

Juan Gabriel Guerrero Grijalva

**CONTROL OF A QUADROTOR USING TS FUZZY
TECHNIQUES**

Dissertação submetida ao Programa de Pós-graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.
Orientador: Prof. Dr. Eugênio B. Castelan
Coorientador: Prof. Dr. Michael Klug
IFSC

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Guerrero Grijalva, Juan Gabriel
Control of a quadrotor using TS fuzzy techniques
/ Juan Gabriel Guerrero Grijalva ; orientador,
Eugênio B. Castelan; coorientador, Michael Klug -
SC, 2017.
112 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia de Automação e Sistemas,
Florianópolis, 2017.

Inclui referências.

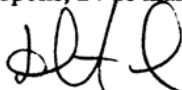
1. Engenharia de Automação e Sistemas. 2.
Robótica. 3. Controle não linear . I. Castelan,
Eugênio B. . II. Klug, Michael. III. Universidade
Federal de Santa Catarina. Programa de Pós-Graduação
em Engenharia de Automação e Sistemas. IV. Título.

Juan Gabriel Guerrero Grijalva

**CONTROL OF A QUADROTOR USING TS FUZZY
TECHNIQUES**

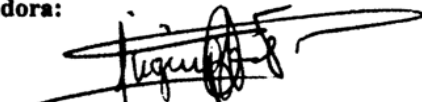
Esta Dissertação foi julgada adequada para obtenção do Título de "Mestre em Engenharia de Automação e Sistemas" e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina.

Florianópolis, 24 de maio de 2017.



Prof. Dr. Daniel Ferreira Coutinho
Coordenador do PPGEAS

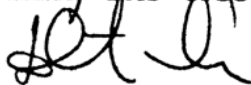
Banca Examinadora:




Prof. Dr. Eugênio B. Castelan
Presidente - DAS - UFSC



Prof. Dr. Edson Roberto de Pieri
Examinador - DAS - UFSC



Prof. Dr. Daniel Ferreira Coutinho
Examinador - DAS - UFSC



Prof. Dr. Henrique Simas
Examinador - EMC - UFSC

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Eu agradeço ao Prof. Eugênio Castelan e ao Prof. Michael Klug por me fornecer seu tempo e conhecimentos. Assim como também a SENESCYT do Equador pelo apoio econômico recebido.

”La ocasión solamente encuentra a quien esta preparado. Tiene un gran ideal: Amadlo, cultivadlo, preparaos para obtenerlo. Y tarde o temprano, si tenéis constancia y un corazón entusiasta. Dios suscitará una circunstancia, tal vez imprevista y que parecía poco probable, que hará explotar la chispa de la gran ocasión y obtendréis vuestro ideal.”

(San Juan Bosco)

RESUMO

Nos últimos anos as técnicas de controle fuzzy Takagi Sugeno (TS) têm sido utilizadas com sucesso para controlar sistemas não lineares mecatrônicos. Neste trabalho, são apresentadas algumas abordagens em tempo contínuo e discreto aplicadas em um sistema que representa um quadrotor. Teoremas baseados em desigualdades matriciais lineares que usam funções de Lyapunov são utilizados para estabilizar o sistema. Compensação distribuída paralela é a estrutura padrão de controle em malha fechada usada no presente documento. A modelagem fuzzy é baseada na utilização de regras locais não lineares que representam o sistema de um modo exato. Visando um processamento eficiente é apresentada uma modelagem fuzzy com poucas regras sem perder informação da dinâmica do sistema.

Neste trabalho é considerado o comportamento híbrido contínuo-discreto do sistema para desenvolver o algoritmo de controle que pode ser implementado em uma aplicação real. Os sensores têm a limitação fornecida pelo tempo de amostragem que é maior do que a largura de banda usada no processador, porém uma incorreta escolha do período de amostragem pode provocar processamento desnecessário ou instabilidade. O sistema pode demandar restrições na entrada de controle devido às características dos atuadores, assim como nas saídas dos estados pitch, roll, yaw e altura do sistema do quadrotor. Algumas soluções para tratar essas limitações e restrições também são apresentadas para controladores nos quais a sua lei de controle é calculada *online e offline*. As técnicas fuzzy TS com compensação distribuída paralela podem ser utilizadas com outras técnicas de controle como controle preditivo baseado no modelo ou alocação de polos. Por conseguinte também é apresentado uma comparação entre essas técnicas. Finalmente é mostrado um algoritmo genérico que pode ser embarcado em qualquer processador de fonte aberta assim como em simulações numéricas.

Palavras-chave: Modelagem fuzzy. Controle não linear. Quadrotor.

ABSTRACT

Takagi Sugeno (TS) fuzzy techniques have been plenty used successfully in the last decades to control nonlinear mechatronic systems. Therefore, many approaches, in continuous and discrete time, are presented in this work applied in a quadrotor system. Linear matrix inequalities (LMIs) theorems based in Lyapunov functions are used to stabilize the system. Parallel distributed compensation (PDC) is the standard control structure employed through all this work. The fuzzy modeling is based in local nonlinear rules which represent accurately the system. Aiming an efficient processing, it is showed a modeling with a small number of rules without losing information about the dynamic of the system.

There have been developed many researches in continuous time algorithms for quadrotors. Thus, in this document is considered the hybrid behavior, continuous and discrete-time, of the closed loops system to develop an algorithm which actually could be used in a practical implementation. The sensors are limited by the sampling time (T_s) which is greater than the bandwidth of the processor; then an incorrect selection of the T_s could led in an unnecessary processing or instability. The system could demand constraints in the control input due to the features of the actuators as well as in the output states pitch, roll, yaw and altitude of the quadrotor system. Many ways to deal with these constraints and limitations are showed for controllers based in TS fuzzy model, in which its feedback is computed online or offline.

TS fuzzy techniques with PDC are flexible to be used with other control techniques such as model predictive control and poles allocation approaches. Thus, a comparison of these techniques is presented too. Finally, a generic algorithm that could be embedded in any open source processor is presented with numerical simulations.

Keywords: TS fuzzy model. Quadrotor. Nonlinear control.

RESUMO EXPANDIDO

Neste trabalho é considerado o comportamento híbrido contínuo-discreto do sistema quadrotor para desenvolver o algoritmo de controle que pode ser implementado em uma aplicação real. As técnicas fuzzy TS com compensação distribuída paralela podem ser utilizadas com outras técnicas de controle como controle preditivo baseado no modelo ou alocação de polos. Por conseguinte também é apresentado uma comparação entre essas técnicas. Finalmente é mostrado um algoritmo genérico que pode ser embarcado em qualquer processador de fonte aberta assim como em simulações numéricas.

Objetivos

De modo geral, o principal objetivo é usar técnicas de controle baseadas em modelagem fuzzy TS para propor algoritmos de controle para um quadrotor. Como objetivos específicos em tempo contínuo tem-se desenvolver um modelo fuzzy TS com um baixo número de regras e validar esse modelo para assim depois testar alguns teoremas de estabilidade baseados em LMI para empregar controladores com estrutura baseada em compensação distribuída paralela. Em tempo discreto tem-se como objetivo específico testar alguns períodos de amostragem no cálculo dos ganhos e concluir sobre a perda de desempenho ou instabilidade que pode ocasionar. Finalmente, propor um algoritmo padrão de controle para quadrotores que pode ser embarcado em qualquer processador de fonte aberta.

Contextualização

Do ponto de vista de controle o quadrotor é um sistema altamente não linear. Portanto, uma solução relevante para controlar esse tipo de sistemas tem de ser baseado em um controle não linear. As técnicas de controle fuzzy TS tem sido provadas como eficientes ferramentas para controlar sistemas mecatrônicos. Os modelos fuzzy TS baseiam-se na utilização de um conjunto de regras fuzzy para descrever um sistema não linear em termos de submodelos lineares/afins invariantes no tempo e locais, conectados por funções de pertinência que controlam a lei de interpolação entre as regras.

Esta representação facilita, através da utilização da teoria de Lyapunov, a descrição dos problemas de controle na forma de desigualdades matriciais lineares, e portanto a obtenção de solução numérica confiável. Como consequência tem-se uma alta carga computacional. Neste contexto, o número de regras para representação do modelo TS pode

tornar o problema de projeto de controle computacionalmente intratável. Portanto, um dos desafios é conseguir um modelo com o menor número de regras possível.

Para ter a possibilidade de definir dinâmicas diferentes no sistema de altitude e atitude se apresenta uma estrutura de controle em cascada. A estratégia de controle que vai ser utilizada é compensação distribuída paralela. Que é uma técnica que já tem sido testada com sucesso em outros casos utilizando técnicas fuzzy TS. Se tem a possibilidade de usar as técnicas fuzzy junto com outras abordagens como controle preditivo, modos deslizantes, região D entre outras. Os teoremas podem ser usados online e offline, no caso online a carga computacional é muito maior.

Em tempo discreto a escolha do período de amostragem é importante para garantir estabilidade e bom desempenho. Além disso, em tempo discreto é fundamental considerar limitações e restrições nos atuadores e sensores, assim pode-se garantir sucesso em um processo de implementação prática.

Contribuições da dissertação

Dentre as contribuições da pesquisa realizada, tem-se, no Capítulo 2 é apresentado uma modelagem fuzzy TS do quadrotor com um número baixo de regras e algoritmos de controle em tempo contínuo.

No capítulo 3 obtém-se um algoritmo genérico em tempo discreto considerando limitações e restrições nos atuadores e sensores.

No capítulo 4 são apresentadas simulações dos algoritmos amostrados. Nos capítulos 3 e 4 são apresentados casos em que o período de amostragem é escolhido de uma maneira errada.

Finalmente, no capítulo 5 são apresentadas conclusões e possíveis trabalhos futuros.

Conclusão

Nesta dissertação é desenvolvido um procedimento para obter um modelo fuzzy TS de um quadrotor com um número baixo de regras o que garante um eficiente processamento. Baseado na análise em tempo discreto é apresentado um algoritmo genérico para quadrotores que garante estabilidade no qual são consideradas limitações no período de amostragem dos sensores e restrições na potência dos motores DC.

Palavras-chave: Modelagem fuzzy. Controle não linear. Quadrotor.

