajectory at each control update, and apply the new inputs to the first section of the loop;
- the trajectory presented must correspond to the trajectory desired in order to obtain precise visual tracking for the captures of the photos.

Therefore, the dynamic model which was used and which will provide an idea of the control inputs and their function when controlling the system will be presented. Moreover, the paper will present the defined control laws: the nested loop which allows to control the position and attitude of the RPAS built on some assumptions that were made in order to optimize the trajectory for the task it was meant for.

Based on these principles, some equations to obtain trajectory tracking were used with the desired position as a reference point. At the end, a problem proposal definition and the results obtained will be presented to gather with comments for some future works.

1. Dynamic model

The vehicle's movements can be defined by six degrees of freedom. In order to characterize them, the inertial frame (a) and body frame (i) were created. The inertial frame is composed by a_1 , a_2 and a_3 , being the positive part of a_3 pointing upwards. In addition, the body frame (i), which has the same principles or the inertial reference, with the origin latch onto the RPAS center of mass was defined, as seen in Figure 3.

To characterize the system, Z - X - Y Euler angles, which enable to model the rotation of the RPAS, will be used.

In addition, to define the vehicles attitude a rotation matrix, ${}^a R_i$, presented by equation (1), is established. With this it is possible to describe a vector from the inertial system, a , in the vehicle's body system, i :

$${}^a R_i = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}. \quad (1)$$



Figure 3. The defined coordinate systems

The $s\theta$ and $c\theta$, represent the $\sin(\theta)$ and $\cos(\theta)$ respectively.

The variable r is the geometric representation of the distance from the vehicle's body system (i) to the inertial frame (a).

The primary forces on the system are the ones produced by gravity, in the i_3 direction, next to the ones produced by the motors F_i , in the i_3 direction. To summarize this, equation (2) that represents the acceleration with respect to the center of mass is provided (Kumar, 2016):

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}. \quad (2)$$

The angular velocities are defined as p , q and r , and are related with the derivatives of roll, pitch and yaw angles, as seen in equation (3):

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (3)$$

1.1. Control inputs

In the defined referential system, the RPAS state is established on the basis of longitudinal, transversal and vertical positions, the first three elements in equation (4). Nevertheless, its orientation over the three axes should also be considered; these are represented in the last three elements of equation (4). With this, it is possible to obtain the quadrotor's state (Kumar, 2016):

$$q' = [x \ y \ z \ \phi \ \theta \ \psi]^T; \quad (4)$$

$$x' = [q \ \dot{q}']^T. \quad (5)$$

Regarding equation (5), it represents the rigid body state and is composed of the RPAS' state vector (q') and its rate of change (\dot{q}').

In this model, a series of simplifications were made. It is assumed that the thrust produced by the rotors can be swapped immediately. This is not completely accurate, since a dynamical system is characterized as being a system on which the effects of actions do not occur immediately (Kumar, 2016). It is also limited in terms of the vehicle's body rate by the gyroscopes and tracking controllers. In addition, the force produced by each motor, a_i , was limited, being the maximum value produced by all rotors and amounting to 1.5 times of the quadcopters weight.

Here $i = 1, 2, 3, 4$.

$$a_{\min} \leq a_i \leq a_{\max}. \quad (6)$$

Here a represents linear acceleration. The value of a_{\min} is either positive or zero, since it is assumed that this RPAS does not possess the ability to stop or reverse the direction of the propellers' rotation, which is also proved by previous reports (Hehn, 2011).

The maximum thrust is limited by peak motor torque. This is defined as the sum of vertical forces which considers only the forces produced by the motors plus the force of gravity applied on the vehicle:

$$\vec{F}_{Total} = \vec{F}_{motor} + \vec{F}_{gravity}. \quad (7)$$

From discretizing equation (7), it is possible to obtain equation (8), in which m represents the mass of the vehicle, g is the force of gravity, ω the angular speed of the motors and k_F the proportionality constant of the force produced by the motors (Kumar, 2016):

$$\sum_{i=1}^4 k_F \omega_i^2 - mg. \quad (8)$$

1.2. Equations of motion

The forces produced by motors create moments over the three-referential axis which, depending on the direction of spin, dictate the moment generated. Since motors 1 and 3 spin on the i_3 direction, they will produce a positive moment. Motors 2 and 4 spin on the i_3 direction, producing negative moments. The variable L represents the distance between the RPAS center of mass and the position of the motor, which can also be called arm's length. With this defined, the inertia matrix related to the moments produced by the components of the vehicle can be seen in Equation (9) (Kumar, 2016):

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (9)$$

1.3. Motor model

The motors were modeled by a test of steady-state fixed rotor in order to determine a constant of proportionality that multiplied by the angular speed would allow estimating the force produced:

$$F_i = k_F \omega_i^2. \quad (10)$$

Experimentation with a fixed rotor at steady-state shows that $k_F \approx 6.11 \times 10^{-3} \text{ N/rpm}$ (Kumar, 2016).

