MASTER THESIS

Modeling and Identification of a Quadrotor using LPV techniques

Abel Torren Larroya

SUPERVISED BY

Raúl Benitez

Universitat Politècnica de Catalunya
Master in Aerospace Science & Technology

July 2014

This Page Intentionally Left Blank

Modelization and Identification of a Quadrotor

BY

Abel Torren Larroya

DIPLOMA THESIS FOR DEGREE

Master in Aerospace Science and Technology

AT

Universitat Politècnica de Catalunya

SUPERVISED BY:

Raul Benítez

Department of Automatic Control (ESAII)

This Page Intentionally Left Blank

ABSTRACT

The aim of this project is to modelize a LPV model and to compare different identification methods. In order to accomplish that, first of all a white box model of the quadrotor has been provided. This model has been used to find a LPV description that can be used in this case. With this data various identification procedures has been tested. This project is focused in the use of particle filters to identify the parameters of the system.

Table of Contents

INTRODUCTION 1

CHAPTER 1 LPV MODEL 3

1.1. Modeling a quadrotor 3
 1.1.1. Introduction to a Quadrotor 3
 1.1.2. Deriving the physical equations 4
1.2. LPV modeling 8
1.3. Quadrotor LPV model 9

CHAPTER 2 STANDARD IDENTIFICATION METHODS 12

2.1. Identification of the System State Space model 12
2.2. Ident 13
 2.2.1. Probing the method 13
 2.2.2. Using the model 13

CHAPTER 3 BAYESIAN IDENTIFICATION 14

3.1. Introduction to Bayesian Inference 14
 3.1.1. Bayesian Inference 14
3.2. Dynamic Linear Models 15
3.3. Models with unknown parameters 15
 3.3.1. Maximum Likelihood Estimator 15
 3.3.2. Parameter Learning 16

CHAPTER 4 PROBABILISTIC IDENTIFICATION METHODSERROR! BOOKMARK NOT DEFINED.

4.1. Maximum likelihood in R 22
 4.1.1. First Test 22
 4.1.2. Result with the quadrotor model 23
4.2. Liu and West 24
 4.2.1. Algorithm 24
 4.2.2. Results 25
4.3. Liu and West with changepoints 27
 4.3.1. Algorithm 27
 4.3.2. Results 28

CHAPTER 5 CONCLUSIONS AND FUTURE WORKS 32

5.1. Conclusions 32
5.2. Future works 32
5.3. Contributions 32

BIBLIOGRAPHY 33

List of Figures

Figure 1.1 Simplified quadrotor.. 3

Figure 2.1 Simulink Model 12

Figure 4.1 Relative error vs data length in a MLE identification for 1 parameter 23

Figure 4.2 Relative error vs data length in a MLE identification for 2 parameters..................24

Figure 4.3 Parameter estimation and relative error for parameter estimation.......................25

Figure 4.4 Gt and variances of the uncertainties identification using Liu and West 26

Figure 4.5 $Gt(t)$ and variances of the uncertainties identification using Liu and West...........26

Figure 4.6 Detail of the identification of the varying parameter 27

Figure 4.7 Identification and relative error of a parameter identified using changepoints 29

Figure 4.8 Identified parameter and real value in the case of a time varying parameter........29

Figure 4.9 Identification and relative error of a time varying parameter identified using changepoints (β=5%) 30

Figure 4.10 Identification and relative error of a time varying parameter identified using changepoints (β=10%) 30

Figure 4.11 Identification and relative error of a time varying parameter identified using changepoints (β=10% and lower variance) 31

Figure 4.12 Identification and relative error of a time varying parameter identified using changepoints (β=10% and uniform prior)..31

This Page Intentionally Left Blank

INTRODUCTION

The purpose of this project is to provide a comparison of various methods to identify a system. In this case this will be a LPV model of a quadrotor.

The first chapter contains an introduction to quadrotors and then there is a summary of the LPV theory that is necessary to understand the performances needed in order to identify it.

Then in the second chapter the algorithms of the procedure and the results using the standard identification methods found in the identification toolbox of Matlab.

In the third chapter the Bayesian inference theory is provided and different probabilistic identification methods are introduced.

In the fourth chapter the algorithms and results of the probabilistic methods are provided.

Chapter 1

LPV Model

In this chapter, the model of our system will be provided. To do so the physical equations will be used [ref. 1]. Then these relationships will be turned into a LPV model.

1.1. Modeling a quadrotor

1.1.1. Introduction to a Quadrotor

A quadrotor, or quadcopter, is a multirotor helicopter that is lifted and propelled by four rotors. The lift force is generated by a set of revolting airfoils.

The quadrotor will be modeled with its four rotors in cross configuration. The structure that connects them will be assumed to be rigid, so the only variable that can vary is the speed of the propellers. In this case, all the possible movements of the quadcopter will be directly related to the rotor velocity.

The front and rear rotors rotate counter-clockwise, while the left and right ones turn clockwise. Doing this there is no yaw rotation in hovering and the tail rotor, which is used in standard helicopters, isn't needed.

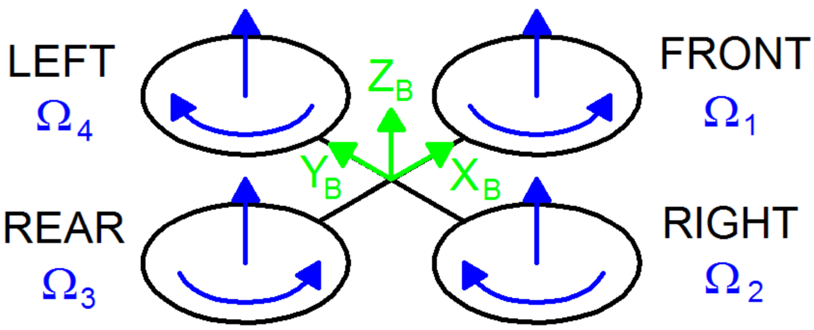

Figure 1.1 Simplified quadrotor

Despite the quadrotor 6 degrees of freedom, it's equipped with four propellers, so it can only reach the set point in four. These are related with the basic movements that allow the helicopter to reach a certain altitude and attitude.

1. Throttle
   This is provided by varying all the propeller rotations by the same amount. In the case that the quadrotor isn't in horizontal position, this will provide a horizontal and vertical acceleration in the inertial frame.

2. Roll
   This is provided by increasing the rotation of the right propeller while decreasing the rotation of the left one, or the opposite. It produces a torque with respect to the Xb axis.
3. Pitch
   This movement is similar to the roll, but in this case it is produced increasing the velocity of the front propeller and decreasing the velocity of the rear one. In this case, it produces a torque with respect to the Yb axis.
4. Yawn
   This is provided by increasing the velocity of the clockwise rotation propellers while decreasing the velocity of the counter-clockwise ones.

1.1.2. Deriving the physical equations

In order to derivate the physical model, two frames have been defined.

- The earth inertial frame
- The body-fixed frame

The kinematics of a generic 6 degree of freedom body, can be defined as

$$\dot{\xi} = J_{\Theta} \upsilon \qquad \textbf{(1.1)}$$

where $\dot{\xi}$ is the generalized velocity vector with respect to the earth inertial frame, υ is the one with respect to the body-fixed frame and J_{Θ} is the generalized matrix. Furthermore ξ is composed of the quadrotor linear and angular position with respect to the earth inertial frame.

$$\xi = [X\ Y\ Z\ \phi\ \theta\ \psi]^T \qquad \textbf{(1.2)}$$

Similarly υ is composed of the quadrotor linear and angular velocity with respect the body-fixed frame.

$$\upsilon = [u\ v\ w\ p\ q\ r]^T \qquad \textbf{(1.3)}$$

In addition, J_{Θ} is composed of 4 sub-matrices. Where 0_{3x3} is a 3 times 3 matrix filled with zeros, R_{Θ} is the rotation matrix and T_{Θ} is the transfer one.

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3x3} \\ 0_{3x3} & T_{\Theta} \end{bmatrix} \qquad \textbf{(1.4)}$$

The rotation and transfer matrices are defined as

$$R_\Theta = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi c_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad \textbf{(1.5)}$$

$$T_\Theta = \begin{bmatrix} 1 & -t_\theta s_\phi & t_\theta c_\phi \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \quad \textbf{(1.6)}$$

where $c_k = \cos k\,, s_k = \sin k\,, t_k = \tan k$. The dynamics of the generic 6 degree of freedom rigid-body takes into account the mass of the body, $m\ [kg]$, and its inertia matrix, $I\ [N\ m\ s^2]$ and it's described by

$$\begin{bmatrix} m\, I_{3x3} & 0_{3x3} \\ 0_{3x3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad \textbf{(1.7)}$$

where I_{3x3} is a 3 times 3 identity matrix, $\dot{V}^B$ is the linear acceleration vector $[m\, s^{-2}]$ with respect the body-fixed frame, $\dot{\omega}^B$ is the angular acceleration vector $[rad\ s^{-2}]$ with respect the same frame, F^B is the quadrotor forces vector $[N]$ and τ^B is the quadrotor torques vector $[N\ m]$ with respect the body-fixed frame.

Then it's assumed that the origin of the body-fixed frame is coincident with the center of mass and that the inertial matrix is diagonal.