LISTA DE FIGURAS

Figura 1	Quadrotor configuration	28
Figura 2	Quadrotor Euler angles	29
Figura 3	Cascade scheme control	30
Figura 4	Standard quadrotor joystick	31
Figura 5	Fuzzy interface system	33
Figura 6	Scheme of control PDC	34
Figura 7	Membership function $M_1(z_1)$ and $M_2(z_1)$	44
Figura 8	Membership function $N_1(z_2)$ and $N_2(z_2)$	45
Figura 9	Membership function $S_1(z_3)$ and $S_2(z_3)$	45
Figura 10	Membership function $T_1(z_4)$ and $T_2(z_4)$	48
Figura 11	Circular D-region for poles location on S-Plane	50
Figura 12	Altitude-attitude TS-FC scheme	51
Figura 13	Quadrotor electronic devices scheme	56
Figura 14	Control structure in discrete time	57
Figura 15	Poles placement regions	65
Figura 16	Tracking performance in continuous-time	72
Figura 17	Control inputs in continuous-time	73
Figura 18	DC motors revolutions in continuous-time	74
Figura 19	Weights of the TS fuzzy model in continuous-time	75
Figura 20	Tracking performance comparison in discrete-time	77
Figura 21	Control inputs in discrete time for attitude subsystem	78
Figura 22	DC motor revolutions in discrete-time	79
Figura 23	Weights of the TS fuzzy model in discrete-time	80
Figura 24	Inappropriate sampling time test	81
Figura 25	Simulink validation scheme	99
Figura 26	Validation of ϕ , θ , ψ , x , y and z states	100
Figura 27	Augmented PI scheme	112

LISTA DE TABELAS

Tabela 1	Parameter symbols of rotational subsystem	41
Tabela 2	Parameter symbols in translational subsystem	46
Tabela 3	Quadrotor parameters values	70

LISTA DE ABREVIATURAS E SIGLAS

TS	Takagi-Sugeno
PDC	Parallel distributed compensation
RPM	Revolutions per minute
UAV	Unmanned aerial vehicles
LMI	Linear matrix inequality
FC	Fuzzy controller
LPV	Linear parameter varying
IMU	Inertial measurement unit
MPC	Model predictive control
RTG	Remote trajectory generator
DC	Direct current
DOF	Degrees of freedom
T_s	Sampling time
PI	Proportional integral
ESC	Electronic speed controller
SDP	Semidefinite programming
GPS	Global Positioning System
θ_{MR}	Maximum rate of theta
ϕ_{MR}	Maximum rate of phi
M_A	Maximum angle
PDLF	Parameter dependent Lyapunov function

LISTA DE SÍMBOLOS

\prod	N-ary product
\subset	Subset
\in	Included
\forall	For all
\Re	Set of real numbers
\mathbb{Z}	Set of integer numbers
$\Re^{n \times m}$	Set of $n \times m$ dimensional real matrices
x_i	i^{th} element of vector x
A^T	Transpose of matrix or vector A
A^{-1}	Inverse of matrix A
$diag$	Block diagonal matrix with main diagonal elements
*	Symmetric block with respect to the main diagonal of a matrix
$f(\cdot)$	Operator, set of all the possible inputs-outputs
$A > 0$	Positive definite matrix A
$A \geq 0$	Positive semi-definite matrix A
$I(0)$	Identity (zero) matrix, with appropriate dimension

SUMÁRIO

1	INTRODUCTION	25
1.1	CONTEXTUALIZATION AND BASIC CONCEPTS	26
1.1.1	Unmanned Aerial Vehicles (UAVs)	27
1.1.1.1	UAVs Classification	28
1.1.2	Quadrotors	29
1.1.2.1	Cascade control	30
1.1.3	Fuzzy modeling	31
1.1.3.1	Construction of TS fuzzy models	33
1.1.4	Parallel distributed compensation (PDC)	34
1.2	OBJECTIVES AND MOTIVATION	35
1.2.1	Specific objectives in continuous time	36
1.2.2	Specific objectives in discrete time	36
1.3	TEXT ORGANIZATION	36
2	CONTINUOUS TIME APPLICATION	39
2.1	QUADROTOR TS FUZZY MODELING	39
2.1.1	Rotational subsystem	40
2.1.2	Translational subsystem	46
2.2	STABILIZATION	48
2.2.1	Poles placement	49
2.3	CONTROLLER SYNTHESIS	51
2.3.1	Attitude controller	52
2.3.2	Altitude controller	52
3	DISCRETE TIME APPLICATION	55
3.1	QUADROTOR TS FUZZY MODELING DISCRETE-TIME	56
3.1.1	Quadrotor electronic devices	56
3.1.2	Control Structure	57
3.1.3	TS fuzzy modeling of rotational subsystem	58
3.1.4	TS fuzzy modeling of translational subsystem	60
3.2	STABILITY CONDITIONS	61
3.2.1	Optimal TS-FC	62
3.3	CONTROLLERS SYNTHESIS	64
3.3.1	Attitude control	64
3.3.2	Altitude control	65
3.3.3	Generic algorithm	66
4	SIMULATION AND DISCUSSION	69
4.1	SYSTEM PARAMETERS DESCRIPTION	69
4.2	STEPS TO APPLY A TS-FC WITH PDC	70

4.3	CONTINUOUS TIME SIMULATION	71
4.4	DISCRETE TIME SIMULATION	76
5	CONCLUSION	83
5.1	CONCLUSIONS ABOUT THE CONTINUOUS TIME APPLICATION	83
5.2	CONCLUSIONS ABOUT THE DISCRETE TIME APPLICATION	83
5.3	PERSPECTIVES	84
	REFERENCES	85
	ANNEX A – Summary - modeling of a quadrotor	91
	ANNEX B – Validation of the TS modeling.....	99
	ANNEX C – Simulink schemes	103
	ANNEX D – Augmented PI dynamic in state equation	111

1 INTRODUCTION

There have been rapid changes and improvements in the fields of electronics, computer and control systems over the last decades. As a result of this, computer controlled systems have been increasing in almost every field. The number of only mechanically operating systems is very few. Therefore, the design, production and maintenance of advanced products are no longer a single subject. This made Mechatronics emerge as a new discipline in the 70's. Unmanned aerial vehicle (UAV) quadrotor is a good example of a mechatronic system due to the mixture of mechanical elements (where is considered the aerodynamics, strength of materials, mechanisms design, etc) and electronic elements, analogical or digital, such as the sensors, actuators and the processor where is embedded the control algorithm. From the perspective of control this under-actuated system has highly nonlinear behavior being a good challenge to develop and test a nonlinear control algorithm.

There are many modern control solutions developed and used to obtain better performances to control nonlinear mechatronic systems such as: model predictive control (CAMACHO; ALBA, 2013), fuzzy control (PRECUP et al., 2008), and adaptive control (BLAŽIČ; ŠKRJANC; MATKO, 2003), among others. It is shown in Cairano et al. (2007) that these methods must include the model of the nonlinear plant, and the possibility to estimate the parameters and the state variables. In Ho e Chou (2007) is presented an optimal controller design based on TS fuzzy models, and the state variables are expressed as orthogonal functions. In Klug (2015) is showed a generalized nonlinear fuzzy (or, N-fuzzy) model that defines the classical TS fuzzy model like a special case of N-fuzzy model.

TS fuzzy controller (TS-FC) is flexible to operating point changes. The linear submodels used for the TS-FC design can be treated as LPV system allowing the application of well-established Lyapunov and LMI based tools for LPV control systems. A key issue when applying a TS fuzzy representation for control purposes is the model accuracy, since TS models can exactly or approximately represent the original nonlinear system to be controlled. Even though the exact TS fuzzy representation has identical dynamics to the original nonlinear system, but the convexity of the model can only be guaranteed in a specific domain of the state space (KLUG, 2015).

In the last years, stability analysis and controllers in discrete time have been developed in Klug, Castelan e Coutinho (2015), Klug et al. (2014, 2015) being powerful tools to be considered in the moment of deal with hybrid continuous-discrete nonlinear systems. In Klug (2010) was shown that a small number of rules led in a soft processing than a large. Accordingly, a goal of this research is to develop a fuzzy model with a smaller number of rules than other papers related in the literature such as Lee e Kim (2014), Yacef et al. (2012).

1.1 CONTEXTUALIZATION AND BASIC CONCEPTS

UAV quadrotors are widely used nowadays in many applications such as in security and surveillance tasks, production of videos for different purposes, by general public like a hobby and by researchers to implement new technologies and ideas over these devices. From a control system perspective, a quadrotor is inherently a nonlinear and unstable dynamical system that needs some feedback controllers to carry through different tasks as hovering and tracking references, among others. Thus, many linear and nonlinear controllers have been proposed in the literature aiming at improving the stability and performance of quadrotors in different situations (LEE; KIM, 2014; CHOI; AHN, 2015; CISNEROS et al., 2016).

Among the nonlinear techniques used for synthetizing the control laws, the so-called parallel distributed compensation (PDC) (WANG; TANAKA; GRIFFIN, 1996) technique based on the use of TS fuzzy models has been shown to be attractive both from the practical and theoretical points of view. The PDC setting offers a simple and natural procedure to handle the nonlinear control systems with the possibility to engage the basic knowledge of linear systems and to use tools from the robust and LPV control (KLUG, 2015). For example, effective solutions in continuous-time to track references are presented in Yacef et al. (2012) and Lee e Kim (2014). A drawback in the references cited before is that they use a high number of rules in the TS fuzzy model. However, to implement these control laws in digital processors, it is necessary to discretize them. Thus, if the discretization technique and/or the sampling time are not chosen correctly, the closed-loop performance can deteriorate, up to making the system unstable, or lead to excessive data processing. In special, the sampling rates of the ultrasonic sensor

(used to measure the altitude of the quadrotor) and the IMU (used to measure the attitude of the quadrotor), are considered in the present work as critical points due to the capture speed they are featured with. The available actuators such as the DC motors are bounded with a top speed too. It could be necessary that the user wants or needs to constraint the output states like the inclination or the speed of the system. Such kind of practical constraints and limitations should be considered in the design process to effectively implement a control law. TS fuzzy model in continuous time can be easily handled using a common Lyapunov function (TANAKA; WANG, 2004) or an extended approach considering D-region to allocate arbitrary the poles (HONG; NAM, 2003) in an circle on the S-Plane. Otherwise, in discrete-time using a contractive coefficient allocating the poles arbitrary on the Z-Plane (KLUG, 2010) as well as optimal control such as TS fuzzy MPC (FENG, 2010) optimizing the worst case of an infinite horizon cost function.

In quadrotors applications, remote trajectory generators (RTG) are used to emit the references to the plant. A personal computer is employed in Gautam e Ha (2013) to track positional references in X, Y and Z axis. On the other hand, numerous people such as cameramen, athletes and landscape video recording amateurs use a joystick as interface, and in this case pitch, roll, yaw angles and altitude are the states to be reached. Thus, the effectiveness of linear parameter varying (LPV) controller has been tested in Cisneros et al. (2016) for high speed trajectory tracking. In Torres et al. (2016), it is presented an attitude controller in continuous-time considering the ultrasonic sensor and the I.M.U through eight local submodels. Small number of submodels and rules in fuzzy modeling causes a lighter processing than a large (KLUG, 2015). Thus, to obtain less submodels without losing information about the dynamics of the system we apply a simpler modeling than the one exposed in Torres et al. (2016), the sector nonlinearity approach (OHTAKE; TANAKA; WANG, 2003) based on weighting and membership functions. Also, in contrast to Yacef et al. (2012) and Lee e Kim (2014), we assume the employment of joystick without positional control.