With the same approach, it is possible to determine a constant of proportionality to estimate the moment produced, as seen in Equation (11):

$$M_i = k_M \omega_i^2. \quad (11)$$

It was determined that the constant k_M is about $1.5 \times 10^{-9} \text{ N/rpm}$ (Kumar, 2016).

The components of angular velocity, relative to the body frame, are shown in Equation (12):

$$a \omega_i = p i_1 + q i_2 + r i_3. \quad (12)$$

2. Control

2.1. Control model

Rigid body dynamics has a slower response compared to motor dynamics. Incorporating it leads to a fifth order dynamic model that implies additional complexity without significant improvement in performance (Kumar, 2016).

Using Euler angles, it is possible to parameterize the orientation, position and velocity of the center of mass:

$$x = [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T. \quad (13)$$

Neglecting the parameterization by the position and velocity of the center of mass, the rotation matrix and angular velocity can be simplified to equation (14) (Kumar, 2016):

$$u = [u_1 \ u_2 \ u_3 \ u_4]^T. \quad (14)$$

where u_1 is the force from all the propellers, and u_2 , u_3 , and u_4 are the moments related to the body frame axis (Kumar, 2016).

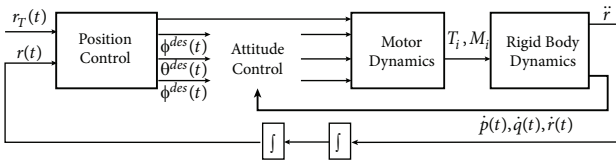


Figure 4. Nested control loops for position and attitude control (Kumar, 2016)

This nested loop, has an inner loop that corresponds to the attitude control and an outer loop that corresponds to the position control. In the inner loop the orientation is specified by the rotation matrix or a series of roll, pitch and yaw angles. From the angular rates, u_2 will be calculated. This is a function of the thrust and the moments that are obtained from the motor speeds. It will be calculated based on the desired attitude. In the outer loop, the aim is to obtain the position vector. This will be compared with the actual position and actual velocity that enables to obtain u_1 .

2.2. Attitude control

Since this is an initial approach, the waypoints are defined so that the quadcopter would complete an itinerary that contains all the blades in order to complete a visual survey with photos on them. To obtain a quality inspection, a high-quality photo which requires a stable flight is needed: no aggressive maneuvers and mostly possible at a near hover state. Thus, a control based on small angles is presented. This controller is based on the linearization of the equations of motion and motor models, at an operating point that corresponds to the nominal hover state (Kumar, 2016):

$$\theta \approx \phi \approx 0; \quad (15)$$

$$\psi \approx \psi_0. \quad (16)$$

Small angles of pitch and roll are considered:

$$c\phi \approx 1, c\theta \approx 1, s\phi \approx 0, s\theta \approx 0. \quad (17)$$

Therefore, it is possible to deduce the nominal values at a hover state for the first two inputs defined: the thrust of the motors, u_1 , and the quadcopter's attitude, u_2 :

$$u_{1,0} = mg; \quad (18)$$

$$u_{2,0} = 0. \quad (19)$$

Moreover, it is possible to assume that each rotor produces a quarter of the total thrust needed to maintain a hover:

$$F_{i,0} = \frac{mg}{4}. \quad (20)$$

As such, it is possible to deduce the nominal angular velocity produced by the motors, as shown in equation (21):

$$\omega_{i,0} = \omega_h = \sqrt{\frac{mg}{4k_F}}. \quad (21)$$

Thus, it is possible to establish a relation between the desired rotor speeds and the desired moments ($u_{2,des}$, $u_{3,des}$ and $u_{4,des}$):

$$u_{des} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_FL & 0 & -k_FL \\ -k_FL & 0 & k_FL & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_{1,des}^2 \\ \omega_{2,des}^2 \\ \omega_{3,des}^2 \\ \omega_{4,des}^2 \end{bmatrix}. \quad (22)$$

2.3. Trajectory tracking

For the position control, two position methods based on roll and pitch angles as inputs will be presented. Both methods are based on the linearization of the equations of motion, as shown above. The first method is for the nominal state, a hover, used for maintaining the position. The second one is designed to track and follow a desired trajectory, $r(t)$, over the three directions. In these two methods, the position control algorithm will determine the desired roll and pitch angles, ϕ_{des} and θ_{des} , which can be used to compute the commanded accelerations (Kumar, 2016). The yaw angles can be constant, ψ_0 , or variable over time, $\psi_c(t)$. This method is similar to a backstepping approach (Das, 2009).