A generalized force vector can be defined as

$$\Lambda = \left[F_x\ F_y\ F_z\ \tau_x\ \tau_y\ \tau_z\right]^T \quad \textbf{(1.8)}$$

Using this equation, the dynamics can be rewritten as

$$M_B \dot{\upsilon} + C_B(\upsilon)\upsilon = \Lambda \quad \textbf{(1.9)}$$

where $\dot{\upsilon}$ is the generalized acceleration vector with respect the body-fixed frame, M_B is the inertia matrix and $C_B(\upsilon)$ is the Coriolis-centripeta matrix, both with respect the body-fixed frame.

$$M_B = \begin{bmatrix} m\, I_{3x3} & 0_{3x3} \\ 0_{3x3} & I \end{bmatrix} \quad \textbf{(1.10)}$$

Thanks to the assumptions made before, M_B is a diagonal and constant matrix.

$$C_B(\upsilon) = \begin{bmatrix} 0_{3x3} & -mS(V^B) \\ 0_{3x3} & -S(I\omega^B) \end{bmatrix} \quad \textbf{(1.11)}$$

$S(K)$ is defined as a skew-symmetric operator, that given a three dimension vector is defined as follows.

$$S(K) = \begin{bmatrix} 0 & -K_3 & K_2 \\ K_3 & 0 & -K_1 \\ -K_2 & K_1 & 0 \end{bmatrix}, \qquad K = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} \tag{1.12}$$

In the case of a quadrotor, Λ can be divided in three different vectors according of which of the quadcopter contributions describes.

The first of all is the gravity vector, $G_B(\xi)$ given by the acceleration due to the gravity g $[m\ s^{-2}]$.

$$G_B(\xi) = mg\left[s_\theta \ -c_\theta s_\phi \ -c_\theta c_\phi \ 0\ 0\ 0\right]^T \tag{1.13}$$

The second contribution is due to gyroscopic effects due to the propeller rotation, since two rotates clockwise and the other two counterclockwise, there is an imbalance when the sum of the rotation is not zero. There is also a contribution of the pitch and the roll.

$$O_B(\upsilon)\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = J_{TP}\begin{bmatrix} 0_{3x1} \\ -q \\ p \\ 0 \end{bmatrix}\Omega \tag{1.14}$$

where Ω is defined as the propeller's overall speed $[rad\ s^{-1}]$, O_B is the gyroscopic propeller matrix, J_{TP} is the total rotational moment of inertia around the propeller axis calculated in the next section $[N\ m\ s^2]$. Additionally Ω_j is the speed of the propeller $[rad\ s^{-1}]$ as defined in the figure 1.1.

$$\Omega = (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \tag{1.15}$$

The third contribution takes into account the forces and torques produced by the main movement inputs. From aerodynamics consideration, it follows that both torques and forces are proportional to the square propellers' speed. Therefore the movement vector is defined as follows.

$$U_B(\Omega) = [0\ 0\ U_!U_2U_3U_4]^T \tag{1.16}$$

Where $U_!$, U_2, U_3 and U_4 are the throttle, roll, pitch and the yaw respectively.

$$U1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{1.17}$$
$$U2 = lb(\Omega_4^2 - \Omega_2^2) \tag{1.18}$$
$$U3 = lb(\Omega_3^2 - \Omega_1^2) \tag{1.19}$$
$$U4 = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \tag{1.20}$$

In these relationships b is defined as the thrust factor $[N\ s^2]$, d is the drag factor $[N\ m\ s^2]$ and l is the distance between the center of the quadcopter and the center of the propeller $[m]$.

These equations are with respect with the body-fixed frame. This reference is widely used in 6 degree of freedom rigid bogy equations, however in this case the equations will be expressed in terms of a hybrid frame. That reference will be used because it's easy to express the dynamics combined with the control. In this frame a generalized velocity vector, ζ, would be defined following the next equation.

$$\zeta = \left[\dot{X}\ \dot{Y}\ \dot{Z}\ p\ q\ r\right]^T \tag{1.21}$$

The dynamics in the hybrid frame can be rewritten as follows.

$$M_H\dot{\zeta} + C_H(\zeta)\zeta = G_H + O_H(\zeta)\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\\ \Omega_4\end{bmatrix} + U_H \tag{1.22}$$

The system inertia with respect the hybrid frame is equal to the one with respect the fixed-body frame, so $M_H = M_B$. The Coriolis-centripetal matrix is defined as

$$C_H(\zeta) = \begin{bmatrix}0_{3x3} & 0\\ 0_{3x3} & -S(I\omega^B)\end{bmatrix} \tag{1.23}$$

The gravitational contribution with respect the hybrid frame is

$$G_H = -[0\ 0\ mg\ 0\ 0\ 0]^T \tag{1.24}$$

The gyroscopic effects by the propeller rotation are unvaried because it affects only the angular equations referred to the body-fixed frame. Finally the movement vector is different because the input U_1 affects all the linear equations through the rotation matrix.

$$U_H(\Omega) = \begin{bmatrix}R_\Theta & 0_{3x3}\\ 0_{3x3} & I_{3x3}\end{bmatrix}U_B(\Omega) \tag{1.25}$$

Isolating the derivate generalized velocity in the dynamics equation with respect to the hybrid frame, the system can be modelized as

This leads to the following equations.

$$\dot{\phi} = \frac{d\phi}{dt} \tag{1.26}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{I_{yy}-I_{zz}}{I_{xx}} - \frac{J_{TP}}{I_{xx}}\dot{\theta}\Omega + \frac{lb}{I_{xx}}U2 \tag{1.27}$$

$$\dot{\theta} = \frac{d\theta}{dt} \tag{1.28}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\frac{I_{zz}-I_{xx}}{I_{yy}} + \frac{J_{TP}}{I_{yy}}\dot{\phi}\Omega\frac{lb}{I_{yy}}U3 \tag{1.29}$$

$$\dot{\psi} = \frac{d\psi}{dt} \tag{1.30}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{I_{xx}-I_{yy}}{I_{zz}} + \frac{d}{I_{zz}}U4 \tag{1.31}$$

$$\dot{x} = \frac{dx}{dt} \tag{1.32}$$

$$\ddot{x} = (\cos\phi\cos\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}(U1) \tag{1.33}$$

$$\dot{y} = \frac{dy}{dt} \tag{1.34}$$

$$\ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \frac{1}{m}(U1) \quad \textbf{(1.35)}$$

$$\dot{z} = \frac{dz}{dt} \quad \textbf{(1.36)}$$

$$\ddot{z} = -g + \cos\phi \cos\theta \frac{1}{m} U1 \quad \textbf{(1.37)}$$

The inputs of this system will be the rotational speed of the propellers instead of the voltage input in the DC rotor.

1.2. LPV modeling

The LPV paradigm was introduced in the PhD. Thesis of Shamma [2] for the analysis of the control design practice of “gain-scheduling”. The nonlinear system is described as a parameterized linear system, where these parameters depend on the state; however in the LPV model this dependence is ignored. The parameters are considered exogenous and varying within a bounded region. These systems can be formulated as:

$$\dot{x} = A(\rho)x + B(\rho)u, \quad \rho \in \Omega \quad \textbf{(1.38)}$$

where ρ is the state depended parameter varying in the region Ω. Some properties of LPV systems in the gain scheduling context are:

- There is a relationship between the parameter and the states such that the LPV description and the nonlinear system are equal.

$$A(\sigma(x))x + B(\sigma(x))u = f(u,x) \quad \textbf{(1.39)}$$

- The relation $\sigma(x)$ depends only on measured signals.
- The relation $\sigma(x)$ is known.
- The LPV description is as close as possible to the nonlinear for all the values in the region Ω.

The first property ensures that the trajectories of the original nonlinear system are also trajectories in the LPV system. The second and third properties ensure that the parameters are available for the controller and that an explicit nonlinear feedback controller is obtainable from the LPV description.

Finding a nonlinear description from a nonlinear system is a non-trivial task. One method is to hide the nonlinearities in parameters. So, depending how these parameters are defined, a system can have different LPV descriptions.

A LPV system might be seen as an extension of a linear time invariant system (LTI system) as they coincide when the parameter is known in advance.

In this project, the LPV description will be a system with affine parameter dependence.

$$A(\rho) = A_0 + \sum_{i=1}^{\rho} A_i \rho_i \tag{1.40}$$

The parameter ρ is bounded and at least piecewise continuous. These bounds can be bounded as a hyper-cube.

$$\rho \in \Gamma = \left\{ \rho \middle| \underline{\rho}_i \leq \rho_i \leq \overline{\rho}_i, i = 1, \dots, \rho \right\} \forall t \geq t_0 \tag{1.41}$$

The time derivative of the parameter is assumed to be bounded also in the same fashion in a hyper-cube. The affinity dependence can be transformed into a interpolation of the vertices of Γ. This is called polytopic dependence. The polytopic coordinates are denoted as $\alpha \epsilon \Re^{2\rho}$ and these varies as follows,

$$\sum_{i=1}^{2\rho} \alpha_i = 1, \quad \alpha_i > 0 \tag{1.42}$$

1.3. Quadrotor LPV model

Using the method explained previously, the following expressions can be obtained.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = A\left(\Psi\big(p(t)\big)\right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + B\left(\Psi\big(p(t)\big)\right) \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{1.43}$$

where

$$A\left(\Psi\big(p(t)\big)\right) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_3 & 0 & 0 & 0 & a_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_5 & 0 & a_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.44}$$

$$B\left(\Psi\left(p(t)\right)\right) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 \\ 0 & 0 & 0 & 0 \\ b_5 & b_6 & b_7 & b_8 \\ 0 & 0 & 0 & 0 \\ b_9 & b_{10} & b_{11} & b_{12} \\ 0 & 0 & 0 & 0 \\ b_{13} & b_{14} & b_{15} & b_{16} \end{bmatrix} \tag{1.45}$$

$$a_1 = \frac{I_{yy} - I_{zz}}{2I_{xx}}\dot{\theta}$$
$$a_2 = \frac{I_{yy} - I_{zz}}{2I_{xx}}\dot{\psi}$$
$$a_3 = \frac{I_{zz} - I_{xx}}{2I_{yy}}\dot{\phi}$$
$$a_4 = \frac{I_{zz} - I_{xx}}{2I_{yy}}\dot{\psi}$$
$$a_5 = \frac{I_{xx} - I_{yy}}{2I_{zz}}\dot{\phi}$$
$$a_6 = \frac{I_{xx} - I_{yy}}{2I_{zz}}\dot{\theta}$$

$$b_1 = \frac{J_{TP}}{I_{xx}}\ \dot{\theta}$$

$$b_2 = -\frac{J_{TP}}{I_{xx}}\ \dot{\theta} - \frac{lb}{I_{xx}}\Omega_2$$
$$b_3 = \frac{J_{TP}}{I_{xx}}\ \dot{\theta}$$
$$b_4 = -\frac{J_{TP}}{I_{xx}}\ \dot{\theta} + \frac{lb}{I_{xx}}\Omega_4$$
$$b_5 = -\frac{J_{TP}}{I_{yy}}\ \dot{\phi} - \frac{lb}{I_{yy}}\Omega_1$$
$$b_6 = \frac{J_{TP}}{I_{yy}}\ \dot{\phi}$$
$$b_7 = -\frac{J_{TP}}{I_{yy}}\ \dot{\phi} + \frac{lb}{I_{yy}}\ \Omega_3$$
$$b_i = \frac{d}{I_{zz}}\ \Omega_i, i = 8, \dots, 11$$
$$b_j = \cos\phi\cos\theta\frac{b}{m}\Omega_j, \quad j = 12, \dots, 16$$