1.1.1 Unmanned Aerial Vehicles (UAVs)

For our purposes, UAVs are defined as small aircrafts that are flown without pilot. These vehicles can either be remotely operated by a human with a joystick or they can be autonomous controlled by an

onboard computer. UAVs are mainly used in military applications, but in the last years they are being deployed in civil applications too, such as journalism recording videos, building security and people's hobbies, etc (AZZAM; WANG, 2010).

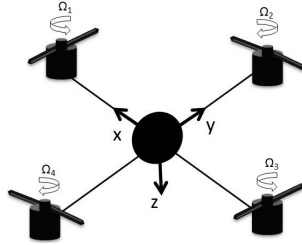


Figura 1 – Quadrotor configuration

1.1.1.1 UAVs Classification

UAVs can be classified into four main categories based on their aerodynamic configuration as follows (CARRILLO et al., 2012):

1. **Fixed-wing UAVs:** require a run-way to take-off and land. They can fly for a long time and at high cruising speeds. They are mainly used in scientific applications such as meteorological reconnaissance and environmental monitoring.
2. **Rotary-wing UAVs:** they can take off and land vertically. They can also hover and fly with high maneuverability. The Rotary-wing UAVs can be further classified into four groups:
 - (a) **Single-rotor:** they have a main rotor on top and another rotor at the tail for stability, same like the helicopter configuration.
 - (b) **Coaxial:** they have two rotors rotating in opposite directions mounted to the same shaft.
 - (c) **Quadrotor:** they have four rotors fitted in a cross-like configuration.
 - (d) **Multi-rotor:** UAVs with six or eight rotors. They are agile type and fly even when a motor fails, as there is redundancy due to the large number of rotors.

3. **Blimps UAVs:** which may look like balloons or airships, they ensure lifting by their helium-filled body. They are very light and have a large size. They can fly for a long time and at low speeds.
4. **Flapping-wing UAVs:** they are inspired from birds and flying insects. These UAVs have small wings and have an extremely low payload and endurance. On the other hand, they have low power consumption and can perform vertical take-off and landing. This class of UAVs is still under development.

1.1.2 Quadrotors

The quadrotors concept has more than a century. It was reported that the Breguet-Richet quadrotor built in 1907 had actually flown. A quadrotor mainly consists of four rotor, each located in one end of a cross-like structure as shown in Figure 1 Each rotor consists of a propeller fitted to a separately powered DC motor. Propellers 1 and 3 rotate in the same direction while propellers 2 and 4 rotate in an opposite direction leading to balancing the total system torque and cancelling the gyroscopic and aerodynamics torques in stationary flights. The quadrotor is a 6 DOF device, thus 6 variables are used

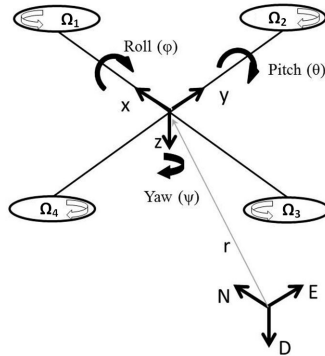


Figura 2 – Quadrotor Euler angles

to express its position in space (x, y, z, ϕ, θ and ψ). x, y and z represent the distances of the quadrotor center of mass along the X, Y and Z axis respectively from an Earth fixed inertial frame. The symbols ϕ, θ and ψ represent the three Euler angles representing the orientation of the quadrotor. The roll angle is represented by ϕ which is the angle

about the X -axis, θ is the pitch angle about the Y -axis, while ψ is the yaw angle about the Z -axis as depicted in Figure 2. The roll and pitch angles are usually called the attitude of the quadrotor, while the yaw angle is referred to as the heading of the quadrotor. Aiming to get a better organization in this work we include the heading in the attitude control. For the linear motion, the distance from the ground is referred as the altitude and the x and y position in space is often called the position of the quadrotor.

1.1.2.1 Cascade control

To deal with this underactuated system (fewer number of actuators than degrees of freedom) a cascade control scheme is used. In Figure 3 is depicted how the rotational subsystem affects to the translational subsystem. Also, there exists an inner loop that control the euler angles ϕ, θ and ψ , this is called attitude controller. The external loop control the position in the X and Y axis, this one is called positional control. Instead of this, a joystick could be used to determinate the position of the quadrotor. There also exist an independent loop, which is in charge to control the Z position of the quadrotor, this is called altitude controller.

In Figure 3 the references could be predefined through a onboard positional control or a remote control joystick. If it is chosen a positional control, there are many ways to implement it, for example, to move the UAV from one point to other could be defined an arbitrary flying route or an optimized algorithm to determinate the shortest route between the points. On the other hand, if a joystick is chosen, the arbitrary references will be given via remote control to the quadrotor.

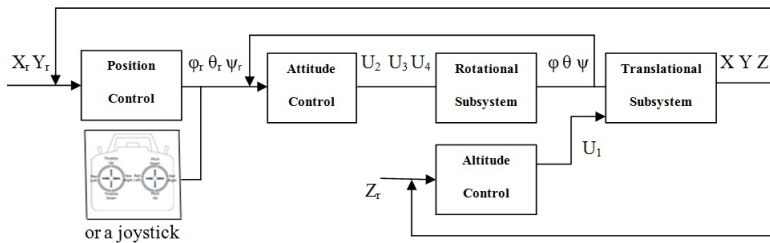


Figura 3 – Cascade scheme control

In the first case the yaw angle is avoided in the attitude control, but

in the second case the yaw angle is one of the buttons of the joystick as depicted in Figure 4. In the most of actual applications people use quadrotors with a joystick, as consequence this work will focus on the nonlinear control of a quadrotor using a remote control. The standards

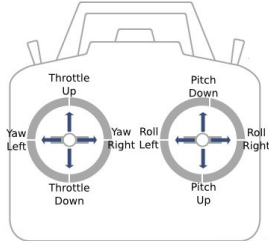


Figura 4 – Standard quadrotor joystick

buttons are:

1. THROTTLE (up/down): This button manages the altitude of the quadrotor $Z - axis$.
2. YAW(right/left): Controls the heading of the quadrotor, the yaw angle.
3. PITCH(up/down): In charge of the forward and backward motion, so the pitch angle.
4. ROLL(right/left): It sends orders to obtain right and left moves influencing over the roll angle.

1.1.3 Fuzzy modeling

It is not new the idea of multi-model approach (BINDER et al., 1981), but the idea of fuzzy modeling using the concept of the fuzzy set theory proposes a new technique to build multi-models of the process based on the input-output data or the original mathematical model of the system, besides of a clearly understandable linear combination that deals the fuzzy model.

Thus, Takagi and Sugeno (TAKAGI; SUGENO, 1985) proposed a fuzzy model described by fuzzy IF-THEN rules which represents local input-output relations of a nonlinear system. The main skill in this

technique is to express the local dynamics of each fuzzy implication or rule by a linear system model. Fuzzy blending of the linear systems sub-models, relay the overall fuzzy model of the complete nonlinear system.

This technique employs fuzzy rules, which are *IF-THEN* statements implicating fuzzy sets, fuzzy logic and fuzzy inference. Fuzzy rules play an overriding role in performing expert modeling knowledge and experience in linking the input variables of fuzzy controllers to output variables. There are two main kinds of fuzzy rules, namely, Mandami and TS fuzzy rules. An example of a Mandami rule describing the motion of a quadrotor is:

IF *Rate* is *Big* **THEN** *Slope* is *Modest*,

where *Rate* is an input variable and *Slope* through pitching or rolling is an output variable, *Big* and *Modest* are fuzzy sets, the first one is called input fuzzy set while the last one is named the output fuzzy set. The variables as well as linguistic terms, such as "*Big*", can be represented by mathematical symbols. Thus, a Mamdani fuzzy rule for a fuzzy controller of a quadrotor can be described as follows:

IF x is M **THEN** y is N ,

where x is an input variable, for example the rate or speed of a quadrotor, y an output variable, for example a rotor speed, and M and N fuzzy sets. The part of the statement before "THEN" is called rule antecedent and the other part rule consequent. Now, lets funnel the TS fuzzy rules, unlike Mamdani fuzzy rules, TS rules use functions of input variables as the rule consequent. For fuzzy control, a TS rule corresponding to the Mamdani rule of the last example is:

IF x is M **THEN** $y = f(x)$,

where $f(\cdot)$ is a real function of any type. Fuzzy inference or rule-based systems are schematically shown in Figure 5. They are composed of five conventional block: a rule base containing a number of fuzzy IF-THEN rules, a database which defines the membership functions of the fuzzy sets used in the fuzzy rules, a decision-making unit which performs the inference operations on the rules, a fuzzification interface which transform the crisp inputs into degrees of match with linguistic values and a defuzzification interface which transform the fuzzy results of the inference into a crisp output.

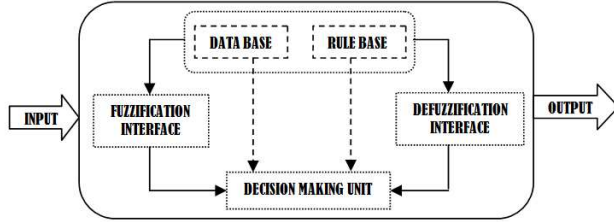


Figura 5 – Fuzzy interface system

1.1.3.1 Construction of TS fuzzy models

There are two main methods to construct TS fuzzy models (Klug, 2010, 2015): approximated and exact. In this work will be used the second one, the first one is possible to be applied though. Lets represent a nonlinear plant by:

$$\begin{aligned} \dot{x} &= \mathcal{A}(x)x + \mathcal{B}(x)u, \\ y &= Cx, \end{aligned} \quad (1.1)$$

with $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$ and $y \in \mathcal{Y} \subset \mathbb{R}^{n_y}$. The functions $\mathcal{A}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$, with $\mathcal{A}(0) = 0$, $\mathcal{B}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x \times n_u}$ are continuous and bounded for all $x \in \mathcal{X}$. \mathcal{X} is defined as a region that belongs to the state space domain containing the origin and $C \in \mathbb{R}^{n_y \times n_x}$ is a matrix with constant values. The TS fuzzy model for (1.1) is:

$$R_i \quad : \quad \begin{cases} \text{IF} & v_{(1)} \text{ is } M_1^i, v_{(2)} \text{ is } M_2^i, \dots, v_{(n_s)} \text{ is } M_{n_s}^i \\ \text{THEN} & \dot{x} = A_i x + B_i u \\ & y = Cx \end{cases} \quad (1.2)$$

where R_1, \dots, R_{n_r} are fuzzy rules, $v := [v_{(1)}, v_{(2)}, \dots, v_{(n_s)}]$ represent the premise variables, $M_j^i, j = 1, \dots, n_s$, represent the fuzzy sets and A_i, B_i perform the matrices that define the fuzzy local sub-models.