In this case, the position and orientation are fixed, which implies that all velocities and accelerations are zero as well as the roll and pitch angles. The input u_2 must be close to zero, as well the rates p, q and r. The yaw angles can be non-zero if they are fixed, $\psi_c(t) = \psi_0$ (Mellinger, 2012). It is assumed that u_1 is very close to the weight.

With these assumptions, it is now possible to define expressions to determine the accelerations over the three axes from Equation (2):

$$\ddot{r}_1 = g(\Delta\theta \cos \psi_0 + \Delta\phi \sin \psi_0); \quad (23)$$

$$\ddot{r}_2 = g(\Delta\theta \sin \psi_0 + \Delta\phi \cos \psi_0); \quad (24)$$

$$\ddot{r}_3 = \frac{1}{m}u_1 - g. \quad (25)$$

In relation to the products of inertia, once the axes are close the principal ones, it is possible to assume Equation (26):

$$I_{xx} \approx I_{yy}. \quad (26)$$

Since the RPAS is symmetric, it is possible to write the rates of change based on the linearization of equations (2) and (3):

$$\dot{p} = \frac{u_2}{I_{xx}} = \frac{L}{I_{xx}}(F_2 - F_4); \quad (27)$$

$$\dot{q} = \frac{u_3}{I_{yy}} = \frac{L}{I_{yy}}(F_3 - F_1); \quad (28)$$

$$\dot{r} = \frac{u_4}{I_{zz}} = \frac{\gamma}{I_{zz}}(F_1 - F_2 + F_3 - F_4). \quad (29)$$

The error related to the position is defined as the difference between the desired state and the actual state:

$$e_i = (r_{i,des} - r_i). \quad (30)$$

Furthermore, a PD controller will be incorporated. This contains a proportional gain which affects the responsiveness of the system, depending only on the desired position and the actual one. The derivative gain is an extrapolation of the future error based on the current one and produces an inverse response related to the ratio experienced by the system. Thus, equation (31) can be derived in order to assure that the error becomes zero:

$$(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_i) + k_{p,i}(r_{i,des} - r_i) = 0. \quad (31)$$

2.4. Hover state

In this state, the RPAS needs to maintain a defined position, which means that the acceleration and velocity are zero:

$$\ddot{r}_{i,c} = \dot{r}_{i,c} = 0. \quad (32)$$

Moreover, it must be established that this condition is guaranteed. Thus, applying Equation (31), leads to:

$$\ddot{r}_{i,des} + k_{d,i}\dot{r}_{i,des} + k_{p,i}(r_{i,des} - r_i) = 0. \quad (33)$$

Based on equations (23–25), and assuming that the pitch and roll angles must be maintained at a constant value in order to achieve this state, the equation can be simplified to (Mellinger, 2012):

$$\ddot{r}_{1,des} = g(\theta_{des} \cos \psi_T + \phi_{des} \sin \psi_T); \quad (34)$$

$$\ddot{r}_{2,des} = g(\theta_{des} \sin \psi_T + \phi_{des} \cos \psi_T); \quad (35)$$

$$\ddot{r}_{3,des} = \frac{1}{m}u_1 - g. \quad (36)$$

Moreover, with a PD controller, as indicated in the equation (31), the error shall exponentially reach zero and also satisfy a second order differential equation (33) (Kumar, 2016). In relation to this equation, there is a set of terms related to the specified trajectory, $\ddot{r}_{i,des}$, $\dot{r}_{i,des}$ and $r_{i,des}$, and a set of terms related with the actual trajectory \dot{r}_i and r_i . Thus, the need to calculate the com-

manded sum of the thrusts from the motors, which will define the force that is required on the RPAS in order to maintain its position:

$$u_1 = mg + m\ddot{r}_{3,des} = mg - m(k_d\dot{r}_3 + k_p(r_{3,des} - r_3)). \quad (37)$$

The commanded roll and pitch angles can be obtained once the commanded acceleration in the x and y directions is known, as shown in equations (38) and (39), respectively:

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,des} \sin \psi_{des} - \ddot{r}_{2,des} \cos \psi_{des}); \quad (38)$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,des} \cos \psi_{des} - \ddot{r}_{2,des} \sin \psi_{des}). \quad (39)$$

