Identification and Validation of a Quadrotor’s Model Dynamics

Rui Miguel Martins de Oliveira
rui.oliveira@tectno.ulisboa.pt
Instituto Superior Técnico
October 2014

Abstract—The dynamic simulation of Unmanned Aerial Vehicles (UAVs) is an essential tool for the development of autonomous behaviors for these platforms. The goal of this thesis is to identify a Quadrotor’s dynamics using a black-box approach, allowing the development of a simulation platform. It is aimed to establish not only the physical model of the quadcopter, but also to identify the stability controller and other internal processes.

The first part of this thesis presents a quadrotor’s model, by defining its kinematics and its dynamics, as well as the development of control methods.

The second part, describes the identification methods used, firstly the Process Models, and secondly the Gaussian Processes.

In the last part of this thesis, the results of the proposed identification methods are presented, firstly in a simulation environment based on the quadrotor’s mathematical model, and afterwards applied to the real system. Finally, after analyzing the results, the simulation platform is implemented.

Index Terms—Unmanned Aerial Vehicles, Quadrotor, Black-Box Identification, Process Models, Gaussian Processes.

I. INTRODUCTION

In the last few years, technological innovation has been growing significantly, namely in robotics, due to its great diversity and applicability in different areas. A particular field of robotics, is the aerial robotics, especially Unmanned Aerial Vehicles (UAVs). For this growth contributed, for example, size optimization of sensors and actuators, essential factor for increasing the flexibility and usability of such vehicles in many applications. UAVs have an extremely varying application domains that include both indoor and outdoor applications.

Some of those applications include aerial surveillance, aerial photography and video, monitoring and traffic pattern analysis, maintenance, search and rescue ([1]).

In order to correctly control this kind of vehicles, a very good model of the system is required, and identification is a crucial step for the development of these platforms. Several methods have been proposed to identify the UAVs’ models, for instance in [2], a Neural Network was used, where input-output data was provided from a nonlinear simulation of a X-Cell 60 helicopter. In [3], a time domain system identification method was applied in extracting a linear time-invariant system model, where the acquired model was used to design a feedback controller.

Gaussian Processes (GP) is a probabilistic framework approach, which have been applied to the problem of system identification from training data. In [4], it was used to identify and control a blimp. Here a GP model was used in conjunction with a non-linear model to learn the errors. The experiments were performed in a lab environment and it was not affected by wind disturbances.

Currently, there is a motion capture system, Vicon 1, which is based on multiple cameras with very high frame rates, capable of detecting short and sharp movements. There are a variety of motivations to use this type of system [5], for example, to acquire high precision position and posture data of the vehicle, necessary for system identification and validation. This system also allows external control of the vehicle through the data obtained. In [6], it is used to perform acrobatic maneuvers.

II. QUADROTOR MODEL AND CONTROL

A. Definitions and Basis Concepts

To describe the movement and attitude of the quadrotor, it’s necessary to define two coordinate systems [7]. The Inertial Frame \{I\} is fixed and centered in \(O^I\) with \(x^I\) pointing towards North, \(y^I\) pointing towards West and \(z^I\) pointing upwards as shown in Figure 1(a). The second frame, \{B\}, is a moving referential fixed on the quadrotor and centered in \(O^B\), the center of mass of the vehicle, with \(x^B\) pointing towards quadrotor’s front, \(y^B\) pointing towards quadrotor’s right and \(z^B\) pointing upwards as shown in Figure 1(b).

Fig. 1: Coordinate Systems [8].

In the inertial frame it is defined the position of the quadrotor, denoted \(\Gamma^I = [X,Y,Z]^T\), that corresponds to the displacement from \(O^I\) to \(O^B\), and its attitude \(\Theta^I = [\phi, \theta, \psi]^T\), that defines the rotation of the body frame relatively to the inertial frame. In the body frame is defined the

1Vicon motion capture: http://www.vicon.com/ [Consulted in 2014].