Let consider $\mu_j^i(v_{(j)})$ as the weights of the fuzzy sets M_j^i associated to the premise variable $v_{(j)}$, and $\omega^i(v) = \prod_{j=1}^{n_s} \mu_j^i(v_{(j)})$. Assuming

$$\mu_j^i(v_{(j)}) \geq 0, \text{ it implies } \omega^i(v) \geq 0, \forall \quad i = 1, \dots, n_r \text{ and } \sum_{i=1}^{n_r} \omega^i(v) > 0.$$

The normalized weight of each rule h_i referred to as the membership function of i^{th} local sub-model, provide:

$$h_{(i)} = h(v_{(i)}) = \frac{\omega^i(v)}{\sum_{i=1}^{n_r} \omega^i(v)}, \quad \forall \quad i = 1, \dots, n_r,$$

Thus, we have the TS fuzzy model:

$$\dot{x} = A(h)x + B(h)u \quad (1.3)$$

with:

$$[A(h) \quad B(h)] = \sum_{i=1}^{n_r} h_i [A_i \quad B_i]$$

Note that (1.3) can be viewed as an LPV system with a polytopic structure.

1.1.4 Parallel distributed compensation (PDC)

Once the TS fuzzy model is obtained, a parallel distributed compensation can be applied, which offers a procedure to design a fuzzy controller (see Figure 6). In the PDC design, each control rule is de-

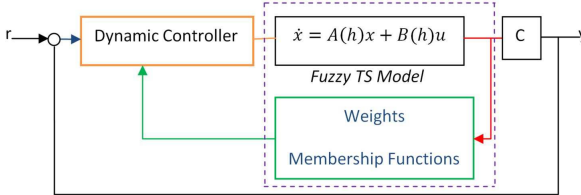


Figura 6 – Scheme of control PDC

signed from the corresponding rule of a TS fuzzy model. For instance, the control rules can be established as follows:

$$CR_i : \begin{cases} IF & v_{(1)} \text{ is } M_1^i, \quad v_{(2)} \text{ is } M_2^i, \dots, v_{(n_s)} \text{ is } M_{n_s}^i \\ THEN & u = -K_i x \end{cases} \quad (1.4)$$

where CR_i represents each fuzzy control rule. These rules have a linear controller (state feedback control laws in this case) in the consequent

parts. The overall fuzzy controller is represented by:

$$u = -\frac{\sum_{i=1}^{n_r} \omega^i(v) K_i x}{\sum_{i=1}^{n_r} \omega^i(v)} = -\sum_{i=1}^{n_r} h_i(v) K_i x, \quad (1.5)$$

where:

$$\sum_{i=1}^{n_r} h_i(v) [K_i] = K(h)$$

With PDC it is possible to have a simple and natural procedure to handle the nonlinear control systems. Though, the possibility to engage the basic knowledge of linear systems, the opposite happens in other nonlinear control techniques which require special and rather involved knowledge. Therefore, the problems of stability, poles location and other performance requirements can be reduced to LMI issues. Optimal control, H_2 , H_∞ , sliding modes approaches can be handled using PDC (FENG, 2010) too.

To obtain the values of the feedback, it is necessary to solve LMIs theorems that warrants at least stability in our system. Then, in this work will be used semidefinite programming (SDP). This programming is a sub-field of convex optimization, that is with respect to a proper convex cone. As this work mainly focus on implementation issues, it will not be argued topics about convexity or quadratic cone programming.

1.2 OBJECTIVES AND MOTIVATION

This research mainly focus on some implementation issues of TS-FC techniques in UAV quadrotors. In continuous time, it is expected to use the effective solutions presented in Tanaka e Wang (2004), Hong e Nam (2003). Also, an efficient TS fuzzy modeling which can be used as in continuous as in discrete time approaches. In discrete time is implemented the techniques shown in Klug (2010) and the TS fuzzy MPC presented in Feng (2010). To excel other related works existent in the literature, the next specific objectives are defined:

1.2.1 Specific objectives in continuous time

1. In spite of using a full dynamic model of the plant, considering the employment of large angle references and aerodynamic effects, obtain a TS fuzzy model with the lowest quantity of rules as possible. As consequence a soft processing carried by the processor.
2. Validate the TS fuzzy model.
3. Simulate and compare the techniques without considering the digital parts involved in the system and conclude about the contributions of each technique.

1.2.2 Specific objectives in discrete time

1. Based on the digital elements, set and test appropriate/inappropriate sampling time for both inner and outer control loop in the cascade control structure.
2. Considering the actual hybrid behavior of the system, so including digital parts of the quadrotor, propose online and offline generic algorithms that could be embedded in any open source processor.
3. Through simulation, compare the performance obtained from the approaches. Conclude about the contributions of each case.

1.3 TEXT ORGANIZATION

This document is organized in five chapters including the introduction which covers some basic concepts about fuzzy modeling, PDC, cascade structure and UAV quadrotors. Also, the motivation and objectives.

Chapter 2 describes a continuous time solutions, common Lyapunov function and poles allocation by D-region, to synthesize PDC controllers and detailed explanation about the TS fuzzy modeling employed along the document.

Chapter 3 presents a discrete time approach to obtain a controller for the system, this time considering limitations and constraints offered by the actuator and sensors. In the last part is described an

online-offline generic algorithm which can be embedded in a processor.

Chapter 4 embeds the simulation of the solutions presented in chapter 3 and 4. Furthermore, a comparison and discussion about the performance obtained in all the cases.

Finally, *Chapter 5* presents conclusions, remarks, suggestions and overtures to possible future works. The annexes include some additional information which complement the understanding of the preceding chapters.

2 CONTINUOUS TIME APPLICATION

This chapter is highlighted with a description of the TS fuzzy modeling of a quadrotor by using the so-called sector nonlinearity approach (TANAKA; WANG, 2004). The fuzzy model in this case will represent exactly the original model. The main challenge in this point is to find a TS fuzzy model with few rules in spite of using a complete dynamic system. In practical implementations the IMU and the ultrasonic sensor measure all the needed states to compute the controller. So, it is not necessary to develop an estimator.

The two theorems employed to calculate the feedback control law are based on the use of common Lyapunov functions being one of them complemented with a D-region condition for regional poles placement. Therefore, we have the possibility to set some desired time performance in that case. As consequence, the LMI theorem with D-region condition demands more processing (KLUG, 2015), to compute the gains, than the one without D-region condition. On the other hand, the calculation of gains in both cases is performed offline. Besides, the TS fuzzy modeling can be expressed as LPV systems which may simplify the explanation and interpretation of the controller synthesis results.

2.1 QUADROTOR TS FUZZY MODELING

There are several works, see: (MAHONY; KUMAR; CORKE, 2012; PATEL; PATEL; VYAS, 2012; GAITAN; BOLEA, 2013), describing step by step the dynamic modeling of quadrotors. Therefore, we omit this procedure. However, a brief summary of the modeling is presented in Annex A. The dynamic system expressed by states equations used in this paper was taken from Yacef e Boudjema (2011). The choice of this model was done under the fact that it is valid for large angle variations. Also, it includes aerodynamic effects, such as air friction, making this model closer to the actual plant. The states $\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, \mathcal{Z}$ and $\dot{\mathcal{Z}}$ are represented for $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ and x_8 respectively. Axis orientations X, Y, Z and rotor speed Ω_i of each motor are based in the scheme of Figure 1 and 2. Because of the cascade control strategy, the system was divided in two subsystems.

2.1.1 Rotational subsystem

Defining $x_a = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T$ and for the control inputs $U_a = [U_2 \ U_3 \ U_4]^T$, the rotational subsystems is expressed by the following state equation:

$$\dot{x}_a = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -e_1 & 0 & a_1 x_6 - a_2 \Omega_r & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_3 x_6 + a_4 \Omega_r & 0 & -e_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & a_5 x_4 & 0 & 0 & 0 & -e_3 \end{bmatrix} x_a + \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix} U_a \quad (2.1)$$

where:

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}, \quad a_2 = \frac{Jr}{I_{xx}}, \quad a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}}, \quad a_4 = \frac{Jr}{I_{yy}}, \quad b_1 = \frac{L}{I_{xx}}, \quad b_2 = \frac{L}{I_{yy}},$$

$$b_3 = \frac{L}{I_{zz}}, \quad a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}}, \quad e_1 = \frac{\mathcal{K}_{ax}}{I_{xx}}, \quad e_2 = \frac{\mathcal{K}_{ay}}{I_{yy}} \quad \text{and} \quad e_3 = \frac{\mathcal{K}_{az}}{I_{zz}}.$$

Description of each symbol used in (2.1) is depicted in Table 1 where $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ due to a torque compensation. Of course, we can assume any range for x_4, x_6 and Ω_r to construct a fuzzy model. However, we assume $x_4 \in [-0.5, 0.5]rad/s$, $x_6 \in [-0.5, 0.5]rad/s$ and $\Omega_r \in [4, -4]rad/s$. We selected these values under the fact that are actual values for the model of quadrotor we are using. The premise variables in (2.1) are defined as $z_1 = x_4$, $z_2 = x_6$ and $z_3 = \Omega_r$. Then, we have:

$$\dot{x}_a = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -e_1 & 0 & a_1 z_2 - a_2 z_3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_3 z_2 + a_4 z_3 & 0 & -e_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & a_5 z_1 & 0 & 0 & 0 & -e_3 \end{bmatrix} x_a + \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix} U_a \quad (2.2)$$

Next, we calculate the minimum and maximum values of z_1, z_2 and

Tabela 1 – Parameter symbols of rotational subsystem

Symbol	Parameter
L	Quadrotor axis length
I_{xx}, I_{yy} and I_{zz}	Inertia in X, Y and Z axis
$\mathcal{J}r$	Rotor inertia
$\mathcal{K}_{ax}, \mathcal{K}_{ay}$, and \mathcal{K}_{az}	Aerodynamic coefficients
$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$	Rotor relative speed
U_2, U_3 and U_4	Control inputs

z_3 under $x_4 \in [-0.5, 0.5]$, $x_6 \in [-0.5, 0.5]$ and $\Omega_r \in [4, -4]$, as follows:

$$\begin{aligned}
 \max_{x_4, x_6, \Omega_r} z_1 &= 0.5 = f_1, & \min_{x_4, x_6, \Omega_r} z_1 &= -0.5 = f_2, \\
 \max_{x_4, x_6, \Omega_r} z_2 &= 0.5 = p_1, & \min_{x_4, x_6, \Omega_r} z_2 &= -0.5 = p_2, \\
 \max_{x_4, x_6, \Omega_r} z_3 &= 4 = q_1, & \min_{x_4, x_6, \Omega_r} z_3 &= -4 = q_2,
 \end{aligned} \tag{2.3}$$