The desired roll and pitch angles are defined in Equations (40) and (41), respectively:

$$\phi_{des} = \frac{1}{g}(\ddot{r}_{1,c} \sin \psi_{des} - \ddot{r}_{2,c} \cos \psi_{des}); \quad (40)$$

$$\theta_{des} = \frac{1}{g}(\ddot{r}_{1,c} \cos \psi_{des} - \ddot{r}_{2,c} \sin \psi_{des}). \quad (41)$$

Furthermore, the pitch and roll velocities are considered to be zero, since these angles are constant throughout the hover:

$$p_{des} = 0; \quad (42)$$

$$q_{des} = 0; \quad (43)$$

$$r_{des} = \dot{\psi}_T(t). \quad (44)$$

For the feedback loop, it is assumed that the commanded roll, pitch and yaw angles are known, as well their derivatives. Thus, all that it is required for the attitude feedback control is the actual values of the roll, pitch and yaw angles plus their derivatives to determine u_2 :

$$u_2 = \begin{bmatrix} k_{p,\phi}(\phi_{des} - \phi) + k_{d,\phi}(p_{des} - p) \\ k_{p,\theta}(\theta_{des} - \theta) + k_{d,\theta}(q_{des} - q) \\ k_{p,\psi}(\psi_{des} - \psi) + k_{d,\psi}(r_{des} - r) \end{bmatrix}. \quad (45)$$

This model was designed to provide a faster feedback by the attitude controller than the position controller loop, as seen in Figure 3.

2.5. 3-D Trajectory

Since the nominal state has already been defined, now a controller that will enable to follow a three-dimensional trajectory with the acceleration close to near-hover state is going to be presented. This model follows the same assumptions made above with the difference that $\ddot{r}_{i,c}$ and \dot{r}_i are no longer zero (Mellinger, 2012).

It is considered that the near-hover state assumptions of linear equations hold. To generate the desired acceleration, Equation (31) is used, so the error goes to zero exponentially.

From Equation (23), it is possible to determine the commanded state over the longitudinal and transversal directions, respectively:

$$\ddot{r}_{1,c} = \ddot{r}_{1,des} + k_{d,1}(\dot{r}_{1,des} - \dot{r}_1) + k_{p,1}(r_{1,des} - r_1); \quad (46)$$

$$\ddot{r}_{2,c} = \ddot{r}_{2,des} + k_{d,2}(\dot{r}_{2,des} - \dot{r}_2) + k_{p,2}(r_{2,des} - r_2). \quad (47)$$

However, since this model has some errors and limitations related to the input thrusts, the commanded trajectory may present some oscillations when sharp turns are encountered.

With this restriction, the need to provide a modification to describe the trajectories better arises. Defining \hat{t} as the trajectory unit's tangent vector and \hat{n} as the normal unit's vector, the binormal vector, \hat{b} , is obtained by a cross product of the trajectory unit's tangent vector and the normal unit's vector, $\hat{b} = \hat{t} \times \hat{n}$. With this, it is possible to define position and velocity errors (Mellinger, 2012):

$$e_p = ((r_{i,des} - r_i) \cdot \hat{n})\hat{n} + ((r_{i,des} - r_i) \cdot \hat{b})\hat{b}; \quad (48)$$

$$e_v = \dot{r}_{i,des} - \dot{r}_i. \quad (49)$$

Only the position error over the normal plane to the curve's closest point is considered. The position error in tangential direction is ignored. With the expression defined in Equation (31), it is possible to calculate the commanded acceleration, $\ddot{r}_{i,des}$:

$$(\ddot{r}_{i,des} - \ddot{r}_i) + k_d e_v + k_p e_p = 0. \quad (50)$$

All other assumptions made for the hover state hold; with the exceptions clarified in this section, it becomes possible to generate a trajectory in 3D (Mellinger, 2012).

4. Trajectory generation

To generate the trajectory, some equations will be presented and, depending on the precision required, their order will increase. Below an equation for the minimum snap trajectory that requires 7th order polynomials is presented. Each polynomial piece travels between two waypoints, taking a stipulated amount of time, T_i .