Quadrotors are aircraft that are equipped with four identical rotors in a cross configuration. As shown on Figure 1(a), the blades are paired and each pair rotates in a different direction. Motors (1) and (3) have a counter-clockwise rotation, and motors (2) and (4) have clock-wise rotation. Although the quadrotors have six degrees of freedom (DOF), they are only equipped with four propellers, so it is not possible to control all the 6 DOF, but only 4. However, thanks to its structure, it is trivial to chose the variables to control. These four variables correspond to the four basic movements of the vehicle, which allow to reach a desired altitude and attitude.

To obtain these movements, the angular speed of the motors $\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$ are adjusted (Figure 2). It follows the description of these movements:

- **Thrust** - $U_1[N]$:
  The vertical movement is obtained by increasing (or decreasing) all the propeller speeds by the same amount as shown on Figure 2(a) e (b).

- **Roll** - $U_3[N.m]$:
  The roll movement corresponds to a rotation of the quadrotor about the $x^B$ axis, it is obtained by increasing (or decreasing) $\Omega_2$ and by decreasing (or increasing) $\Omega_4$. This movement is shown on Figure 2(c) e (d).

- **Pitch** - $U_3[N.m]$:
  The pitch movement corresponds to a rotation of the quadrotor about the $y^B$ axis, it is obtained by increasing (or decreasing) $\Omega_1$ and by decreasing (or increasing) $\Omega_3$. This movement is shown on Figure 2(e) e (f).

- **Yaw** - $U_4[N.m]$:
  The yaw movement corresponds to a rotation of the quadrotor about the $z^B$ axis, it is obtained by increasing (or decreasing) the pair $\Omega_1 - \Omega_3$ and by decreasing (or increasing) the pair $\Omega_2 - \Omega_4$. This movement is shown on Figure 2(g) e (h).

**B. Kinematics and Dynamics**

The Kinematics of the quadrotor relate state variables on different coordinate frames. Equation 1 relates de position vector $\Gamma^I$ with the linear velocity vector $V^B$, and equation 2 relates the attitude vector $\Theta^I$ with the angular velocity vector $\omega^B$.

\[
\dot{\Gamma}^I = R_\theta V^B \tag{1}
\]

\[
\dot{\Theta}^I = T_\theta \omega^B \tag{2}
\]

In these equations, $R_\theta$ is the rotation matrix and $T_\theta$ is the transformation matrix, defined as:

\[
R_\theta = \begin{bmatrix}
c_\phi c_\theta & c_\phi s_\theta s_\phi - s_\phi c_\phi & s_\phi s_\theta s_\phi + c_\phi c_\phi \\
s_\phi c_\theta & c_\phi s_\theta s_\phi + s_\phi c_\phi & s_\phi s_\theta c_\phi - c_\phi s_\phi \\
-s_\theta & c_\theta s_\phi & c_\phi c_\theta
\end{bmatrix}
\]

\[
T_\theta = \begin{bmatrix}
1 & s_\theta t_\phi & c_\phi t_\phi \\
0 & c_\phi & -s_\phi \\
0 & s_\phi/c_\phi & c_\phi/c_\phi
\end{bmatrix}
\]

In the previous equation the following notation has been adopted: $c_k \equiv \cos(k)$, $s_k \equiv \sin(k)$ e $t_k \equiv \tan(k)$.

The dynamics of the quadrotor can be described in equations (5-6), using Newton’s second law:

\[
F^B = m \dot{V}^B + \omega^B \times (mV^B) \tag{5}
\]

\[
\tau^B = I \dot{\omega}^B + \omega^B \times (I \omega^B) \tag{6}
\]

where $m[Kg]$ is the mass of the quadrotor and $I[N.m.s^2]$ is the inertia matrix, which is considered diagonal due to the symmetry of the vehicle.