From (2.3), z_1 , z_2 and z_3 can be rewritten as:

$$\begin{aligned}
 z_1 = x_4 &= M_1(z_1) \cdot (0.5) + M_2(z_1) \cdot (-0.5), \\
 z_2 = x_6 &= N_1(z_2) \cdot (0.5) + N_2(z_2) \cdot (-0.5), \\
 z_3 = \Omega_r &= S_1(z_3) \cdot (4) + S_2(z_3) \cdot (-4),
 \end{aligned} \tag{2.4}$$

where:

$$M_1(z_1) + M_2(z_1) = 1,$$

$$N_1(z_2) + N_2(z_2) = 1,$$

$$S_1(z_3) + S_2(z_3) = 1,$$

Therefore, the membership functions can be calculated as:

$$\begin{aligned}
 M_1 &= \frac{z_1 + 0.5}{1}, & M_2 &= \frac{0.5 - z_1}{1}, \\
 N_1 &= \frac{z_2 + 0.5}{1}, & N_2 &= \frac{0.5 - z_2}{1}, \\
 S_1 &= \frac{z_3 + 4}{8}, & S_2 &= \frac{4 - z_3}{8},
 \end{aligned} \tag{2.5}$$

Illustrations of the trapezoidal membership functions are depicted in Figures 7, 8 and 9. The expressions in (2.4) can be rewritten as:

$$\begin{aligned}
 z_1 &= \sum_{i=1}^2 M_i(z_1) f_i, \\
 z_2 &= \sum_{i=1}^2 N_i(z_2) p_i, \\
 z_3 &= \sum_{i=1}^2 S_i(z_3) q_i,
 \end{aligned} \tag{2.6}$$

From (2.6) we construct the following TS fuzzy model for the rotational equations of the quadrotor:

$$\dot{x}_a = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 N_i(z_2) M_j(z_1) S_k(z_3) \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -e_1 & 0 & a_1 p_i - a_2 q_k & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_3 p_i + a_4 q_k & 0 & -e_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & a_5 f_j & 0 & 0 & 0 & -e_3 \end{bmatrix}}_{A_{a_{ijk}}} x_a + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix}}_{B_a} U_a. \tag{2.7}$$

Employing the relation $n_r = 2^{N_n}$ (KLUG, 2015), being n_r and N_n the number of rules and number of nonlinearities respectively, we calculate eight rules in our fuzzy model. We name the membership functions in this case "Positive", "Negative", "Positive", "Negative", "Above" and "Below", respectively. Then, the nonlinear system (2.2) is represented by the following fuzzy rules:

Model Rule 1:

IF z_1 is "Positive" and z_2 is "Positive" and z_3 is "Above"
THEN $\dot{x}_a = A_{a_1} x_a + B_a U_a$.

Model Rule 2:

IF z_1 is "Positive" and z_2 is "Negative" and z_3 is "Above"
THEN $\dot{x}_a = A_{a2}x_a + B_aU_a$.

Model Rule 3:

IF z_1 is "Negative" and z_2 is "Positive" and z_3 is "Above"
THEN $\dot{x}_a = A_{a3}x_a + B_aU_a$.

Model Rule 4:

IF z_1 is "Negative" and z_2 is "Negative" and z_3 is "Above"
THEN $\dot{x}_a = A_{a4}x_a + B_aU_a$.

Model Rule 5:

IF z_1 is "Positive" and z_2 is "Positive" and z_3 is "Below"
THEN $\dot{x}_a = A_{a5}x_a + B_aU_a$.

Model Rule 6:

IF z_1 is "Positive" and z_2 is "Negative" and z_3 is "Below"
THEN $\dot{x}_a = A_{a6}x_a + B_aU_a$.

Model Rule 7:

IF z_1 is "Negative" and z_2 is "Positive" and z_3 is "Below"
THEN $\dot{x}_a = A_{a7}x_a + B_aU_a$.

Model Rule 8:

IF z_1 is "Negative" and z_2 is "Negative" and z_3 is "Below"
THEN $\dot{x}_a = A_{a8}x_a + B_aU_a$.

So, (2.7) is rewritten as:

$$\dot{x}_a = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 N_i(z_2) M_j(z_1) S_k(z_3) (A_{a_{ijk}}x_a + B_aU_a) \quad (2.8)$$

The defuzzification is carried out as:

$$\dot{x}_a = \sum_{i=1}^8 h_i(z) (A_{ai}x_a + B_a U_a), \quad (2.9)$$

where:

$$\begin{aligned} h_1(z) &= N_1(z_2)M_1(z_1)S_1(z_3), \\ h_2(z) &= N_1(z_2)M_1(z_1)S_2(z_3), \\ h_3(z) &= N_1(z_2)M_2(z_1)S_1(z_3), \\ h_4(z) &= N_1(z_2)M_2(z_1)S_2(z_3), \\ h_5(z) &= N_2(z_2)M_1(z_1)S_1(z_3), \\ h_6(z) &= N_2(z_2)M_1(z_1)S_2(z_3), \\ h_7(z) &= N_2(z_2)M_2(z_1)S_1(z_3), \\ h_8(z) &= N_2(z_2)M_2(z_1)S_2(z_3), \end{aligned}$$

The corresponding LPV representation of (2.9) is given by:

$$\dot{x}_a = A_a(h)x_a + B_a U_a \quad (2.10)$$

In (2.10) $A_a(h) = \sum_{i=1}^{n_r} h_i(A_{ai})$.

The fuzzy model exactly represents the nonlinear rotational system of the quadrotor in the region $[-0.5, 0.5] \times [-0.5, 0.5] \times [-4, 4]$ on the x_4 , x_6 and Ω_r space.

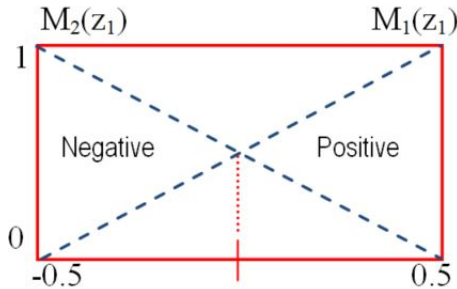


Figure 7 – Membership function $M_1(z_1)$ and $M_2(z_1)$

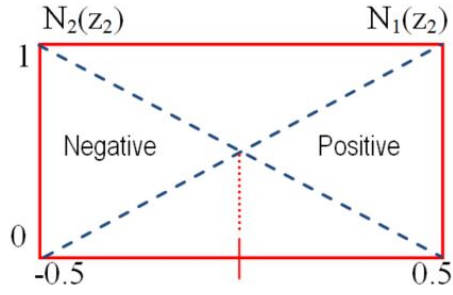


Figure 8 – Membership function $N_1(z_2)$ and $N_2(z_2)$

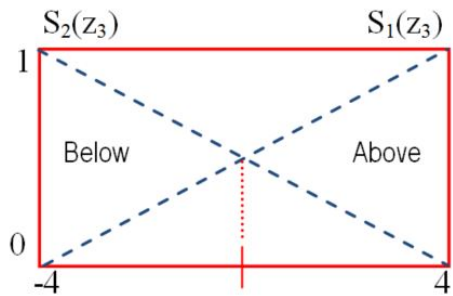


Figure 9 – Membership function $S_1(z_3)$ and $S_2(z_3)$

2.1.2 Translational subsystem

In the subsection 1.1.2.1 was shown the control strategy through a cascade control scheme. In Figure 3 we can see that if it is desired to control the altitude of the quadrotor, it is only necessary to rule the position in $Z - axis$.

Thus, as in the last subsection, from Yacef et al. (2012) is taken the part that corresponds to the states \mathcal{Z} and $\dot{\mathcal{Z}}$. Now, defining $x_b = [x_7 \ x_8]^T$ and $e_6 = \frac{\mathcal{K}_{rz}}{m_q}$, we have:

$$\dot{x}_b = \begin{bmatrix} 0 & 1 \\ 0 & e_6 \end{bmatrix} x_b + \begin{bmatrix} 0 \\ -\frac{\cos x_1 \cos x_3}{m_q} \end{bmatrix} U_1 + \underbrace{\begin{bmatrix} 0 \\ g_r \end{bmatrix}}_G \quad (2.11)$$

The symbols employed in (2.11) are detailed in Table 2:

Tabela 2 – Parameter symbols in translational subsystem

Symbol	Parameter
m_q	Mass of the quadrotor
g_r	Gravity
\mathcal{K}_{rz}	Aerodynamic coefficient
U_1	Control input

Both the states x_1 and x_3 belong to the interval $[-\pi/3, \pi/3]rad$. For the nonlinear terms in (2.11), define $z_4 = \cos x_1 \cos x_3$. Then, it is obtained:

$$\dot{x}_b = \begin{bmatrix} 0 & 1 \\ 0 & e_6 \end{bmatrix} x_b + \begin{bmatrix} 0 \\ -\frac{z_4}{m_q} \end{bmatrix} U_1 + \begin{bmatrix} 0 \\ g_r \end{bmatrix} \quad (2.12)$$

After, we calculate the minimum and maximum values of z_4 under x_1 and $x_3 \in [-\pi/3, \pi/3]$ as follows:

$$\max_{\cos x_1 \cos x_3} z_4 = 1 = d_1, \quad \min_{\cos x_1 \cos x_3} z_4 = \cos^2\left(\frac{\pi}{3}\right) = d_2,$$

Now, using the maximum and minimum values z_4 can be represented by:

$$z_4 = \cos x_1 \cos x_3 = T_1(z_4) \cdot (1) + T_2(z_4) \cdot (\cos^2(\frac{\pi}{3})), \quad (2.13)$$

where:

$$T_1(z_4) + T_2(z_4) = 1$$

The expression in (2.13) is rewritten as:

$$z_4 = \sum_{i=1}^2 T_i(z_4) d_i \quad (2.14)$$

Now from (2.12) we construct the TS fuzzy model for the altitude dynamic in the quadrotor system as follows:

$$\dot{x}_b = \sum_{i=1}^2 T_i \left(\underbrace{\begin{bmatrix} 0 & 1 \\ 0 & e_6 \end{bmatrix}}_{A_b} x_b + \underbrace{\begin{bmatrix} 0 \\ d_i \end{bmatrix}}_{B_{bi}} U_1 + \underbrace{\begin{bmatrix} 0 \\ g_r \end{bmatrix}}_G \right) \quad (2.15)$$

Thus, the membership functions can be calculated as:

$$T_1 = \frac{z_4 + 1}{1 - \cos^2(\frac{\pi}{3})}, \quad T_2 = \frac{-(z_4 + \cos^2(\frac{\pi}{3}))}{1 - \cos^2(\frac{\pi}{3})},$$

In this case the membership functions are called "Maximum" and "Minimum". Then, the nonlinear system in (2.12) is represented by the next fuzzy rules:

Model Rule 1:

$$\begin{aligned} &\mathbf{IF} z_4 \text{ is "Maximum"} \\ &\mathbf{THEN} \dot{x}_b = A_b x_b + B_{b1} U_1 + G. \end{aligned}$$

Model Rule 2:

$$\begin{aligned} &\mathbf{IF} z_4 \text{ is "Minimum"} \\ &\mathbf{THEN} \dot{x}_b = A_b x_b + B_{b2} U_1 + G. \end{aligned}$$

Thus. (2.15) can be expressed as:

$$\dot{x}_b = \sum_{i=1}^2 T_i (A_b x_b + B_{bi} U_1 + G), \quad (2.16)$$

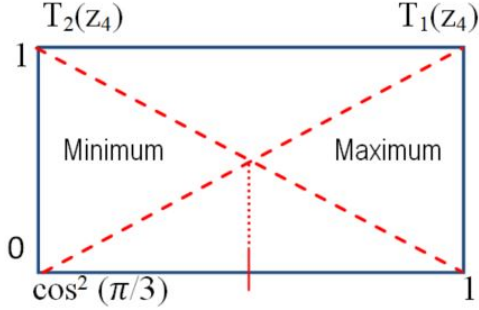


Figure 10 – Membership function $T_1(z_4)$ and $T_2(z_4)$

The defuzzification is carried out as:

$$\dot{x}_b = \sum_{i=1}^2 \bar{h}_i(z) (A_b x_b + B_{b_i} U_1 + G), \quad (2.17)$$

where:

$$\begin{aligned} \bar{h}_1(z) &= T_1(z_4), \\ \bar{h}_2(z) &= T_2(z_4), \end{aligned}$$

The corresponding LPV representation of (2.17) is given by:

$$\dot{x}_b = A_b x_b + B_b(\bar{h}) U_1 + G \quad (2.18)$$

In (2.18) $B_b(\bar{h}) = \sum_{i=1}^{n_r} \bar{h}_i(B_{b_i})$.