As p_i , the polynomial between positions ω_i and ω_{i+1} , takes T_i , time to complete. The total amount of time is represented by Equation (51):

$$S_i = \sum_{k=0}^{i-1} T_k. \quad (51)$$

The polynomial that represents the restrictions through the waypoints is represented in Equation (52) (Kumar, 2016):

$$p_i(t) = \alpha_{i0} + \alpha_{i1} \frac{t - S_i}{T_i} + \alpha_{i2} \left(\frac{t - S_i}{T_i} \right)^2 + \dots + \alpha_{i7} \left(\frac{t - S_i}{T_i} \right)^7. \quad (52)$$

To complete this equation, all the coefficients α_{ij} have to be determined. For this end, it is necessary to establish constraints. Firstly, it must go through all the waypoints (Kumar, 2016):

$$p_i(0) = \omega_{00} = \alpha_{00}; \quad (53)$$

$$p_i(S_i) = \omega_{i+1} \text{ for all } i = 0 \dots n. \quad (54)$$

Secondly, it must be defined that the quadcopter will start and stop at rest (Kumar, 2016):

$$p_n^{(k)}(0) = 0 \quad p_n^{(k)}(S_n) = 0 \text{ for all } 0 \leq k \leq 3; \quad (55)$$

$$p_n^{(k)}(S_n) = 0 \text{ for all } 0 \leq k \leq 3. \quad (56)$$

It must also be defined that the four derivatives are continuous between the waypoints (Kumar, 2016):

$$p_{i-1}^{(k)}(S_i) = p_i^{(k)}(S_i) \text{ for all } i = 1 \dots n \text{ and } k = 1 \dots 7. \quad (57)$$

Over these assumptions, it is possible to convert the constraints and unknown coefficients α_{ij} , into an equation matrix:

$$A\alpha = b. \quad (58)$$

Here α contains all the unknown coefficients, and A and b represent the constraints (Kumar, 2016).

4.1. Gain tuning – Ziegler–Nichols method

There are twelve gains to tune – half for the position, the other half for the attitude. A trial and error method approach becomes inefficient. Thus, the need to define a more systematic approach to gain tuning, the Ziegler-Nichols method, is required (Kumar, 2016; Ziegler, 1942).

The Ziegler-Nichols method is used to tune proportional, proportional integrative, proportional derivative and proportional integrative and derivative controllers. In this case, proportional derivative control will be applied.

This method has a set of steps that should be followed.

1. Set the proportional and derivative gains to zero.
2. Increase the proportional gain until the system outputs sustained oscillations that will be denominated as ultimate gain, K_u .
3. Measure the period, T_u , described by the oscillations.
4. Define the proportional gain as $K_p = 0,8 \times K_u$.
5. Define the derivative gain as $K_v = T_u / 8$.

Sometimes, after this procedure the controller's response is too slow. To improve it, it is advised to scale the proportional and derivative gains by small increments until a good response is obtained (Kumar, 2016; Ziegler, 1942).

5. Wind turbine blade modeling

A trajectory based on the dimensions of a wind turbine model GE 1.5sle is generated (Rosenbloom, 2014) and shown in Figure 5. The vehicle used is a Phantom 4 which has the specifications provided in Table 1. These parameters were included in the code in order to provide a better approximation for the problem.

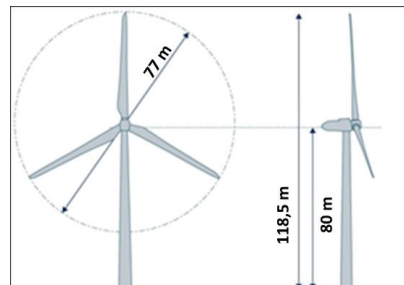


Figure 5. Model GE 1.5sle of a wind turbine that was used

Table 1. Parameters of the RPAS Phantom 4 (DJI, 2016)

Specifications of Phantom 4	
MTOW RPAS	1380 g
Propellers	240 x 127 mm
Diagonal Size (Propellers excluded)	350 mm
Maximum ascendant speed	6 m/s
Maximum descendant speed	4 m/s
Maximum speed	20 m/s
Maximum Service Ceiling Above Sea Level	6000 m
Maximum Wind Speed Resistance	10 m/s
Maximum flight time	28 min
Photo quality	12 MP
Satellite Positioning Systems	GPS/GLONASS
Video quality	4k at 120 fps (frames per second)

6. Results

The results that were obtained for solving the problem statement presented in the previous section, are shown below: the gains were tuned in order to improve the state estimation for the best. To obtain precise images together with an accurate state of the whole structure, the two trajectories were idealized.

6.1. Complete inspection from an arbitrary point

In the first scenario, the trajectory was executed through a first order polynomial based on a series of defined waypoints. It incorporates the root and the tip of each blade, enabling the camera to capture it and have a better general idea of its state. The proposed flight is designed for a velocity of 0.75 m/s which would start and rest at the origin of the referential. It would take around 800 seconds to complete this path. It was considered that the wind turbine is located at 10 meters from the position of the RPAS.