Knowing that

$$\Omega_i \in \{100{,}500\}\,, \quad i = 1{,}2{,}3{,}4$$
$$\phi, \theta \in \left\{0, \frac{\pi}{3}\right\}$$
$$\psi \in \{-\pi, \pi\}$$
$$\dot{\phi}, \dot{\theta}, \dot{\psi} \in \{-0.5{,}0.5\}$$

The parameters can be bounded as

$$a_{1,}a_2 \in \left\{-\frac{I_{yy}-I_{zz}}{4I_{xx}}, \frac{I_{yy}-I_{zz}}{4I_{xx}}\right\}$$
$$a_3, a_4 \in \left\{-\frac{I_{zz}-I_{xx}}{4I_{yy}}, \frac{I_{zz}-I_{xx}}{4I_{yy}}\right\}$$
$$a_5, a_6 \in \left\{-\frac{I_{xx}-I_{yy}}{4I_{zz}}, \frac{I_{xx}-I_{yy}}{4I_{zz}}\right\}$$
$$b_1, b_3 \in \left\{\frac{-J_{TP}}{2I_{xx}}, \frac{J_{TP}}{2I_{xx}}\right\}$$

$$b_2 \in \left\{-\frac{J_{TP}}{2I_{xx}} - 500\frac{lb}{I_{xx}}, \frac{J_{TP}}{2I_{xx}} - 100\frac{lb}{I_{xx}}\right\}$$
$$b_4 \in \left\{-\frac{J_{TP}}{2I_{xx}} + 100\frac{lb}{I_{xx}}, \frac{J_{TP}}{2I_{xx}} + 500\frac{lb}{I_{xx}}\right\}$$

$$b_5 \in \left\{-\frac{J_{TP}}{2I_{yy}} - 500\frac{lb}{I_{yy}}, \frac{J_{TP}}{2I_{yy}} - 100\frac{lb}{I_{yy}}\right\}$$
$$b_6, b_7 \in \left\{\frac{-J_{TP}}{2I_{yy}}, \frac{J_{TP}}{2I_{yy}}\right\}$$
$$b_7 \in \left\{-\frac{J_{TP}}{2I_{yy}} + 100\frac{lb}{I_{yy}}, \frac{J_{TP}}{2I_{yy}} + 500\frac{lb}{I_{yy}}\right\}$$
$$b_i \in \left\{100\frac{d}{I_{zz}}, 500\frac{d}{I_{zz}}\right\}, \quad i = 8, \dots, 11$$
$$b_j \in \left\{25\frac{b}{m}.\,500\frac{b}{m}\right\}, \quad j = 12, \dots, 16$$

Chapter 2

Standard Identification Methods

In this chapter, space state methods to identify a model will be provided and used in our case.

2.1. Identification of the System State Space model

The prediction error method consists in building mathematical models of dynamic models using measured input-output data. To do that, first of all the model of the system has been implemented in Simulink.

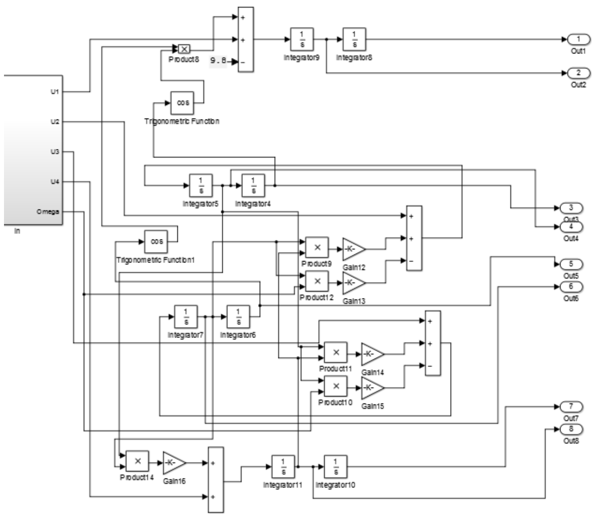

Figure 2.1 Simulink Model

Then using the data that has been extracted from the simulation, the model can be found using the Matlab function `ssest` and `n4sid`.

```
pss12 = ssest(data,8, 'Form','canonical','DisturbanceModel', 'none');

n4s12 = n4sid(data,8,'Form',' canonical' ,'DisturbanceModel', 'none',
'Ts',0, Opt);
```

In both cases, the result is given in an observable canonical form and there are no disturbances in the model because the simulation is perfect. Then the results are studied using the function `idssdata` that returns the matrices of the space-state model and its uncertainties.

2.2. Ident

2.2.1. Probing the method

A first example has been set in order to probe the method. In this case, the space state model is the following,

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -6 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{2.1}$$

```
Continuous-time state-space model:
    dx/dt = A x(t) + B u(t) + K e(t)
     y(t) = C x(t) + D u(t) + e(t)
A =
                 x1           x2           x3
   x1  -2.606e-07            1  -1.345e-06
   x2   1.345e-06    6.465e-06            1
   x3          -1           -5           -6
B =
                 u1
   x1   2.606e-07
   x2  -1.345e-06
   x3           1
```

Figure 3.2 Result of the identification using Matlab

The method identifies correctly the value and the variance of this is much lower. The next scenario would be studying the case of a model with a parameter that has a time variance.

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -6 + \sin(t) \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{2.2}$$

```
Continuous-time state-space model:
    dx/dt = A x(t) + B u(t) + K e(t)
     y(t) = C x(t) + D u(t) + e(t)
A =
                x1           x2           x3
   x1     0.003081         1.01       0.0172
   x2   -0.0001602    -0.001294       0.9992
   x3      -0.9873       -4.982       -5.871
B =
               u1
   x1   -0.004119
   x2   0.0004781
   x3      0.9938
```

Figure 3.3 Results of the identification in a model with parameter that evolves with time

As can be seen, this tool converges to a value and if the deviations are studied the parameter has a value of $-5.871 \pm\ 0.0149$. With that can be concluded that this method is not a good one to identify time varying parameters because it can't capture the dynamics of the time varying parameter.

2.2.2. Using the model

If the model of the quadrotor, however can't find the model because is nearly singular and the uncertainty is much bigger than the nominal value. To solve that a method that doesn't tries to identify the space state model but the parameters in a system whose form is known is preferred.

Chapter 3

Bayesian identification

In this chapter, an introduction to Bayesian identification using Dynamic Linear Models is provided. This method is used because of the flexibility in modeling structural changes and because this method is defined sequentially as new observed data is available.

3.1. Introduction to Bayesian Inference

In every real data analysis, even with accurate deterministic models, there are uncertainties because of measurement errors for example. To modelize that, in Bayesian statistics the uncertainty is described by means of probability. Then using probability theory it is possible to assign these probabilities coherently.

3.1.1. Bayesian Inference

The Bayesian inference is a method in which the Bayes rule (3.1) is used to update the probability estimate as additional data is acquired.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.1}$$

In this expression, A represents the event of interest, while B is an experimental result. Knowing the probability of A, B and B given A, the probability of learning the probability of learning about an event from the experimental data is solved by computing $P(A/B)$.

In statistical inference, the data is the result of sampling data, represented by a vector Y. The variables of interest are represented by the vector θ of parameters of the model. Bayesian inference consists on θ consists of calculating its conditional distribution given the sampling results. Supposing a conditional distribution $\pi(y|\theta)$ for Y given θ and a prior distribution $\pi(\theta)$ that express the uncertainty of the parameter, it's possible to generalize the Bayes' theorem to compute the conditional density of θ given y, where $Y=y$.

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\pi(y)} \tag{3.2}$$

Where $\pi(y)$ is the marginal distribution of Y.

$$\pi(y) = \int \pi(y|\theta)\pi(\theta)d\theta \tag{3.3}$$

3.2. Dynamic Linear Models

A dynamic linear model (DLM) is a special case space model, which is linear and Gaussian. Estimations of these models can be obtained recursively using Kalman filters.

A dynamic linear model is specified by a prior for the p-dimensional state vector at *t=0,* θ_0, and a pair equations for each *t>1.* These equations describe the outputs of the system, Y_t, and the states of the system, θ_t, in a given time.

$$\theta_0 \sim \mathcal{N}_p(m_0, C_0) \tag{3.2}$$
$$Y_t = F_t\theta_t + v_t, \quad v_t \sim \mathcal{N}_m(0, V_t) \tag{3.3}$$
$$\theta_t = G_t\theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p(0, W_t) \tag{3.4}$$

In these equations G_t and F_t are known matrices and w_t and v_t are independent Gaussian random vectors with known variance matrices and mean zero. The prior is assumed to be independent of the Gaussian vectors.