The fuzzy model exactly represents the nonlinear altitude system of the quadrotor in the region $[-\pi/3, \pi/3]$ on the z_4 space under convexity conditions.

2.2 STABILIZATION

Once we have a fuzzy model as (1.3), a sufficient quadratic stability condition derived by Tanaka and Sugeno (TANAKA; SUGENO, 1992) for ensuring stability is given as follows:

Theorem 1 (TANAKA; WANG, 2004): *The TS fuzzy model in closed loop (1.3) is quadratically stable for any feedback K_i (1.5), via PDC structure, if exists a common positive definite matrix P such that:*

$$\begin{aligned}
 &G_{ii}^T P + P G_{ii} < 0 \\
 &\forall i = 1, 2, \dots, n_r. \\
 &\left(\frac{G_{ij} + G_{ji}}{2} \right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2} \right) \leq 0 \\
 &\forall i, j = 1, 2, \dots, n_r \quad \text{and} \quad i < j. \quad (2.19)
 \end{aligned}$$

where $G_{ij} = A_i - B_i K_j$.

The conditions in (2.19) are LMIs in P when K_i is predefined. Nonetheless, our objective is to get stabilization of the system, then, design the gain matrix K_i such that the conditions in (2.19) are fulfilled. So, K_i is not a predetermined matrix, but decision variable. This quadratic stability problem could be turn in an LMI feasibility problem using a linear transformation $\mathcal{M}_i = K_i \mathbb{X}$ with $\mathbb{X} = P^{-1}$. Besides, (2.19) is rewritten as an LMI problem in \mathcal{M}_i and \mathbb{X} :

$$\begin{aligned}
 &\mathbb{X} > 0 \\
 &-\mathbb{X} A_i^T - A_i \mathbb{X} + \mathcal{M}_i^T B_i^T + B_i \mathcal{M}_i > 0 \\
 &\quad -\mathbb{X} A_i^T - A_i \mathbb{X} - \mathbb{X} A_j^T - A_j \mathbb{X} \\
 &+\mathcal{M}_j^T B_i^T + B_i \mathcal{M}_j + \mathcal{M}_i^T B_j^T + B_j \mathcal{M}_i \geq 0 \\
 &\forall i, j = 1, 2, \dots, n_r \quad \text{and} \quad i < j. \quad (2.20)
 \end{aligned}$$

More detailed information about these results can be found in Tanaka e Wang (2004).

2.2.1 Poles placement

In the synthesis of controllers in addition to the stability of the system, other features in the performance are required by the users. The stability conditions (2.20) do not guarantee a desired behavior in the transient response of the states. On the other hand, a satisfactory

transient response can be acquired by confining its poles in a prescribed region. Thereby, in this section is discussed a pole clustering region in terms of LMI. For this reason, we introduce the following stability representation based in LMI:

Theorem 2 (HONG; NAM, 2003): *The closed loop TS fuzzy model (1.3) is D-stable, it means all the poles lying in a D-region, in this case the circle represented in Figure 11, for some feedback K_i if there exists a definite positive matrix \mathbb{X} such that the next condition is satisfied:*

$$\begin{pmatrix} -r\mathbb{X} & q\mathbb{X} + \mathbb{X}(A_i - B_i K_j)^T \\ (q\mathbb{X} + (A_i - B_i K_j)\mathbb{X}) & -r\mathbb{X} \end{pmatrix} < 0 \quad (2.21)$$

The LMI condition (2.21) is not convex, then, it is performed a change

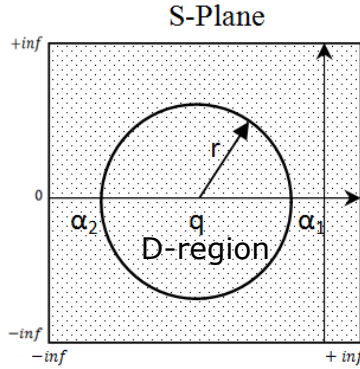


Figura 11 – Circular D-region for poles location on S-Plane

of variables $\mathcal{M}_i = K_i \mathbb{X}$. Now, (2.21) turns in a convex LMI feasibility problem in \mathcal{M}_i and \mathbb{X} as follows:

$$\begin{pmatrix} -r\mathbb{X} & q\mathbb{X} + \mathbb{X}A_i^T - \mathcal{M}_i^T B_i^T \\ (q\mathbb{X} + A_i \mathbb{X} - B_i \mathcal{M}_i) & -r\mathbb{X} \end{pmatrix} < 0 \quad (2.22)$$

$i=1,2,\dots,n_r.$

By combining Theorems 1 and 2, we propose to use the next theorem that fulfill both the requirements of stability and desired transient response:

Theorem 3 (HONG; NAM, 2003): *The TS fuzzy model (1.3) in closed loop is stabilizable in the specified circular D-region, as shown in Figure 11, if exists a common positive symmetric matrix \mathbb{X} and \mathcal{M}_i holding the next LMI conditions:*

$$\begin{aligned}
& \mathbb{X} > 0 \\
& -\mathbb{X}A_i^T - A_i\mathbb{X} + \mathcal{M}_i^T B_i^T + B_i\mathcal{M}_i > 0 \\
& -\mathbb{X}A_i^T - A_i\mathbb{X} - \mathbb{X}A_j^T - A_j\mathbb{X} \\
& + \mathcal{M}_j^T B_i^T + B_i\mathcal{M}_j + \mathcal{M}_i^T B_j^T + B_j\mathcal{M}_i \geq 0 \\
& \begin{pmatrix} -rX & qX + XA_i^T - M_i^T B_i^T \\ qX + A_i X - B_i M_i & -rX \end{pmatrix} < 0 \\
& \forall i, j = 1, 2, \dots, n_r \quad \text{and} \quad i < j. \quad (2.23)
\end{aligned}$$

where $K_i = \mathcal{M}_i \mathbb{X}^{-1}$.

Now the resulting controller offers stability and the desired transient response performance, if the problem is feasible, simultaneously.

In Figure 11 the interval to locate the poles is between $\alpha_1 = q - r$ and $\alpha_2 = q + r$. In the next chapter is developed an analysis in discrete time. Then, the values of α_1 and α_2 can be related with the stability conditions on a Z-plane.

2.3 CONTROLLER SYNTHESIS

According to the cascade control structure showed in Figure 3 and considering the PDC (see Figure 6), it is posed two controllers: altitude and attitude, as depicted in Figure 12.

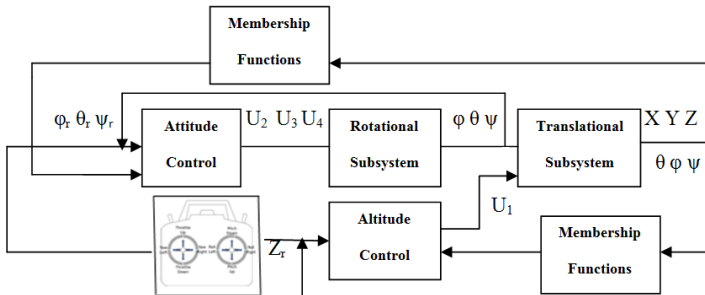


Figura 12 – Altitude-attitude TS-FC scheme

2.3.1 Attitude controller

The attitude TS-FC using PDC is given by:

$$U_a = - \sum_{l=1}^{n_r} h_l(z) K_{al} x_a = K_a(h) x_a \quad (2.24)$$

Note that the controller in (2.24) is nonlinear. Substituting (2.24) in the rotational fuzzy model (2.8), we obtain:

$$\dot{x}_a = \sum_{l=1}^{n_r} h_l(z) (A_{al} - B_a K_{al}) x_a \quad (2.25)$$

where $l = k + 2(j - 1) + 4(i - 1)$. If there exists a feasible solution in Theorems 1 and 3, the closed loop system (2.25) is asymptotically stable. Notice that (2.25) can be expressed as a LPV system as follows:

$$\dot{x}_a = (A_a(h) + B_a K_a(h)) x_a$$

2.3.2 Altitude controller

TS-FC with PDC does not guarantee zero steady state error. To sort out this inconvenient is proposed an augmented PI dynamic, through an integral over the error $\xi = \int e(t) dt$, in the loop of the PDC related to the altitude controller, as follows:

$$U_1 = - \sum_{i=1}^n \bar{h}_i(z) (K_{bi} x_b - F_i \xi - m_q g_r) = -K_b(\bar{h}) + F(\bar{h}) \xi + m_q g_r \quad (2.26)$$

In (2.26), $m_q g_r$ is used to compensate the gravitational force. Substituting (2.26) in the open loop altitude fuzzy model (2.15) we have:

$$\dot{\tilde{x}}_b = \sum_{i=1}^{n_r} \bar{h}_i(z) (\tilde{A}_b - \tilde{B}_{bi} \tilde{K}_{bi}) \tilde{x}_b \quad (2.27)$$

where:

$$\tilde{x}_b = \begin{bmatrix} x_b \\ \xi \end{bmatrix}, \quad \tilde{A}_b = \begin{bmatrix} A_b & 0 \\ C_b & 0 \end{bmatrix}, \quad \tilde{B}_{bi} = \begin{bmatrix} B_{bi} \\ 0 \end{bmatrix} \text{ and } \tilde{K}_{bi} = \begin{bmatrix} K_{bi} & -F_i \end{bmatrix}.$$

The closed loop TS fuzzy system (2.27), if there exists feasible

solution, is robust asymptotically stable and guarantees zero steady state error. Note that (2.27) can be rewritten as a LPV system:

$$\dot{\tilde{x}}_b = \left(\tilde{A}_b(\bar{h}) + \tilde{B}_b \tilde{K}_b(\bar{h}) \right) \tilde{x}_b \quad (2.28)$$

The results presented in this chapter will be tested through simulation in Chapter 4. Thus, it is going to be possible to conclude about the advantages and drawbacks in each case.