The defined waypoints were obtained by performing specific measurements based on trigonometric rules:

$$Waypoints = \begin{bmatrix} 0 & 0 & 10 & 10 & 10 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 33,34 & 0 & -33,34 & 0 & 0 & 0 \\ 0 & 30 & 80 & 118,5 & 80 & 60,75 & 80 & 60,75 & 80 & 30 & 0 \end{bmatrix} \tag{59}$$

In Equation (59), the first row relates to the X direction, the second row relates to the Y direction, and the third row relates to the Z direction.

In Figure 6, it is possible to see the desired trajectory, represented by the blue color, and the actual trajectory, represented by the red color. They describe the RPAS' path. A small deviation from the desired state can be seen in Figure 7. Nevertheless, it is possible to conclude that these are good results since the trajectory incorporated the whole blade from the tip to the root at the desired velocity.

In Figure 7, it is possible to see the deviation from the desired trajectory (blue line) and from the actual trajectory (red line). The first row represents the trajectory over the x axis, the second row represents the trajectory over the y axis, and the third one represents the trajectory over the z axis. Since the type of polynomial equation used to describe this trajectory was of the first order, each waypoint on the graph is represented as a corner.

In Figure 8, it is possible to see the velocity oscillation produced in order to move from one defined waypoint to another.

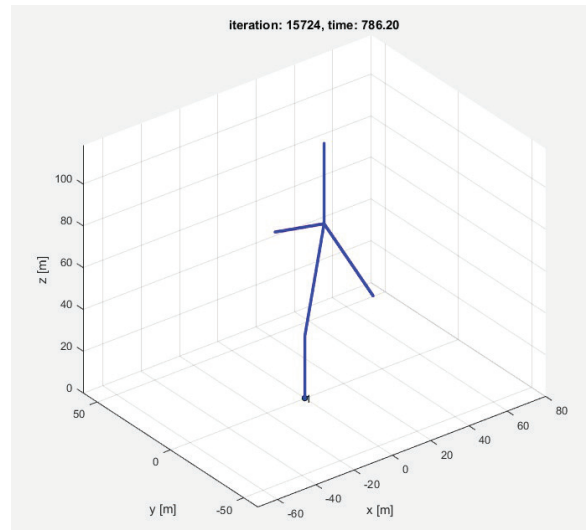


Figure 6. Trajectory described by the quadcopter in the first scenario

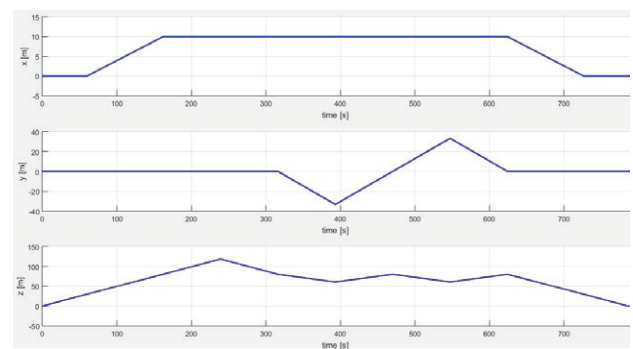


Figure 7. Position oscillations with respect to time

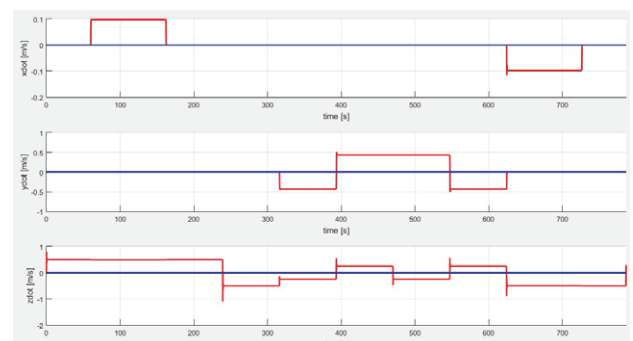


Figure 8. Velocity oscillations with respect to time

6.2. Helicoidal trajectory

In this second scenario, a trajectory based on the following equations to describe the movements over the longitudinal, transversal and vertical axis, respectively, was simulated:

$$x = \cos(2\pi t) \times r ; \quad (60)$$

$$y = \sin(2\pi t) \times r ; \quad (61)$$

$$z = (h_{final} - h_{initial}) \times t / T . \quad (62)$$

In these three equations, r represents the circumference's radius described in the trajectory. The initial height was defined as 60m; however, once the measurements related to Figure 5 were made, the minimum height on which the tip of the two lower blades are displaced was considered, and the final height was defined as 118.5 meters. Here t , represents the current time of the simulation, and, T – the total time of the simulation. This path was idealized to execute three laps until the RPAS would reach the maximum height over the wind turbine, defined as 118.5 m.