The previous equations relate the forces and moments applied with state variables of the vehicle. The relation between these forces and moments with the angular velocity of the motors are shown in equations (7-9).

\[
F^B = \begin{bmatrix}
F_x \\
F_y \\
F_z
\end{bmatrix} = \begin{bmatrix}
R_\theta^{-1} & 0 \\
0 & -mg
\end{bmatrix} \begin{bmatrix}
U_1
\end{bmatrix} \tag{7}
\]

\[
\tau^B = \begin{bmatrix}
\tau_x \\
\tau_y \\
\tau_z
\end{bmatrix} = \begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix} \tag{8}
\]

\[
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} = \begin{bmatrix}
b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
b(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\
b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\
b(\Omega_2^2 + \Omega_3^2 - \Omega_1^2 - \Omega_4^2)
\end{bmatrix} \tag{9}
\]

where $g[N.s^2]$ is the gravitational acceleration, $b[N.s^2]$ is the thrust coefficient, $d[N.m.s^2]$ is the drag coefficient and $l[m]$ the distance between the center of the propeller and the center of the quadrotor.
C. Control

In this section are presented two controllers, that will be implemented in the simulator which is based on the theoretical model of the quadrotor, and which will allow a first phase of tests of the proposed identification methods.

In this thesis it is used a modified Proportional-Integral-Derivative (PID) controller, because the traditional PID structure presents a drawback. A sharp movement can saturate the actuators and push away the system from a linear zone [10]. For this reason, the PID architecture shown in Figure 3, presents the derivative action of the system output. This modified PID structure can be written according to the equations:

\[ u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau - K_d \frac{d}{dt} y(t) \]  
\[ u(s) = (K_p + \frac{K_i}{s}) e(s) - sK_d y(s) \]

1) Stability Control: Figure 4 shows the architecture of the stability control.

It is now possible to write the three controllers in the Laplace domain:

\[ U_2 = \left[ \left( K_{p\phi} + \frac{K_{i\phi}}{s} \right) e_\phi(s) - sK_{d\phi} \phi \right] I_{xx} \]
\[ U_3 = \left[ \left( K_{p\theta} + \frac{K_{i\theta}}{s} \right) e_\theta(s) - sK_{d\theta} \theta \right] I_{yy} \]  
\[ U_4 = \left[ \left( K_{p\psi} + \frac{K_{i\psi}}{s} \right) e_\psi(s) - sK_{d\psi} \psi \right] I_{zz} \]

2) Position Control: Figure 5 shows the architecture of the position control.

It can be seen in the figure that there are two controllers: Inner Loop Controller and Outer Loop Controller. The first one is constituted by the stability and altitude control \((Z)\). The second one controls de horizontal position \((X, Y)\). In the Outer Loop Controller the pitch and roll angles are used to control position in \(x^l\) and \(y^l\). Assuming hovering conditions as before, and according to [11] and [12], by linearizing equation 1 one obtains the equations used in control:

\[
\begin{align*}
\dot{\phi} &= \theta_r c_\psi + \phi_r s_\psi \\
\dot{\theta} &= \theta_r s_\psi - \phi_r c_\psi \\
\dot{\psi} &= -g + (c_\phi c_\theta) U_3 \\
\end{align*}
\]

and the last Inner Loop Controller:

\[
U_1 = \left[ \left( K_{p\phi} + \frac{K_{i\phi}}{s} \right) e_z(s) - sK_{d\phi} Z \right] + g \frac{m}{c_\phi c_\theta}
\]
III. QUADROTOR IDENTIFICATION

A. Black-Box Identification

When a process is too complex to be described by some idealized physical laws or there is not enough information available, it is usual to resort to the black-box identification. The black-box identification dispenses knowledge of physical or chemical laws and is based only on the relationship between the input and the system output.

In this work it was used a black-box identification, because one of the goals is to make the identification of a quadrotor, where some internal processes of the vehicle are unknown, for example, the PID structure used in the stability control.

The typical design cycle during a system identification (Black Box Identification) involves two tasks: the identification of a model and its validation[13](Figure 6). Hence, two data sets are necessary, one to identify and one to validate.

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}
\]  

(17)

The method that presents a lower RMSE value may be considered more effective. In the case of obtaining similar values, it is necessary to resort to other types of analysis, such as the computational load of each method.