3.3. Models with unknown parameters

The matrixes introduced in the previous point (Gt, Ft, Vt and W_t) usually are not completely known. These can be modeled to depend on a parameter that can have a temporal evolution.

3.3.1. Maximum Likelihood Estimator

Supposing that $Y_1, \dots, Y_n$ are n random vectors whose distribution depends on an unknown parameter, ψ. The joint density of these observations for a known value of the parameter will be denoted as $p(y_1, \dots, y_n;\ \psi)$. For a DLM the joint density can be written as

$$p(y_1, \dots, y_n;\ \psi) = \prod_{t=1}^{n} p(y_t | y_{1:t-1};\ \psi) \tag{3.5}$$

where $p(y_t | y_{1:t-1};\ \psi)$ is the conditional density of y_t given the data up to the previous time step and assuming that ψ is the value of the unknown parameter. The terms of the right hand side of the previous equation (3.7) are Gaussian densities with mean f_t and variance Q_t. Therefore it is possible to write the loglikelihood as

$$l(\psi) = -\frac{1}{2}\sum_{t=1}^{n} \log|Q_t| - \frac{1}{2}\sum_{t=1}^{n}(y_t - f_t)' Q_t^{-1}(y_t - f_t) \tag{3.6}$$

where the f_t and Q_t depends implicitly on ψ. The expression (3.8) can be numerically maximized at

$$\hat{\psi} = \text{argmax}_{\psi}\, l(\psi) \qquad \textbf{(3.7)}$$

The variance of the MLE is approximated by the matrix H^{-1}, where H is denoted by the Hessian of l(ψ) evaluated at $\psi = \hat{\psi}$.

The likelihood function of a DLM can have various local maxima, which means that the result can depend on the starting point.

3.3.2. Parameter Learning

3.3.2.1. Bayesian Estimation of a Space State Model

From a Bayesian perspective, the aim of state estimation is to find the probability function of the state given the sampling data, $\pi(\theta_k|y_{1:k})$. Assuming the initial conditions as a probability distribution $\pi(y_0|\theta_0)$, the result desired can be found sequentially,

$$\pi(\theta_k|y_{1:k-1}) = \int \pi(\theta_k|\theta_{k-1})\, \pi(\theta_{k-1}|y_{1:k-1}) d\theta_{k-1} \qquad \textbf{(3.8)}$$

and updated using

$$\pi(\theta_k|y_{1:k}) = \frac{\pi(y_k|\, \theta_k)\pi(\theta_k|y_{1:k-1})}{\pi(y_k|y_{1:k})} \qquad \textbf{(3.9)}$$

where $\pi(y_k|y_{1:k})$ is a normalizing function independent of the state, $\pi(\theta_k|\theta_{k-1})$ is defined by the state function and $\pi(y_k|\, \theta_k)$ by the the measurement function.

3.3.2.2. Particle Filter

The basic idea behind the particle filters is to approximate $\pi(\theta|y)$ using a set of random samples, called particles, θ_k^i with associated weights w_k^i.

$$\pi(\theta_k|y_{1:k}) = \sum_{i=1}^{N} w_k^i\, \delta(\theta_k - \theta_k^i) \qquad \textbf{(3.10)}$$

where $\delta(x)$ is equal to 1 when the $x = 0$, and otherwise equals 0. In this case, $\pi(\theta_k|y_{1:k-1})$ isn't a conventional form of density function. Therefore importance sampling is used to obtain new particles and the weight. To do so, a importance density, $g(\theta_k|y_{1:k})$, must be defined. Then the weight can be computed as

$$w_k^i \propto \frac{\pi(\theta_k|y_{1:k})}{g(\theta_k|y_{1:k})} \qquad \textbf{(3.11)}$$

For a sequential estimation problem, at a point k, the particles can be calculated as

$$\pi(\theta_k|y_{1:k}) \propto \pi(\theta_k, y_{1:k}|y_{1:k-1}) \quad \textbf{(3.12)}$$

Computing that, can be probed that

$$\pi\left(\theta_k^i|y_{1:k}\right) = w_{k-1}^i \pi(y_k|\ \theta_k)\pi\left(\theta_k|\theta_{k-1}^i\right)\delta(\theta_k - \theta_k^i) \quad \textbf{(3.13)}$$

The importance density used is the one proposed by Pitt and Shepard

$$g\left(\theta_k^i|y_{1:k}\right) \propto w_{k-1}^i \pi\left(y_{1:k}|\hat{\theta}_k\right)\pi\left(\theta_k|\theta_{k-1}^i\right)\, \delta\left(\theta_k - \theta_k^i\right) \quad \textbf{(3.14)}$$

where $\hat{\theta}_k$ has been defined as the expected value. Considering all that, the non-normalized weight of the nth draw can be computed as

$$w_k^n = \frac{\pi(y_k|\ \theta_k^n)}{\pi(y_{1:k}|\hat{\theta}_k^n)} \quad \textbf{(3.15)}$$

3.3.2.3. Liu and West

Liu and West is a particle filter method in which there are unknown parameters that will be estimated from the data. These parameters will be denoted by a vector ψ. To do that, this method constructs a target distribution at time k that is continuous for θ_k and ψ. Using importance sampling, values of ψ from a continuous importance density can be drawn, and the values from the previous time step discrete approximation can be forgoten. This approximated can be computed as

$$\pi_{t-1}(\theta_{k-1}|\ \psi) = \sum_{i=1}^{N} w_k^i\, \delta_{(\theta_k, \psi)} \quad \textbf{(3.16)}$$

Marginally

$$\pi_{t-1}(\psi) = \sum_{i=1}^{N} w_k^i\, \delta_{\psi} \quad \textbf{(3.17)}$$

In the Liu and West method δ_{ψ} is replaced by a Normal distribution.

$$\pi_{t-1}(\psi) = \sum_{i=1}^{N} w_k^i\, \mathcal{N}(\psi; m^i, h^2\Sigma) \quad \textbf{(3.18)}$$

where,

$$m^i = a\psi^i + (1-a)\bar{\psi} \quad \textbf{(3.19)}$$

If $a \in \{0,1\}$ and $a^2 + h^2 = 1$, can be probed that

$$E(\psi) = \bar{\psi} \quad \textbf{(3.20)}$$

$$Var(\psi) = \Sigma \quad \textbf{(3.21)}$$

Despite any a in (0,1) can be used, it is recommended to use an a in (0.974,0.995). Using this transformation, the discrete distribution can be transformed as

$$\pi(\theta_k|y_{1:k}) = \sum_{i=1}^{N} w_k^i \mathcal{N}(\psi; m^i, h^2\Sigma)\, \delta(\theta_k - \theta_k^i) \quad \textbf{(3.22)}$$

And using the same method as before, can be probed

$$\pi\left(\theta_k^i \middle| y_{1:k}\right) = w_{k-1}^i \pi(y_k|\, \theta_k, \psi)\pi\left(\theta_k \middle| \theta_{k-1}^i, \psi\right)\mathcal{N}(\psi; m^i, h^2\Sigma)\delta(\theta_k - \theta_k^i) \quad \textbf{(3.23)}$$

Using the importance density proposed by Pitt and Shepard

$$g\left(\theta_k^i \middle| y_{1:k}\right) \propto w_{k-1}^i \pi\left(y_k \middle|\, \theta_k = \hat{\theta}_k^i, \psi = \mathrm{m}^i\right)\pi\left(\theta_k \middle| \theta_{k-1}^i, \psi\right)\mathcal{N}(\psi; m^i, h^2\Sigma)\, \delta\left(\theta_k - \theta_k^i\right) \quad \textbf{(3.24)}$$

A sample of $g\left(\theta_k^i \middle| y_{1:k}\right)$ can be obtained by iterating for every particle (n=1...N) the following steps.

- Draw a variable I_n such as

$$P(I_n = i) \propto w_{k-1}^i \pi\left(y_k \middle|\, \theta_k = \hat{\theta}_k^i, \psi = \mathrm{m}^i\right) \quad \textbf{(3.25)}$$

- Given $I_n = i$, draw $\mathcal{N}(\psi; m^i, h^2\Sigma)$ and set $\psi = \psi^n$
- Given $I_n = i$ and set $\psi = \psi^n$, draw

$$\theta_k^n \sim \pi\left(y_k \middle|\, \theta_k = \hat{\theta}_k^i, \psi = \mathrm{m}^i\right) \quad \textbf{(3.26)}$$

and set

$$\theta_{0:k}^i = (\theta_{0:k-1}^i, \theta_k^i) \quad \textbf{(3.27)}$$

The importance weight of the nth draw is

$$w_k^n = \frac{\pi\left(y_k \middle|\, \theta_k = \theta_k^i, \psi = \psi^i\right)}{\pi\left(y_k \middle|\, \theta_k = \hat{\theta}_k^{I_n}, \psi = \mathrm{m}^{I_n}\right)} \quad \textbf{(3.28)}$$

Normalizing the weight, the posterior distribution can be approximated. This method must be used with a parameter ψ that is one-dimensional. If the parameter is a vector, such as $\psi = (\psi_1, \psi_2)$ and defining the multivariate distribution

$$f(\psi, \gamma) = \mathrm{f}_1(\psi_1; \gamma_1)\mathrm{f}_2(\psi_2; \gamma_2), \quad \gamma = (\gamma_1, \gamma_2) \tag{3.29}$$

where γ_j can be set in such a way that $\mathrm{f_j}(\cdot, \gamma_j)$ has a specific mean and variance.

$$\bar{\psi} = \begin{bmatrix} \bar{\psi}_1 \\ \bar{\psi}_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{bmatrix} \tag{3.30}$$

Then for i=1,..,N

$$\mu_j^i = a\psi_j^i + (1-a)\bar{\psi}_j \tag{3.31}$$

$$\sigma^2{}_j^{(i)} = h^2\Sigma \tag{3.32}$$

So using the mixture is

$$\sum_{i=1}^{N} w_k^i \mathrm{f}_1\left(\psi_1; \gamma_1^i\right)\mathrm{f}_2\left(\psi_2; \gamma_2^i\right)) \tag{3.33}$$

This technique can be generalized to produce kernels with more than two factors because ψ_j can be also considered to be multivariate. Furthermore Liu and West method can be used to identify multivariate parameters by changing the μ_j^n and $\sigma^2{}_j^{(n)}$ values instead of a and m.