In Figure 9, it is possible to see the desired trajectory (represented by the blue color) and the actual trajectory (represented by the red color). This includes the trajectory described by the quadcopter. Since there is no clear distinction between the desired and actual trajectory, it is possible to conclude that these are accurate results for the desired values.

In Figure 10, it the actual trajectory is indicated by the color red and the desired trajectory – by blue. The first row represents the trajectory over the x axis, the second row represents the trajectory over the y axis, and the third one represents the trajectory over the z axis. In this case, no oscillations are seen. Thus, it is possible to say that the gains are properly set in order to provide an accurate trajectory.

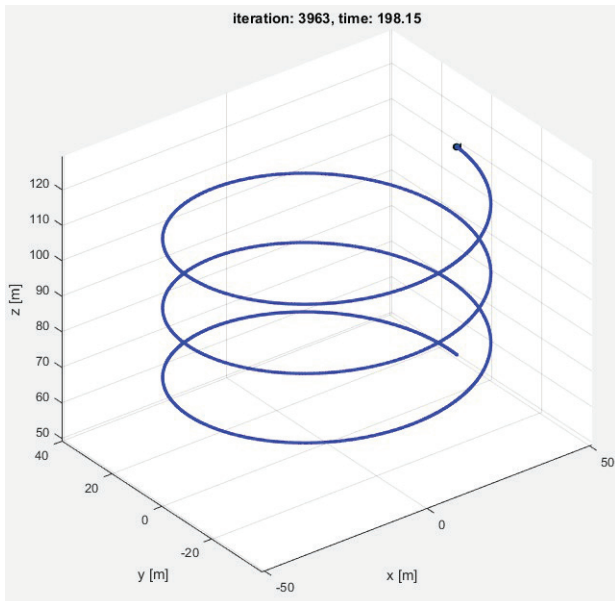


Figure 9. Trajectory defined by the RPAS in the second scenario

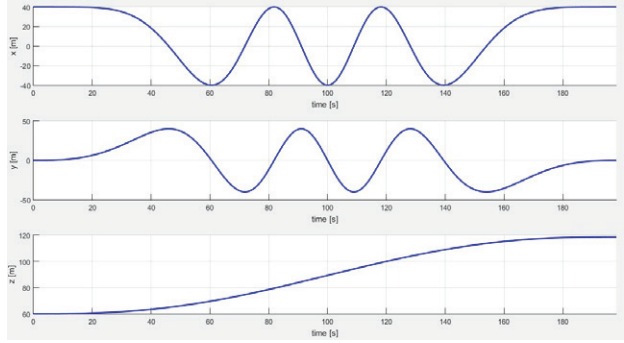


Figure 10. Position over the three axes defined by the RPAS

In Figure 11 the red and blue color represent the actual and desired values, respectively, regarding the velocities described by the RPAS. The first row represents the velocity over the x axis, the second row represents the velocity over the y axis and the third one represents the velocity over the z axis. On this case, no oscillations are shown. Thus, it is possible to say that the gains are properly set in order to provide an accurate trajectory.

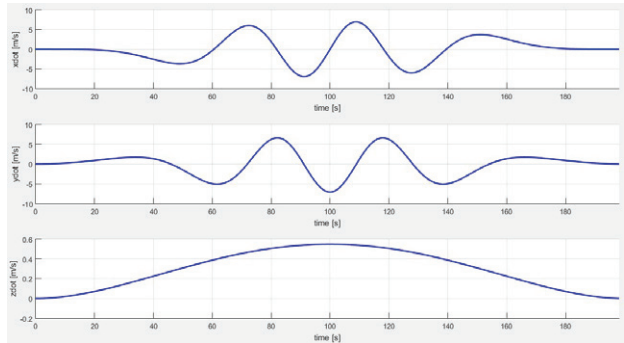


Figure 11. Velocity over the three axes defined by the RPAS

Conclusions

A model to define the trajectory over a wind turbine regarding its maintenance using an RPAS was proposed in this paper.

Of the two scenarios presented in the results section, it is possible to see that the Helicoidal trajectory provides a greater precision. This is due to the fact that the trajectory is described over continuous equations that allow maintaining smooth trajectory oscillations. The other scenario that was presented provides a more accurate inspection of the wind turbine; however, by describing the trajectory with a first order polynomial over a series of waypoints, it is harder to damp the oscillations.