To apply the black-box identification to the quadrotor, it is essential to identify the system inputs. Since the vehicle is maneuvered by remote control, the system inputs are the commands sent. In the case of the quadrotor, used in the thesis, the signals sent by the remote control are the desired roll angle \( \phi_r \), the desired pitch angle \( \theta_r \), the desired yaw angle rate \( \psi_r \), and the thrust \( U_1 \). Therefore, the system can be decoupled into four subsystems which will be identified separately [14](Figure 7).

B. Process Models

Process models (PM) are popular for describing system dynamics in many industries and apply to various production environments. The advantages of these models is their simplicity. The structure of a process model is a simple continuous-time transfer function that describes linear system dynamics in terms of one or more of the following elements:

- Static gain \( K_p \) - characterize the sensitivity of the output to the input signal.
- Time delay \( T_d \) - Time before the system output responds to the input(dead time).
- Time constant \( T_p \) - Characterize the speed of response of a system. It is a measure of the time necessary for a process to adjust to a change in the input.

In this thesis it was used the System Identification Toolbox\textsuperscript{TM}[15] of the Matlab\textsuperscript{®} software, to run the Process Models, which allows the user to create different types of transfer functions to approximate the system dynamics. Figure 8, shows the graphical environment of the toolbox.

As mentioned before, the goal is to identify the vehicle by decoupling its dynamics into four subsystems, which will be identified separately. Roll, pitch and yaw angle rate dynamics are approximated with a 1\textsuperscript{st} order system and a
delay element, while the thrust dynamics is approximated with a static gain. Figure 9 shows a sketch of the architecture of the simulator, where the four dynamics are associated to equation 1 for calculating accelerations.

C. Gaussian Processes

Gaussian Processes (GP) is a powerful, non-parametric tool for regression. A GP is a Gaussian random function, fully characterized by its mean and covariance function. GP can be viewed as collection of random variables which have a joint multivariate Gaussian distribution: 

\[ f(x^1),..., f(x^n) \sim \mathcal{N}(0, \Sigma) \]

where \( \Sigma_{pq} \) gives the covariance between \( f(x^p) \) and \( f(x^q) \) and is a function of the corresponding \( x^p \) e \( x^q \):

\[ \Sigma_{pq} = C(x^p, x^q) \] (18)

The covariance function \( C(.,.) \) can be of any kind, providing it generates a positive definite covariance matrix. Assuming a stationary process [16], a common choice of covariance function is the squared exponential:

\[ C(x^p, x^q) = v_1 \exp \left( -\frac{1}{2} \sum_{d=1}^{D} w_d (x^p_d - x^q_d)^2 \right) \] (19)

Where \( D \) is the length of vector \( x \) and \( v_1, w_1, ..., w_D \) are free parameters, usually called hyperparameters. Typically, covariance functions are chosen so that points close together in the input space are more correlated than points far apart. The parameter \( v_1 \) controls the magnitude of the covariance and hyperparameters \( w_d \) represent the relative importance of each component \( x_d \) of the vector \( x \).

The training set \( T = \{(x^1, y^1), (x^2, y^2), ..., (x^N, y^N)\} \) is assumed to be drawn from the noisy process:

\[ y^i = f(x^i) + \epsilon \] (20)

where \( x^i \) is an input vector in \( \mathbb{R}^D \) and \( y^i \) is a scalar output in \( \mathbb{R} \). The noise term \( \epsilon \) is drawn from \( \mathcal{N}(0, v_0) \). Within this probabilistic framework, we have \( y^1, ..., y^N \sim \mathcal{N}(0, \Sigma) \), with \( \Sigma = \Sigma + v_0 I \), where \( I \) is the \( N \times N \) identity matrix.

Based on a set of \( N \) training data pairs, \( \{x^i, y^i\}_{i=1}^N \), the goal is to find the predictive distribution of \( y^* \) corresponding to a new given input \( x^* \). We can write:

\[ \begin{pmatrix} y \\ y^* \end{pmatrix} \sim \mathcal{N}(0, \mathbf{K}_{N+1}), \] (21)

with covariance matrix:

\[
\mathbf{K}_{N+1} = \begin{bmatrix} \mathbf{K} & \mathbf{k}(x^*) \\ \mathbf{k}(x^*)^T & [k(x^*)] \\ \end{bmatrix}
\] (22)

where \( y = [y^1, ..., y^N]^T \) is an \( N \times 1 \) vector of training targets, \( k(x^*) = [C(x^1, x^*), ..., C(x^N, x^*)]^T \) is the \( N \times 1 \) vector of covariances between training inputs and the test input and \( k(x^*) = C(x^*, x^*) \) is the autocovariance of the test input.