Summarizing this method,

0. Initialize the parameter priors independently from its distribution, set the weight as $w_0^i = 1/N$ and

$$\pi_0 = \sum_{i=1}^{N} w_0^i\, \delta_{(\theta_0^i, \psi^i)} \tag{3.34}$$

1. For k=1…K
   1.1 For i=1…N and j=1,2:

   - Compute $\bar{\psi}_j = E\left(\psi_\mathrm{j}\right)_{\pi_{k-1}}$ and $\Sigma_j = Var_j(\psi_\mathrm{j})$ and for i=1...N, set μ_j^i and $\sigma^2{}_j^{(i)}$ using the equations (3.31) and (3.32) and set

$$\mu^i = (\mu_1^i, \mu_2^i) \tag{3.35}$$

$$\theta_k^i = E\left(\theta_k \middle| \theta_{k-1} = \theta_{k-1}^i, \psi = \mu^i\right) \tag{3.36}$$

   - Solve for γ_j^i the system of equations

$$E_{\mathrm{f_j}\left(\cdot,\gamma_j^i\right)}\left(\psi_j^i\right) = \mu_j^i \tag{3.37}$$

$$Var_{\mathrm{f_j}\left(\cdot,\gamma_j^i\right)}\left(\psi_j^i\right) = \sigma_j^{(i)} \tag{3.38}$$

1.2 For n=1…N

- Draw I_n as

$$P(I_n = i) \propto w_{k-1}^i \pi\left(y_k \middle|\, \theta_k = \hat{\theta}_k^i, \psi = \mathrm{m}^i\right) \tag{3.39}$$

- For j=1,2, draw ψ_j^i from $\mathrm{f_j}\left(\cdot,\gamma_j^i\right)$
- Draw θ_k^n from $\pi\left(\theta_k \middle|\, \theta_{k-1} = \theta_{k-1}^{I_n}, \psi = \psi^n\right)$ and set

$$\theta_k^n = (\theta_{0:k-1}^{I_n}, \theta_k^n) \tag{3.40}$$

- Set the weight as in the equation (3.28)

1.3 Set $\pi_k = \sum_{i=1}^N w_k^i \, \delta_{(\theta_k^i,\psi^i)}$

3.3.2.4. Parameter learning with changepoints

Liu and west is constrained to learning parameters for static parameters. In order to track, the motion of the target requires the parameters of the model to evolve in conjunction with the target's motion. To do it's assumed that there exists m changepoints in the observations. The probability of a changepoint at $k = i$ is β which will be considered continuous. Doing so, Liu and West can be used assuming that the parameters are static and once there is a maneuver, the parameter will be updated by drawing new priors to learn the new value of the parameters. In this case there will be 2 distributions used in the resampling, one with the learnt parameters and another with the new data.

To use the changepoints, the algorithm must be modified in this way

1. For i=1...N
   1.1 Sample a prior ${\theta_k^i}^*$
   1.2 Calculate

$$w_{k,1}^i \propto (1-\beta) w_{k-1}^i \pi\left(y_k \middle|\, \mu_k^i, \mathrm{m}^i\right) \tag{3.41}$$

$$w_{k,2}^i \propto \beta w_{k-1}^i \pi\left(y_k \middle|\, \mu_k^i, {\theta_k^i}^*\right) \tag{3.42}$$

2. For i=1...N
   - Draw $I_n \propto w_{k,1}^i + w_{k,2}^i$

3. For i=1...N

- Define $W_k^i = {w_{k,1}^i}\big/{w_{k,1}^i + w_{k,2}^i}$ and $u \sim \mathcal{U}(0,1)$

3.1 If $W_k^i > u$ update the parameters and assign the weights as before

3.2 Else, define $\theta_k^i = \theta_k^{i\,*}$, propagate using this assumption and compute the weight as

$$w_k^n = \frac{\pi\left(y_k \middle| \, \theta_k = \theta_k^i, \psi = \psi^i\right)}{\pi\left(y_k \middle| \, \theta_k = \hat{\theta}_k^{In}, \psi = \theta_k^{i\,*}\right)} \quad \textbf{(3.43)}$$

Chapter 4

Results using probabilistic methods

In this chapter, various probabilistic identification algorithms are provided. Then they are tested. It should be taken into account that in this case the bounds of the parameters are the information that is looked for.

4.1. Maximum likelihood in R

In the case studied, it is used the R package “dlm”. This creates the class dlm that *uses the variables FF, V, GG, W, C0 and m0. For example, the following DLM

$$\theta_0 \sim \mathcal{N}_p(0{,}10) \quad \mathbf{(4.1)}$$

$$Y_t = \theta_t + v_t, \quad v_t \sim \mathcal{N}_m(0{,}1.4) \quad \mathbf{(4.2)}$$

$$\theta_t = \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p(0{,}0.2) \quad \mathbf{(4.3)}$$

Would be described as,

```
>model <-  dlm(m0 = 0, C0 = 10, FF = 1, V = 1.4, GG = 1, W = 0.2)
```

In order to determine its model using MLE methods, this model must be discretized, to do so the matrix G_t must be defined as $G_k = e^{\theta_t T_s}$, where T_s is the sampling time. To try to find the value of components of this matrix, the function dlmMLE of the same package has been used.

```
>fit = dlmMLE(y,init,modelDLM)
```

This function needs an initialization vector of the unknown parameters, and a build of the model that must be identified that depends on the unknown parameter,

```
>modelDLM <-  function(u)  {dlm(m0 = 0, C0 = 10, FF = 1, V = 1.4, GG = u[1], W = 0.2)}
```

4.1.1. First Test

A first example has been set in order to probe the method. The following dlm has been identified.

$$\theta_0 \sim \mathcal{N}_p\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) \quad \mathbf{(4.4)}$$

$$Y_t = \begin{bmatrix}1 & 0\end{bmatrix}\theta_t + v_t, \quad v_t \sim \mathcal{N}_m(0, V_t) \quad \mathbf{(4.5)}$$

$$\theta_t = \begin{bmatrix}x_1 & 1\\0 & 0\end{bmatrix}\theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p\left(0, x_3\begin{bmatrix}1 & x_2\\x_2 & x_2^2\end{bmatrix}\right) \quad \mathbf{(4.6)}$$

where $x_1 = 0.6$, $x_2 = 0.3$ and $x_3 = 0.05$.

The model used before, this has been identified correctly with an maximum error of around 10% ($x_1 = 0.599$, $x_2 = 0.269$ and $x_3 = 0.049$.).

Then the unknown parameter has been modified to be $x_1 = 0.6 + 0.1t$. Using the same method as before, the result is similar. However, if the parameter is defined as $x_1 = x_{1c} + x_{1t}t$ the MLE method can detect the varying component. With that it is possible to identify the varying components of a system using maximum likelihood.

4.1.2. Result with the quadrotor model

Then the quadrotor model has been linearized in a point, to adapt it into the DLM model, a filter has been used to put the B matrix into the A. With this n points of the matrix have been identified to try to study how large the data extracted from the simulation should be.

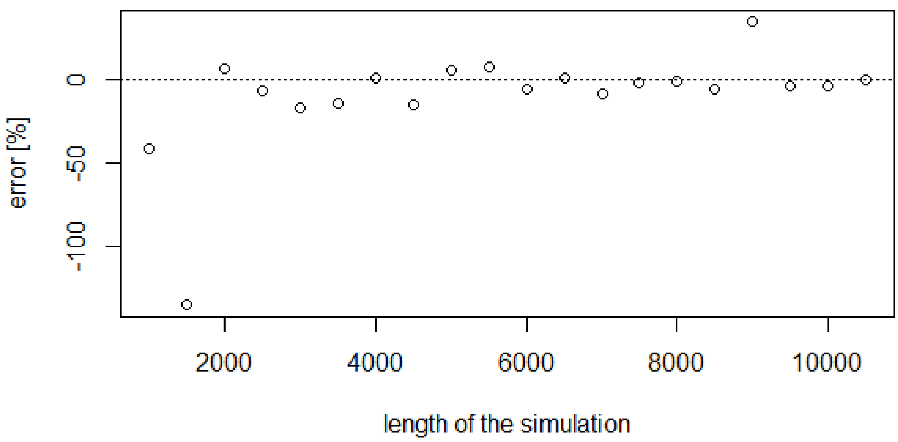

Figure 4.1 Relative error vs data length in a MLE identification for 1 parameter

If the number of parameters increases, increases also the length of the vector with data needed for identify the parameters. So it can be concluded that this method can be used to detect the parameters of a system whose dynamics are known; however the amount of data needed is really big.