It is possible to conclude that MATLAB can provide advantageous tools in order to generate a simulation of a precise trajectory. Although it is also possible to see that there are various limitations in terms of a direct application of the software, since, as mentioned before, composite materials are difficult surfaces to obtain accurate data on.

In future work, restrictions should be imposed on the velocity between waypoints before and after reaching the path over the wind turbine blades. The path approaching the blades can be executed with a higher velocity. While, over the blades, it should be slower in order to capture pictures with better quality, decreasing the error associated with the inspection. The trajectory should also be more complex to include both sides of the blade, the leading edge and the trailing edge, instead of a straight line between the tip and the root. The trajectory should also be adapted for a higher order polynomial in order to achieve greater stability when changing the direction. The rates of linear velocity must be upgraded.

Even though a spline defined by a 7th order polynomial was generated, the polynomial used to describe these waypoints was from the first order, so, instead of a smooth trajectory, it is preferred to maintain an accurate path that allows the quadcopter to pass near the middle of the blades, enabling the user to obtain more detailed information.

Upgrading this algorithm to a PID controller would make the system more accurate by stabilizing the magnitude and time error, which is possible considering the previous oscillations.

Even though this is a very limited method, it is well known that currently used inspection solutions are either outdated, expensive or inadequate. Therefore, embracing automation can bring the experience of qualified workers and technology together enhancing the renewable energies over fossil fuels.

References

- Das, A. (2009). Backstepping approach for controlling a quadrotor using lagrange form dynamics. *Journal of Intelligent and Robotic Systems*, 56(1-2), 127-151. <https://doi.org/10.1007/s10846-009-9331-0>
- DJI. (2016). *Phantom 4 Specs*. Retrieved from <https://www.dji.com/phantom-4/info>
- Esu, O. (2016). Feasibility of a fully autonomous wireless monitoring system for a wind turbine blade. *Renewable Energy*, 97, 89-96. <https://doi.org/10.1016/j.renene.2016.05.021>
- Fried, L. (2017). *Global Wind Energy Council*. Retrieved from http://www.gwec.net/wp-content/uploads/vip/GWEC_PRstats2016_EN_WEB.pdf
- Gupte, S. (2012). A survey of Quadrotor unmanned aerial vehicles. *Proceedings of IEEE* (pp. 1-6). Orlando: Southeastcon. <https://doi.org/10.1109/SECon.2012.6196930>
- Hehn, M. (2011). Quadcopter trajectory generation and control. *IFAC World Congress*, 1485-1491. <https://doi.org/10.3182/20110828-6-IT-1002.03178>
- IRENA. (2012). Wind power. *Renewable Energy Technologies: Cost Analysis Series*, 1(5), 4-34.
- Kovacs, M. (2016). Research on the potentiality of using aerial vehicles for monitoring the environment agent - air. *Environmental Legislation, Safety Engineering and Disaster Management*. Cluj-Napoca.
- Kumar, V. (2016). *Robotics: aerial robotics - lecture notes*. Retrieved from <https://www.coursera.org/learn/robotics-flight>
- Mellinger, D. W. (2012). *Trajectory generation and control for quadrotors*. Pennsylvania: University of Pennsylvania.
- Morgenthal, G., & Hallermann, N. (2016). Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures. *Advances in Structural Engineering*, 17(3), 289-302. <https://doi.org/10.1260/1369-4332.17.3.289>
- Motion, S. I. (2017). *Operations and Maintenance Services*. Retrieved from https://d3icht40s6fxmd.cloudfront.net/sites/default/files/inspeccion-aerea-profunda-de-palas-informes-y-mapeos-de-alta-resolucion-en-espectro-termico-y-spares_danos_en_punta.jpg
- Nurm, M. (2017). *Wind turbine operation and maintenance services*. Retrieved from <http://skyproff.com/wind-turbine-operation-and-maintenance-services/>
- Rosenbloom, E. (2014). *Size specifications of common industrial wind turbines*. Retrieved from <http://www.aweo.org/wind-models.html>
- Saavedra, R. C. (2015). Noise and vibration issues of wind turbines and their impact - a review. *Wind Engineering*, 39(6), 693-702. <https://doi.org/10.1260/0309-524X.39.6.693>
- Staffell, I. (2014). How does wind farm performance decline with age? *Renewable Energy*, 66, 775-786. <https://doi.org/10.1016/j.renene.2013.10.041>
- Ziegler, J. G. (1942). Optimum settings for automatic controllers. *ASME*, 64, 759-768.