We can divide this joint probability into a marginal and a conditional part. The marginal part gives us the likelihood of the training data: \( y|X \sim \mathcal{N}(0, \mathbf{K}) \), where \( X \) is the \( N \times D \) matrix of training inputs.

The unknown hyperparameters of the covariance function, as well as the noise variance \( v_0 \), can be estimated by maximization of the log-likelihood [17]:

\[
\mathcal{L}(\Theta) = \log(p(y|X)) = -\frac{1}{2} \log(|\mathbf{K}|) - \frac{1}{2} y^T \mathbf{K}^{-1} y - \frac{N}{2} \log(2\pi),
\] (23)

where \( \Theta = [w_1, ..., w_D, v_0, v_1] \) is the vector of parameters and \( \mathbf{K} \) is the \( N \times N \) training covariance matrix. The optimization requires the computation of the derivatives of \( \mathcal{L} \) with respect to each of the parameters:

\[
\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta_i} = -\frac{1}{2} \text{trace} \left( \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \Theta_i} \right) + \frac{1}{2} y^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \Theta_i} \mathbf{K}^{-1} y
\] (24)

This involves the computation of the inverse of the covariance matrix \( \mathbf{K} \), which can become computationally demanding for large \( N \). In this thesis it was used the Gaussian Process Regression and Classification Toolbox ², which allows to choose the desired covariance matrix, and given a training data set it is possible to estimate the parameters \( \Theta \).

After optimizing the covariance function hyperparameters, it is possible to obtain a prediction of the GP model for a new input \( x^* \). The conditional part of equation 21 gives the predictive distribution of \( y^* \):

\[
p(y^*|y, X, x^*) = \frac{p(y^*|X)}{p(y|X)}
\] (25)

it can be shown that this distribution is Gaussian with mean and variance[17]:

\[
\mu(x^*) = \mathbf{k}(x^*)^T \mathbf{K}^{-1} y
\] (26)

\[ \sigma^2(x^*) = k(x^*) - k(x^*)^T K^{-1} k(x^*) + \nu_0 \quad (27) \]

The equation 26 is now prediction for a new input, and equation 27 is the uncertainty of the prediction.

The goal is to use the GP to model dynamic systems. Given the model of equation 20, where:

\[ x(t) = \begin{bmatrix} y(t-1), y(t-2), \ldots, y(t-L), u(t-2), \ldots, u(t-L) \end{bmatrix}, \quad (28) \]

is the state vector at \( t \), composed of the previous outputs \( y \) and inputs \( u \), up to a given lag \( L \).

The task is to model the dynamic system using GP, in order to make multiple-step ahead prediction. One way to do multiple-step ahead predictions is to make one-step predictions iteratively by feeding back the predicted output (Figure 10).

In the case of the dynamics of the quadrotor, the GP model is used for modeling each one of the subsystems. It was used the value \( L = 2 \), because for higher values there weren’t relevant improvements. Figure 11, shows a sketch of the architecture of the simulator using GP.

IV. SIMULATION PLATFORM

In order to test the proposed identification methods, a simulator based on the mathematical model of the quadrotor was created. This simulator was implemented and modified from [8], using Simulink® tool from the Matlab® software. Figure 12 shows the architecture of the created simulator.

It was used the option for manual flight, using the joystick from the RCInput block. Four flights were made to identify the dynamics of each subsystem and a flight was made for testing and validation. The results obtained in simulation allowed to assess the methods of identification before being applied to the real vehicle, and both methods have shown great efficacy in a simulation environment.

V. EXPERIMENTAL PLATFORM

The vehicle in which it is intended to identify the system, is a quadrotor developed by UAVision® company. It was acquired the model UX-4001 Mini (Figure 13).