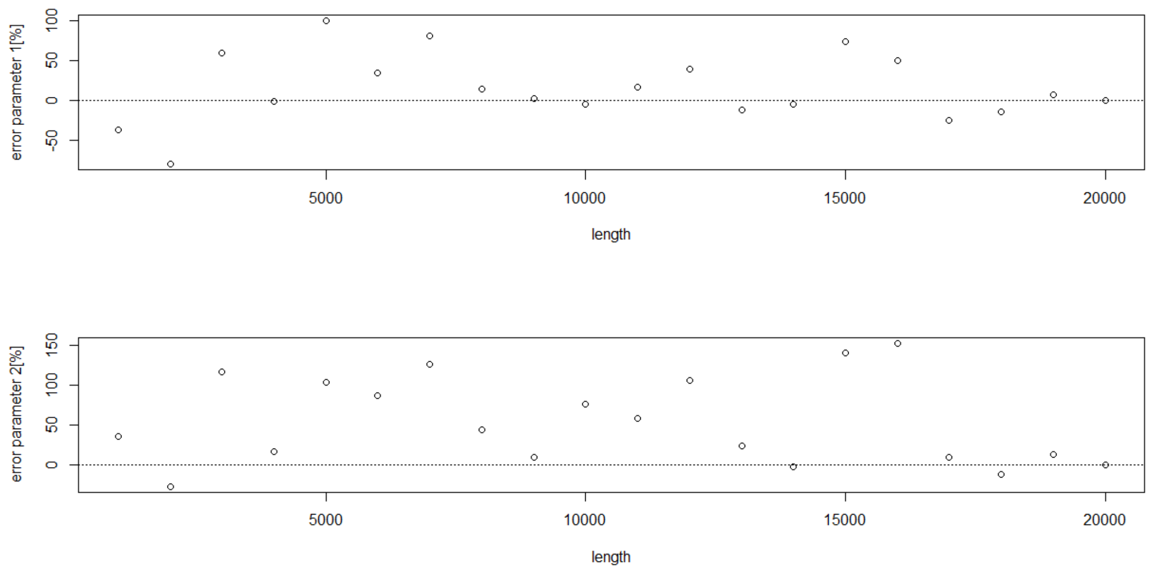

Figure 4.2 Relative error vs data length in a MLE identification for 2 parameters

4.2. Liu and West

4.2.1. Algorithm

In this case, a code following the algorithm proposed before has been used (figure 2.1 and 2.2).

The priors have been selected as a uniform distribution for the unknown parameter and a normal distribution for the system variables.

```
pfOutTheta[1, ] <- rnorm(N, mean = m0,
                        sd = sqrt(C0))
pfOutalpha[1,] <- runif(N,0,3)

wt[1, ] <- rep(1/N, N)

for (it in 2 : (n+1))
{

  meanalpha <- weighted.mean(pfOutalpha[it - 1, ], wt[it - 1,])
  varalpha <- weighted.mean((pfOutalpha[it - 1, ] - meanalpha)^2,
                          wt[it - 1,])
  mualpha <- a * pfOutalpha[it - 1, ] + (1 - a) * meanalpha


  Valpha<- h^2*sum((pfOutalpha[it-1,]-mualpha)^2)/N

  expTheta <- pfOutalpha[it-1,]*pfOutTheta[it-1,]

  probs <- wt[it - 1,] * dnorm(y[it - 1], sd = sqrt(v),
                             mean = pfOutTheta[it - 1, ])

  auxInd <- sample(N, N, replace = TRUE, prob = probs)
```

```
  pfOutalpha[it,] <- rnorm(N,mean=mualpha[auxInd],sd=varalpha)
  pfOutTheta[it,] <- rnorm(N,mean=pfOutalpha[it-1,auxInd]*pfOutTheta[it - 1, auxInd],
                           sd = sqrt(w))
  wt[it, ] <- exp(dnorm(y[it - 1], mean = pfOutTheta[it, ],
                        sd = sqrt(v),
                        log = TRUE) -
                  dnorm(y[it - 1], mean =
                        expTheta[auxInd],
                        sd = sqrt(v),
                        log = TRUE))

  wt[it, ] <- wt[it, ] / sum(wt[it, ])

}
```

4.2.2. Results

As a first approximation, a dlm in which the only parameter is unknown is used. As is shown in the following figure, the Liu and West method converges in a good result really quick.

$$\theta_0 \sim \mathcal{N}_p(0{,}10) \quad \textbf{(4.1)}$$
$$Y_t = \theta_t + v_t, \quad v_t \sim \mathcal{N}_m(0{,}2) \quad \textbf{(4.2)}$$
$$\theta_t = \alpha\theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p(0{,}1) \quad \textbf{(4.3)}$$

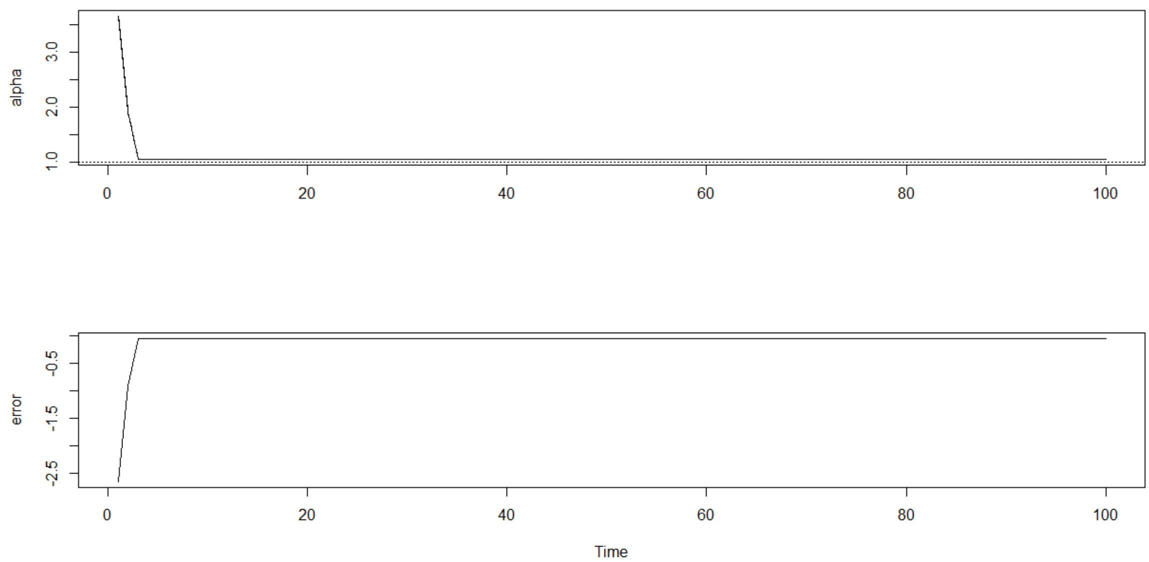

Figure 4.3 Parameter estimation and relative error for parameter estimation

As the figure shows, the Then, the variances of the uncertainties have been supposed unknown. The next figure shows that the identification keeps on being accurate.

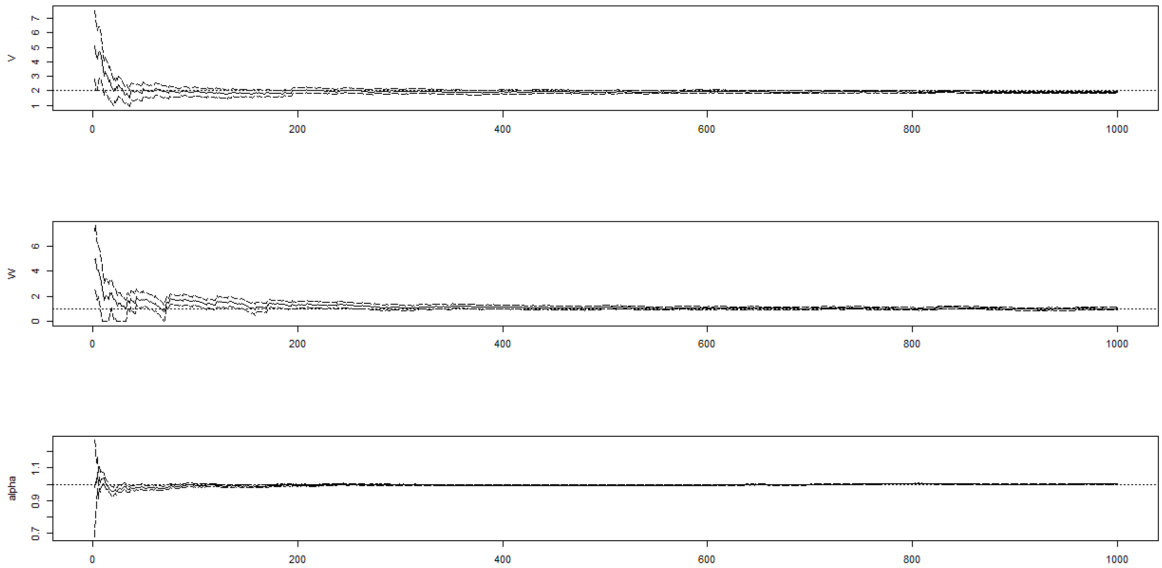

Figure 4.4 G_t and variances of the uncertainties identification using Liu and West

Therefore this method can accurately identify time continuous parameters. Then a test in which the parameter evolves following a slow sinusoidal has been tested.