3 DISCRETE TIME APPLICATION

In this chapter is presented controllers considering the hybrid, continuous and discrete-time, behavior of the closed-loop system. As consequence, we have the appearance of some limitations and constraints in the control system. After analyzing the actuators and sensors involved in the system, it was identified as the main limitation the T_s of the buzzer or ultrasonic sensor and the IMU. The bandwidth of these sensors are too big compared to the other electronic parts of the system. The power of the DC motors is an important constraint to be considered too. The size and weight of the motors depend of the power they are featured with. For this reason we consider in our analysis this control input constraint that affects directly to the altitude subsystem.

The TS fuzzy modeling applied in this chapter is the same one described in the last chapter. One of the objectives of this work is to elaborate a generic algorithm with an efficient processing. Aiming to achieve this objective we present a strategy to reduce the number of rules in the rotational subsystem from eight to four.

The T_s in this chapter depends of the specifications of the components employed. We assume the use of IMU and ultrasonic sensor. There are cases in actual applications where the attitude and altitude states are taken from other electronic components. For example, there is the possibility to use a global positional system (GPS) device, replacing the IMU and the ultrasonic sensor (ORDAZ et al., 2013). In that case the procedure is the same, the T_s of each loop of the control structure depicted in Figure 14 should be chosen according to the specifications of the GPS device. For this reason we call "generic" to the algorithm presented in this chapter because, independently of the electronic parts employed, the algorithm can be used anyway.

There is the possibility to use algorithms in which the gains are computed online and offline. In both cases the T_s should be selected correctly. Otherwise, there is the possibility to turn the system unstable or lose performance. An example of this case is shown in the last part of the next chapter. Finally, we present a generic algorithm for online and offline controllers. This algorithm guarantees stability, desired performance and efficient processing using TS fuzzy model based techniques.

3.1 QUADROTOR TS FUZZY MODELING DISCRETE-TIME

In this section is presented a TS fuzzy model in discrete time considering the sensors limitation. Thereby, it is important to detail the electronic devices of the quadrotor system. Thus, propose a control structure based in the features of the digital components. In the last chapter was shown step by step the way to get a TS fuzzy model, so in this sections is summarized that procedure.

3.1.1 Quadrotor electronic devices

Based on the Figure 13 the on board electronic devices are detailed as follows:

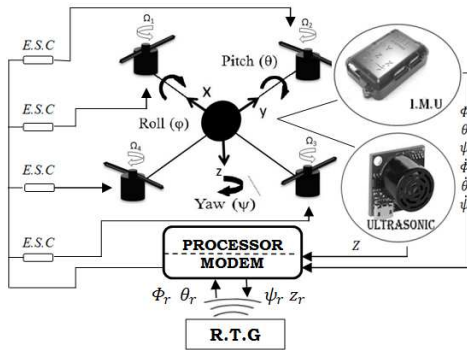


Figura 13 – Quadrotor electronic devices scheme

- *Modem*: Through wireless network, it receives the references from the RTG, next it sends this information to the processor. It is assumed that there is not loss of information in the controller analysis developed in this chapter.
- *Processor*: The algorithm developed is embedded in this part, it could be called the brain of the system because it is in charge of receive the information from the sensors, then it sends orders to the actuators. In our case it is assumed that the processor is enough powerful to compute the information on time.
- *IMU*: This micro electro-mechanical system provides to the pro-

cessor information about the euler angles (ϕ, θ, ψ) and their rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. The readings of quadrotor attitude from this sensor are faster than the readings of quadrotor altitude from the ultrasonic sensor.

- *Ultrasonic sensor*: The altitude of the quadrotor is measured by this element.
- *Electronic speed controller (ESC)*: This part interacts between the processor and the actuators demodulating- conditioning the signals. It is assumed that the dynamic of this component is fast enough according to the system. Therefore, it is considered as static gain.

The critical rates between these devices are in the sensors which operates in a slow rate. Accordingly, these parameters are considered as our starting point.

3.1.2 Control Structure

We dispose the structure of control in Figure 14. It is possible

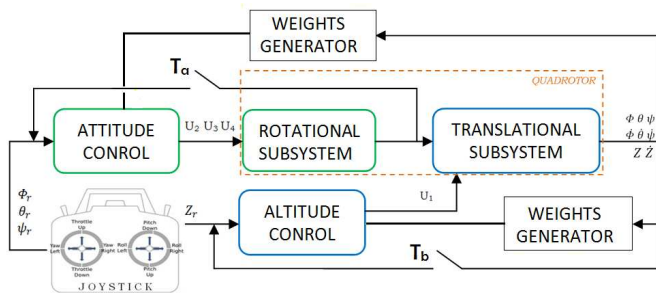


Figure 14 – Control structure in discrete time

to identify two control loops. The control loop that corresponds to the attitude controller is in charge of the euler angles, operating with a sampling time (T_a) defined according to the bandwidth of the IMU. The same analysis is done with the altitude control loop, the ultrasonic sensor reading frequency determines the sampling time (T_b) to be used in. This control structure allows to work singly each control loop. In contrast to (YACEF et al., 2012), this control loop can set a dynamic in the attitude controller faster than the one of the altitude controller.

It could be interpreted like an extra degree of freedom to design our control algorithm. The controller is designed under the scheme depicted in Figure 13. The references ϕ_r , θ_r , ψ_r and \mathcal{Z}_r are generated from the RTG which could be for example a joystick.

3.1.3 TS fuzzy modeling of rotational subsystem

Lets start presenting a discretized aproximation based in Euler-forward method of the continuous time rotational subsystem (2.1):

$$x_a(k+1) = \begin{bmatrix} 1 & T_a & 0 & 0 & 0 & 0 \\ 0 & (1 - e_1 T_a) & 0 & 0 & 0 & a_1 x_4 T_a \\ 0 & 0 & 1 & T_a & 0 & 0 \\ 0 & 0 & 0 & (1 - e_2 T_a) & 0 & a_3 x_2 T_a \\ 0 & 0 & 0 & 0 & 1 & T_a \\ 0 & 0 & 0 & a_5 x_2 T_a & 0 & (1 - e_3 T_a) \end{bmatrix} x_a(k) + \begin{bmatrix} 0 & 0 & 0 \\ b_1 T_a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 T_a & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 T_a \end{bmatrix} U_a(k) + \begin{bmatrix} 0 \\ -a_2 x_4 T_a \\ 0 \\ a_4 x_2 T_a \\ 0 \\ 0 \end{bmatrix} \Omega_r(k) \quad (3.1)$$

In the above equation, differently from the continuous time approach, the variable $\Omega_r(h)$ is considered as an external input that can be adequately compensated by the control law, allowing to reduce the number of rules with respect to the continuous time TS fuzzy model. It is defined the premise variables of (3.1) as $z_1 = x_2(k)$ and $z_2 = x_4(k)$. The maximum rates of ϕ and θ are considered equal to ϕ_{MR} and θ_{MR} respectively. Because of the symmetry of quadrotor the minimum values are expressed as the negative maximum values. Now, the maximum and minimum values of z_1 and z_2 under $x_2(k) \in [-\phi_{MR}, \phi_{MR}]$, and $x_4(k) \in [-\theta_{MR}, \theta_{MR}]$ are calculated as follows:

$$\begin{aligned} \max_{x_2, x_4} z_1(k) &= \phi_{MR} = q_1, & \min_{x_2, x_4} z_1(k) &= -\phi_{MR} = q_2 \\ \max_{x_2, x_4} z_2(k) &= \theta_{MR} = f_1, & \min_{x_2, x_4} z_2(k) &= -\theta_{MR} = f_2 \end{aligned} \quad (3.2)$$

The membership functions can be calculated as:

$$\begin{aligned} V_1 &= \frac{z_1(k) - q_2}{q_1 - q_2}, & V_2 &= \frac{q_1 - z_1(k)}{q_1 - q_2} \\ W_1 &= \frac{z_2(k) - f_2}{f_1 - f_2}, & W_2 &= \frac{f_1 - z_2(k)}{f_1 - f_2} \end{aligned} \quad (3.3)$$

From (3.1) is obtained the fuzzy model by using (3.2) and (3.3):

$$\begin{aligned} x_a(k+1) &= \sum_{j=1}^2 \sum_{t=1}^2 V_t(z_1(k)) W_j(z_2(k)) \\ &\underbrace{\begin{bmatrix} 1 & T_a & 0 & 0 & 0 & 0 \\ 0 & (1 - e_1 T_a) & 0 & 0 & 0 & a_1 f_j T_a \\ 0 & 0 & 1 & T_a & 0 & 0 \\ 0 & 0 & 0 & (1 - e_2 T_a) & 0 & a_3 q_t T_a \\ 0 & 0 & 0 & 0 & 1 & T_a \\ 0 & 0 & 0 & a_5 q_t T_a & 0 & (1 - e_3 T_a) \end{bmatrix}}_{A_{ai}} x_a(k) + \\ &\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ b_1 T_a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_2 T_a & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_3 T_a \end{bmatrix}}_{B_a} U_a(k) + \underbrace{\begin{bmatrix} 0 \\ -a_2 f_j T_a \\ 0 \\ a_4 q_t T_a \\ 0 \\ 0 \end{bmatrix}}_{G_{ai}} \Omega_r(k) \end{aligned} \quad (3.4)$$

The defuzzification of (3.4) is carried out as:

$$x_a(k+1) = \sum_{i=1}^4 h_i(z(k)) (A_{ai} x_a(k) + B_a U_a(k) + G_{ai} \Omega_r(k))$$

where:

$$\begin{aligned} h_1(z(k)) &= V_1(z_1(k)) W_1(z_2(k)) \\ h_2(z(k)) &= V_1(z_1(k)) W_2(z_2(k)) \\ h_3(z(k)) &= V_2(z_1(k)) W_1(z_2(k)) \\ h_4(z(k)) &= V_2(z_1(k)) W_2(z_2(k)) \end{aligned}$$

Notice that $i = j + 2(t - 1)$. The fuzzy model (3.4) represents exactly the rotational equations of motion of the quadrotor in the region $[-\phi_{MR}, \phi_{MR}] \times [-\theta_{MR}, \theta_{MR}]$ on the $x_2(k)$ and $x_4(k)$ space respectively under convexity conditions.