A. Motion Tracking

To enable the acquisition of the position and attitude of the quadrotor with high precision, it was used the Motion Tracking of the Biomechanics Laboratory of Lisbon®. This lab is equipped with 14 infra-red cameras Qualisys ProReflex MCU 500/1000. For the data acquisition, a frequency of 100 Hz and five markers were used. The disposition of the five markers is shown in Figure 14.

After data collection, it is necessary to estimate the attitude and position of the vehicle. Determining the relationship between two coordinate systems through the use of sets

4Motion tracking: http://flb.tecnico.ulisboa.pt/Resources [Consulted in 2014].
of corresponded features measurements is known as the absolute orientation problem. Given two corresponding point sets \( \{m_i\} \in \{d_i\}, i = 1, \ldots, N \), such that they are related by:

\[
d_i = Rm_i + T + V_i, \quad (29)
\]

where \( R \) is the 3 \times 3 Rotation matrix, \( T \) is the translation vector and \( V_i \) a noise vector. To obtain the optimal transformation \( [\hat{R}, \hat{T}] \), that maps the set \( \{m_i\} \) in \( \{d_i\} \), requires minimizing a least squares error criterion given by:

\[
\Sigma^2 = \sum_{i=1}^{N} \|d_i - Rm_i - T\|^2 \quad (30)
\]

To estimate the rotation matrix \( \hat{R} \), the point sets \( \{m_i\} \) and \( \{d_i\} \) should have the same centroid. By defining:

\[
\bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i, \quad d_{ci} = d_i - \bar{d}
\]

\[
\bar{m} = \frac{1}{N} \sum_{i=1}^{N} m_i, \quad m_{ci} = m_i - \bar{m}
\]

Equation 30 can be rewritten and reduced to:

\[
\Sigma^2 = \sum_{i=1}^{N} \|d_i - \hat{R}m_i\|^2 = \sum_{i=1}^{N} (d_i^T d_{ci} + m_{ci}^T m_{ci} - 2d_{ci}^T \hat{R}m_{ci}) \quad (32)
\]

This equation is minimized when the last term is maximized, which is equivalent to maximizing Trace(\( \hat{R}, H \)) [19], where \( H \) is the correlation matrix defined by:

\[
H = \sum_{i=1}^{N} m_{ci}d_{ci}^T \quad (33)
\]

If the singular value decomposition of \( H \) is given by \( H = U \Lambda V^T \), then the rotation matrix, \( \hat{R} \), that maximizes the desired trace is [19]:

\[
\hat{R} = VU^T \quad (34)
\]

and the translation is given by:

\[
\hat{T} = \bar{d} - \hat{R}\bar{m} \quad (35)
\]

The attitude angles can be obtained from estimated rotation matrix. Considering the generic matrix:

\[
R = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} \\
    r_{21} & r_{22} & r_{23} \\
    r_{31} & r_{32} & r_{33}
\end{bmatrix} \quad (36)
\]

The attitude angles are given by:

\[
\phi = \arctan2(r_{32}, r_{33}) \\
\theta = \arctan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\
\psi = \arctan2(r_{21}, r_{11}) \tag{37}
\]

### B. Results

The procedure used was similar to that used in the simulation. It was acquired two distinct data sets, one for system identification and the other to make predictions and validate the model. Figures 15-18, shows the error between the two models predictions and the real values.

#### TABLE I: Predictions Root Mean Square Error.

<table>
<thead>
<tr>
<th>Method</th>
<th>Roll ((\text{rad}))</th>
<th>Pitch ((\text{rad}))</th>
<th>Yaw rate ((\text{rad.s}^{-1}))</th>
<th>Thrust ((\text{N}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>(1.84 \times 10^{-3})</td>
<td>(1.95 \times 10^{-3})</td>
<td>(7.42 \times 10^{-4})</td>
<td>(2.62 \times 10^{-1})</td>
</tr>
<tr>
<td>PM</td>
<td>(1.72 \times 10^{-2})</td>
<td>(1.73 \times 10^{-2})</td>
<td>(7.42 \times 10^{-2})</td>
<td>(6.10 \times 10^{-1})</td>
</tr>
</tbody>
</table>

**Fig. 14:** Five Markers Disposition.

**Fig. 15:** Roll error GP vs PM.

**Fig. 16:** Pitch error GP vs PM.
It can be observed from the data that both methods do a good approximation of the acceleration in the $x$ and $y$ axis. In the case of the acceleration in the $z$, the Gaussian Process model shows a better result. This result matches the fact that the prediction of the thrust is better using this method. The thrust prediction is more prevalent in the calculation of the acceleration in $z$ axis than in the other axes.