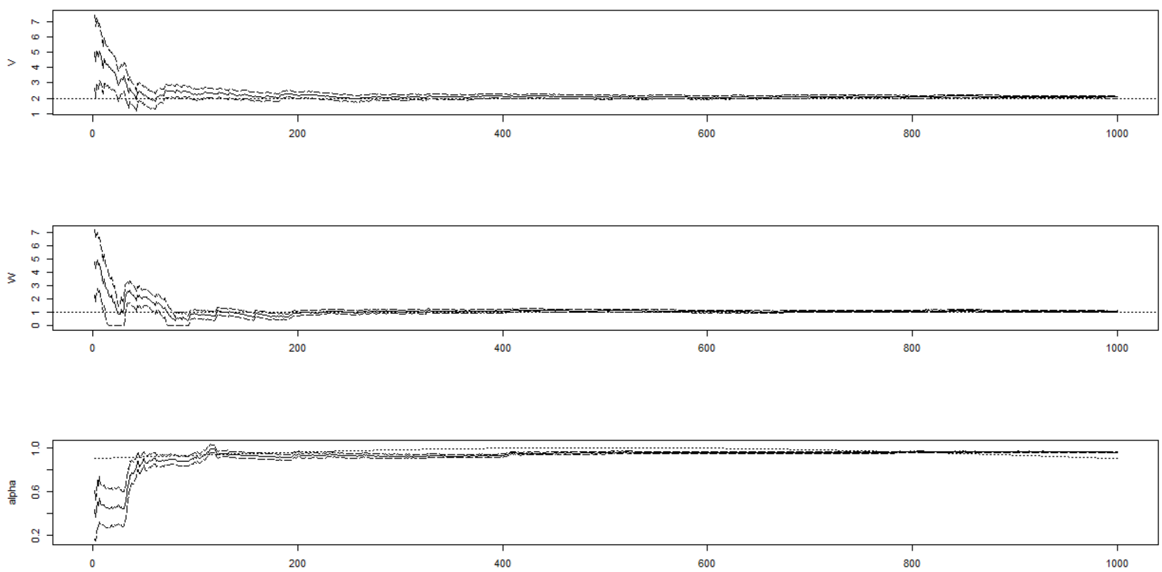

Figure 4.5 $G_t(t)$ and variances of the uncertainties identification using Liu and West

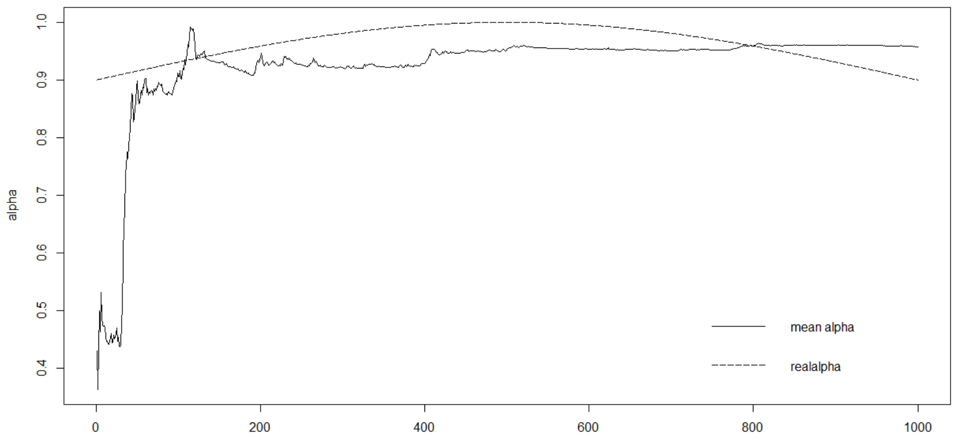

Figure 4.6 Detail of the identification of the varying parameter

As the figures show, the constant parameters keep on identifying correctly. However, that's not the case with the time varying one. To solve that, the changepoints have been added.

4.3. Liu and West with changepoints

4.3.1. Algorithm

In this case, the algorithm used in the section before has been updated to be able to changepoints and try to identify maneuvers in the parameter. To do so, a new prior is introduced in every step to be able to find the new values. However this introduces noise in the divergence components in the identification, in order to avoid these priors will be normal distributions which mean is the one of the computed for the parameter during the previous time-step.

```
for (it in 2 : (n+1))
{


  meanalpha <- weighted.mean(pfOutalpha[it - 1, ], wt[it - 1,])
  varalpha <- weighted.mean((pfOutalpha[it - 1, ] - meanalpha)^2,
                           wt[it - 1,])
  mualpha <- a * pfOutalpha[it - 1, ] + (1 - a) * meanalpha


  Valpha<- h^2*sum((pfOutalpha[it-1,]-mualpha)^2)/N

  expTheta <- pfOutalpha[it-1,]*pfOutTheta[it-1,]
  probalpha <- rnorm(N,mean=meanalpha,sd=sqrt(3*varalpha))

  w1 <- (1-b)*wt[it-1,]*dnorm(y[it-1], mean = expTheta,
                            sd = sqrt(v))
```

```
w2 <- b*wt[it-1,]*dnorm(y[it-1], mean = expTheta,
                        sd = sqrt(v))

ww <-w1/(w1+w2)
ww <- (ww)

auxInd <- sample(N, N, replace = TRUE, prob = w1+w2)

val[it]=varalpha-Valpha

for (i in 1:N){

  if (w2[i] == 0)
  {
    ww[i] <- 1
  }
  if (ww[i]>runif(1,0,1))
  {

    pfOutalpha[it,i] <-rnorm(1,mean=mualpha[auxInd[i]],sd=varalpha)
    pfOutTheta[it,i] <- rnorm(1,mean =pfOutalpha[it-1,auxInd[i]]*pfOutTheta[it - 1, auxInd[i]],
                              sd = sqrt(w))
    wt[it,i] <- exp(dnorm(y[it - 1], mean = pfOutalpha[it,i]*pfOutTheta[it,i],
                          sd = sqrt(v),
                          log = TRUE) -
                    dnorm(y[it - 1], mean =expTheta[auxInd[i]],
                            sd = sqrt(v),
                            log = TRUE))
  }
  else
  {

    setpoint[it-1]= setpoint[it-1]+1
    pfOutalpha[it,i] <- probalpha[auxInd[i]]
    pfOutTheta[it,i] <- rnorm(1,mean =pfOutalpha[it-1,auxInd[1]]*pfOutTheta[it - 1, auxInd[i]],
                              sd = sqrt(w))

    wt[it,i] <- exp(dnorm(y[it - 1], mean = pfOutalpha[it,i]*pfOutTheta[it,i],
                          sd = sqrt(v),
                          log = TRUE) -
                    dnorm(y[it - 1], mean =expTheta[auxInd[i]],
                            sd = sqrt(v),
                            log = TRUE))

  }
}


wt[it, ] <- wt[it, ] / sum(wt[it, ])

}
```

4.3.2. Results

To begin with it, the same test as in the previous section has been made. The prior used in the identification is a normal distribution with its center in the mean of the identified parameter in the previous time-step and a changepoint probability of 5%.

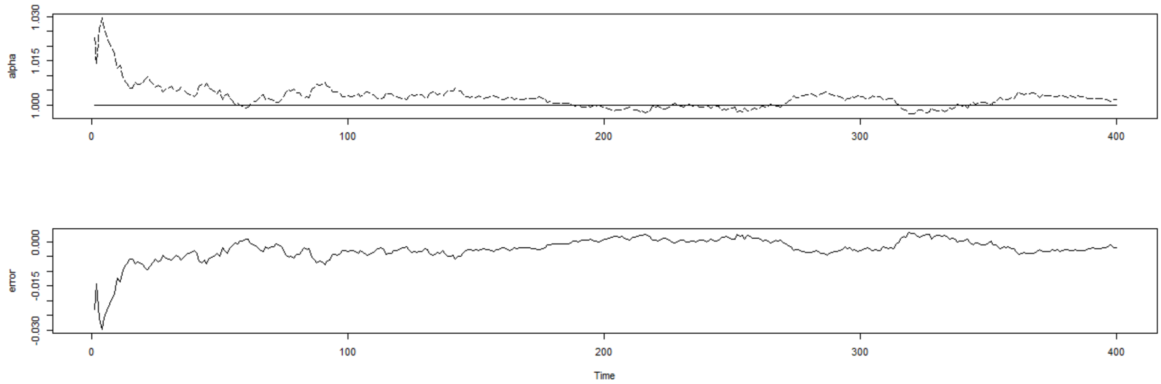

Figure 4.7 Identification and relative error of a parameter identified using changepoints

The figure shows that the method identifies correctly the parameter, however due to the changepoint probability there are some points in which a new prior is used and the system doesn't converge. To solve that, a smaller prior and changepoint probability should be taken.

Then a identification with an unknown parameter that follows a ramp form has been tested.

$$\theta_t = (a + bt)\theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p(0,2) \tag{4.4}$$

It has been used the same prior and changepoint probability as before. The following picture shows that the result is really good. That's because despite it has small peaks the interesting data is the bound and the identified parameter value are similar to the real one.

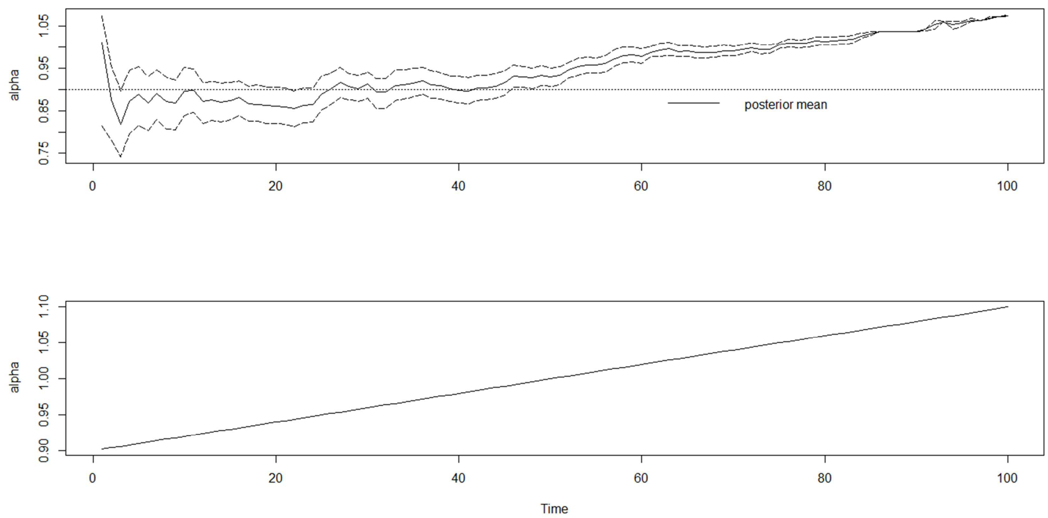

Figure 4.8 Identified parameter and real value in the case of a time varying parameter

Then this method can be used in order to identify a time varying parameter with various sharper time dependent forms. The prior selected for the changepoint and the probability is the same used in both the tests made before.

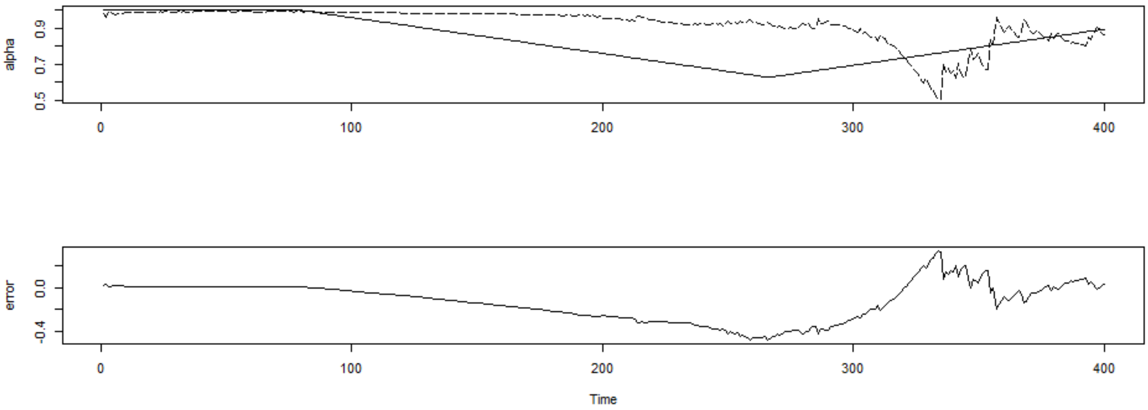

Figure 4.9 Identification and relative error of a time varying parameter identified using changepoints (β=5%)

As the figure shows, the identification doesn't change the set point quick enough. In order to asses this, a higher changepoint probability, β, can be used. By doing that, the system will be more sensitive and will use the prior distribution

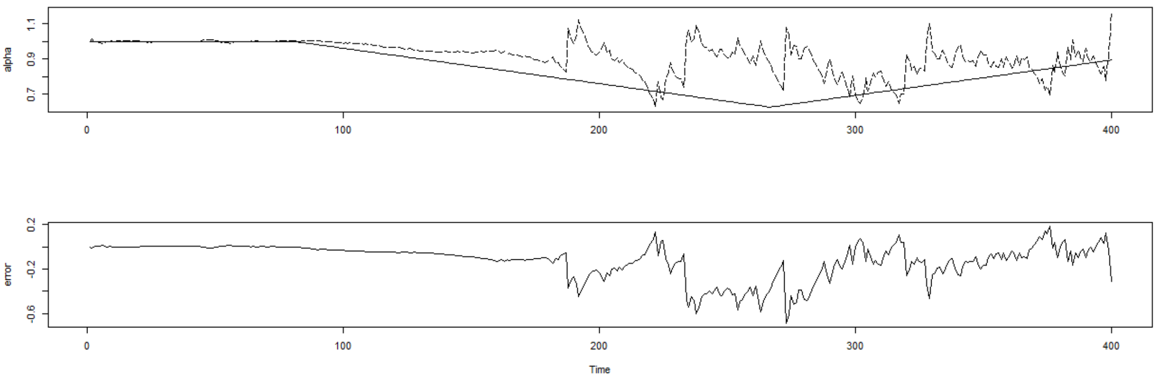

Figure 4.10 Identification and relative error of a time varying parameter identified using changepoints (β=10%)

Doing that the parameter isn't well identified and there are too much peaks in the process, some of them considerably big. To solve this smaller variance in the normal distribution has been used.

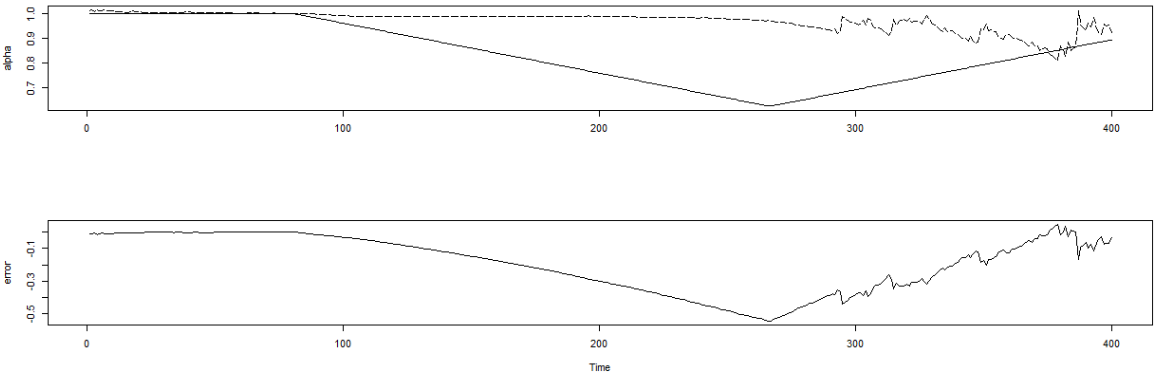

Figure 4.11 Identification and relative error of a time varying parameter identified using changepoints (β=10% and lower variance)

The figure shows that low variances and low changepoint probabilities have similar effects, so the form of the prior has been changed, in this case the prior will be the following uniform distribution

$$\theta_k^{i^*} \sim \mathcal{U}(0.8\bar{\theta}_{k-1}^i, 1.2\bar{\theta}_{k-1}^i)$$

Using this prior, the parameter identification is really good taking into account that in the case of a LPV model, the needed data is the bound of the parameter.

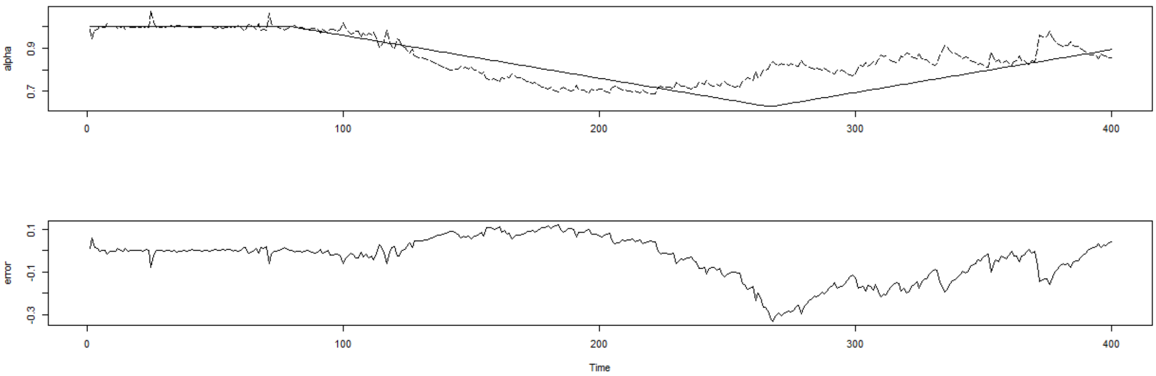

Figure 4.12 Identification and relative error of a time varying parameter identified using changepoints (β=10% and uniform prior)

Chapter 5
Conclusions and Future works

5.1. Conclusions

The model of a quadrotor is easily modelized using the physical relationships and with them it's possible to describe it as a LPV. To modelize this kind of models it's needed to find the bound of time varying parameters.

This model is nearly singular, so the methods that are included in the identification toolbox of Matlab aren't good enough to identify the system linearized in a point. This toolbox is a good tool for models that However this can be done using probabilistic methods because they are focused in identify the parameters.

In the case of a system that has a known shape, maximum likelihood shows good performances. However, the amount of data needed by this method is really big.

Another proposed method is Liu and West. This Bayesian method has really good results with a continuous parameter. However, when the parameter evolves with time, this method doesn't follow this temporal evolution.

Using a parameter learning method changepoints this could be solved; however this method produces peaks and difficult the convergence towards the right value. In order to avoid this, the selection of a good prior and an adequate changepoint probability is important.

5.2. Future works

The parameter learning method with changepoints is a method that seems promising in order to identify the model. To accomplish that a study in depth of different kind of priors and how that affects the identification process is needed.

When this can be accomplished, the next step would be to use it to identify a quadrotor and try a LPV control on it.

5.3. Contributions

The LPV model has been used in a conference [ref. 3] in which a tolerant control has been designed and also in another project in which various LPV control techniques have been compared.

Bibliography

[1] Bresciani T. (2008), “Modelling, Identification and Control of a Quadrotor Helicopter” Department of Automatic Control, Lund University.

[2] Lyung, L.(1999), “System Identification: Theory for the User”, Prentice Hall

[3] Rotondo, D. , Nejjari, F. , Torren, A. , Puig, V.(2013), “Fault Tolerant Control design for polytopic uncertain LPV systems: Application to a quadrotor”, Control and Fault-Tolerant Systems (SysTol): 643-648

[4] Shamma, J. (1992). “Gain scheduling: Potential hazards and possible remedies”, IEEE Control System Magazine 12: 101–107

[5] Shamma, J.(2012), “An Overview of LPV System” Control of Linear Parameter Varying Systems with Applications, J. Mohammadpour and C. Scherer, Springer

[6] Petris, G., Petrone, S. and Campagnoli P. (2009), “Dynamic Linear Models with R”, Springer

[7] Nemeth, C., Fearnhead, P., Mihaylova, L., Vorley D. (2012),”Particle Learning Methods for State and Parameter Estimation”, Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, 9th IET

[8] Pitt, M. and Shephard, N. (1999). “Filtering via simulation: Auxiliary particle filters”, Journal of the American Statistical Association 94: 590–599.