C. Validation

To validate the simulator architecture presented in chapter III, it is not enough to analyse the dynamics predictions. To make the complete validation of the model, it is necessary to compare the real accelerations with accelerations calculated with the predictions. Figures 19-21, shows the error between the real accelerations and the two models predictions.

Again it will be used the root mean square error to compare the methods. Table II shows the root mean square errors for the accelerations for both methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>$x$ Acceleration</th>
<th>$y$ Acceleration</th>
<th>$z$ Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>$2.68 \times 10^{-4}$</td>
<td>$1.89 \times 10^{-4}$</td>
<td>$1.82 \times 10^{-4}$</td>
</tr>
<tr>
<td>PM</td>
<td>$2.13 \times 10^{-4}$</td>
<td>$1.94 \times 10^{-4}$</td>
<td>$3.96 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

**TABLE II: Accelerations Root Mean Square Errors.**

It can be observed from the data that for both methods, the predictions of roll, pitch and yaw dynamics approximates quite well the real quadrotor behaviour. Concerning the thrust prediction, the Gaussian process is more effective.

D. Final Simulator

The development of the final simulation platform, which simulates real quadrotor behaviour, was based on the obtained results. Since the Gaussian processes have a high computational burden, it was chosen to do a combination of the two methods. Thus it was used Process Models to make prediction of the roll, pitch and yaw rate dynamics and Gaussian Process to make thrust predictions.

In the implementation of the simulator, it was used again the Simulink® tool. Figure 22 presents the final architecture of the simulator. Where the block Dynamics has the structure presented in the chapter III.

The implemented simulator was tested by regular drivers of the actual vehicle, and the feedback was very positive, being the flight sensations of the simulator quite similar to the real quadrotor. The only negative aspect was that the simulator is controlled via a joystick and has a different sensitivity compared to the remote control of the vehicle.
VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

In this thesis, two methods have been proposed to perform identification of the quadrotor’s system dynamics, which allowed the development of a platform that simulates the real behaviour of the vehicle. This platform allows a first phase of tests on new developments before they are implemented in the real quadrotor.

To achieve the proposed objectives, it was necessary to develop the dynamics and kinematics of the vehicle, as well as control methods. Two types of controllers were implemented, the stability control and the position control.

Through the mathematical model presented, it was possible to create a simulator that allowed the test of the proposed identification methods, before being applied to the real vehicle. The architecture of the simulator is similar to the system to identify, receiving the same commands through a joystick, and equipped with a stability controller.

For the identification of the quadrotor’s system dynamics, it was used a black-box approach by applying two methods: Process Models and Gaussian Processes. The Process Models are a simpler approach, which approximates the dynamics of the vehicle in a linear manner, and the estimation of a set of parameters is necessary. In the case of Gaussian Processes, it is a probabilist approach which allows the identification of non-linearities of the system, and its main disadvantage is the high computational burden.

The results of the system’s identification were satisfactory for both methods, just noting an advantage of using Gaussian Processes for thrust predictions. For this reason, the implementation of the final simulator is a combination of the two methods used.

B. Future Work

As future goals, the final platform developed can be tested in closed loop. For this, it is necessary to introduce more processing power on the vehicle to be able to estimate the global position.

Another improvement that may be made is to replace the joystick from the simulator with the remote control of the quadrotor, allowing a comparison between flight sensations of simulator and the real system more accurately. This substitution also allows inexperienced people to have a first contact in maneuvering a quadrotor before moving to the actual vehicle.

REFERENCES