3.1.4 TS fuzzy modeling of translational subsystem

Similar to the last subsection, firstly is presented a discretized approximation based in Euler-forward method of the continuous time rotational subsystem (2.11):

$$x_b(k+1) = \begin{bmatrix} 1 & T_b \\ 0 & (1 + T_b e_6) \end{bmatrix} x_b(k) + \begin{bmatrix} 0 \\ -\frac{T_b(\cos x_1 \cos x_3)}{m_q} \end{bmatrix} U_1(k) + \begin{bmatrix} 0 \\ T_b g_r \end{bmatrix} \quad (3.5)$$

Proceeding as it was done in the continuous-time case, the maximum angle of slope under the fuzzy model exactly represents the dynamic of the plant is M_A . The modeling is done under $x_1(k), x_3(k) \in [M_A, -M_A]$. We define the premise $z_3 = \cos x_1(k) \cos x_3(k)$. After, we calculate the minimum and maximum values of z_3 :

$$\max_{\cos x_1 \cos x_3} z_3(k) = 1 = d_1, \quad \min_{\cos x_1 \cos x_3} z_3(k) = \cos^2(M_A) = d_2 \quad (3.6)$$

With the next membership function:

$$J_1 = \frac{z_3(k) - d_2}{d_1 - d_2}, \quad J_2 = \frac{d_1 - z_3(k)}{d_1 - d_2} \quad (3.7)$$

Employing (3.6) and (3.7) the fuzzy model of (3.5) is:

$$x_b(k+1) = \sum_{i=1}^2 J_i \left\{ \underbrace{\begin{bmatrix} 1 & T_b \\ 0 & (1 + T_b e_6) \end{bmatrix}}_{A_b} x_b(k) + \underbrace{\begin{bmatrix} 0 \\ -\frac{T_b d_i}{m_q} \end{bmatrix}}_{B_{bi}} U_1(k) + \underbrace{\begin{bmatrix} 0 \\ T_b g_r \end{bmatrix}}_G \right\} \quad (3.8)$$

The defuzzification of (3.8) is carried out as:

$$x_b(k+1) = \sum_{i=1}^2 \bar{h}_i(z) (A_b x_b + B_{b_i} U_1 + G)$$

where:

$$\bar{h}_1(z(k)) = J_1(z_3(k)) \quad \bar{h}_2(z(k)) = J_2(z_3(k))$$

The fuzzy model (3.8) represents exactly the translational equations of motion of the quadrotor in the region $[-M_A, M_A]$ on the z_3 space under convexity conditions.

3.2 STABILITY CONDITIONS

To determinate both stability and desired transient response, it is used a parameter dependent Lyapunov function (PDLF) $V(x, h)$ as follows:

Definition 1: Considering a positive scalar $\lambda \in (0, 1]$. A closed loop TS fuzzy model in discrete time $\mathbb{A}_C = \{A(h_k) + B(h_k)K(h_k)\} x_k$ is robust asymptotically stable if:

$$\Delta V_\lambda(x_k, h_k) \triangleq V(x_{k+1}, h_{k+1}) - \lambda V(x_k, h_k) < 0 \quad (3.9)$$

$$\forall x_k \in \mathfrak{R}^n.$$

Theorem 4 (KLUG, 2015): *For a given a quadratic function $V = x_k^T Q^{-1}(h_k) x_k$, a TS fuzzy model \mathbb{A}_C in closed loop is asymptotically stable verifying (3.9) if the LMI condition in (3.10) is acquired.*

$$\begin{bmatrix} -Q(h_{k+1}) & \mathbb{A}_C \\ * & -\lambda Q^{-1}(h_k) \end{bmatrix} < 0 \quad (3.10)$$

The stability condition in (3.10) is a LMI in Q . Being our objective to look a feedback K_i we apply a linear transformation $Y_i = K_i \mathbb{U}$ with $\mathbb{U} = Q^{-1}$. Then, we have to find feasible solutions to the following set of LMIs:

$$\begin{bmatrix} -Q_q & \frac{(A_i + A_j)\mathbb{U} + B_i Y_j + B_j Y_i}{\lambda^{-1} \left(\frac{Q_i + Q_j}{2} \right) - \mathbb{U} - \mathbb{U}^T} \\ * & \end{bmatrix} < 0 \quad (3.11)$$

$$\forall q, i = 1, \dots, n_r \quad \text{and} \quad \forall j = i, \dots, n_r.$$

The conditions in (3.11) can be used to synthesize a controller with PDC through the relation $K_i = Y_i U^{-1}$.

More detailed information about these results can be found in Klug (2015).

3.2.1 Optimal TS-FC

The TS fuzzy model in (1.3) can be rewritten in predictive form as:

$$x_{k+\ell+1|k} = A_{h(k+\ell)|h(k)} x_{k+\ell|k} + B_{h(k+\ell)|h(k)} u_{k+\ell|k} \quad (3.12)$$

where $x_{k+\ell|k}$, $\ell \geq 1$ is the predicted state at time $(k + \ell)$, computed at instant k . Thus the current state x_k , in this case is denoted as $x_{k|k}$. The objective of model predictive control is to minimize the worst-case infinite horizon quadratic objective function:

$$\min_{u(k|k), u(k+\ell|k)} \max_{A(k+\ell|k)B(k+\ell|k)} J_0^\infty(k) \quad (3.13)$$

with:

$$\begin{aligned} J_0^\infty(k) &= \sum_{\ell=0}^{\infty} [x_{k+\ell|k}^T Q x_{k+\ell|k} + u_{k+\ell|k}^T R u_{k+\ell|k}] \\ &= x_{k|k}^T Q x_{k|k} + u_{k|k}^T R u_{k|k} + J_1^\infty(k) \end{aligned} \quad (3.14)$$

being $Q > 0$ and $R > 0$ symmetric positive weighting functions. In (3.13) at each time k the states control law $u_{k+\ell|k} = K_{h(k)} x_{k+\ell|k}$ is used to minimize the worst case value of $J_0^\infty(k)$. At each time k is defined a quadratic function $V(x_{k+\ell|k}) = x_{k+\ell|k}^T P_k x_{k+\ell|k}$ where P_k is a definite positive matrix at time k . For any $A_{h(k+\ell|k)}$ and $B_{h(k+\ell|k)}$, $V(x_k)$ satisfies the following stability condition:

$$V(x_{k+\ell+1|k}) - V(x_{k+\ell|k}) \leq -[x_{k+\ell|k}^T Q x_{k+\ell|k} + u_{k+\ell|k}^T R u_{k+\ell|k}]. \quad (3.15)$$

Summing in (3.15) from $\ell = 0$ to infinite and assuming that $V(x_{\infty|k}) = 0$, it follows that:

$$\max_{A(k+l|k)B(k+l|k)} J_1^\infty(k) \leq V(x_{k+1|k}) = x_{k+1|k}^T P_k x_{k+1|k}. \quad (3.16)$$

An upper bound for $J_0^\infty(k)$ can be derived from (3.16) and (3.14) as follows:

$$J_0^\infty(k) \leq x_{k|k}^T Q x_{k|k} + u_{k|k}^T R u_{k|k} + x_{k+1|k}^T P_k x_{k+1|k}. \quad (3.17)$$

Then, its minimization is equivalent to:

$$\min_{\gamma, u(k|k), P_k} \gamma$$

subject to:

$$\begin{aligned} & x_{k|k}^T Q x_{k|k} + u_{k|k}^T R u_{k|k} \\ & + [A_{h(k)|h(k)} x_{k|k} + B_{h(k)|h(k)} u_{k|k}]^T P_k [A_{h(k)|h(k)} x_{k|k} + B_{h(k)|h(k)} u_{k|k}] \leq \gamma \end{aligned} \quad (3.18)$$

Using Schur complement and defining $\mathbb{X}_k = \gamma P_k^{-1} > 0$, the conditions (3.18) and (3.16) can be expressed through LMIs by:

$$\begin{aligned} & \min_{\gamma, u, X, Y} \gamma \\ & \begin{bmatrix} 1 & * & * & * \\ A_{k|k} x(k|k) + B_{k|k} u(k|k) & \mathbb{X}_k & * & * \\ Q^{1/2} x(k|k) & 0 & \gamma I & * \\ R^{1/2} u(k|k) & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \\ & \begin{bmatrix} \mathbb{X}_k & * & * & * \\ A_j \mathbb{X}_k + B_j Y_j & \mathbb{X}_k & * & * \\ Q^{1/2} \mathbb{X}_k & 0 & \gamma I & * \\ R^{1/2} Y_j & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \quad (3.19) \\ & \forall j = 1, \dots, n_s. \\ & \begin{bmatrix} \mathbb{X}_k & * & * \\ \frac{1}{2} [A_j \mathbb{X}_k + A_l \mathbb{X}_k + B_j Y_l + B_l Y_j] & \mathbb{X}_k & * \\ Q^{1/2} \mathbb{X}_k & 0 & \gamma I \end{bmatrix} \geq 0 \\ & 1 \leq j \leq l \leq n_r. \end{aligned}$$

Theorem 5 (FENG, 2010): *For the open loop system $A(h_k)x_k + B(h_k)u_k$, the control law $u_{k+\ell|k} = K_\ell x_{k+\ell|k}$, $\ell \geq 1$ minimize the worst case of (3.14) if there exist a definite positive matrix $\mathbb{X}_k > 0$ and Y_k such that the solution to the conditions in (3.19) are feasible. Then, the resulting closed loop system $(A(h_k) + B(h_k)K(h_k))x_k$, where $A(h_k)$ and $B(h_k)$*

are known matrices with gains $K_i = Y_{ki}\mathbb{X}_k^{-1}$, is robust asymptotically stable.

More information about Theorem 5 can be found in (FENG, 2010).

Remark 3.1: If it is considered the receding horizon, the calculations of the gains K_i through LMIs are performed online. Otherwise, the gains can be computed offline, then this algorithm could be easily embedded in any open source processor. According to Feng (2010) in the first case it is possible to consider constraints over the control input adding (3.20) to Theorem 5:

$$\begin{bmatrix} \mathbb{U} & Y_j(k) \\ Y_j(k)^T & X(k) \end{bmatrix} \geq 0, \quad (3.20)$$

$$\mathbb{U}_{rr} \leq u_{r,max}^2 \quad \text{and} \quad |u_r(k|k)| \leq u_{r,max}$$

In (3.20), \mathbb{U}_{rr} is the diagonal element of the decision matrix \mathbb{U} . More detailed information about these results can be found in Feng (2010).

3.3 CONTROLLERS SYNTHESIS

In this section is shown the PDC control law for the altitude and attitude systems considering limitations and constraints in the actuators and sensors. The feedback gains can be computed using Theorem 4, offline theorem based in poles placement according to Figure 15, or Theorem 5, online theorem based in weighting functions and MPC. In both cases the control law is the same. In discrete time systems, it is important to find efficient solutions in the controller. Thus, is presented an option to reduce the number of rules in the attitude controller occasioning an efficient data processing.

Figure 15 illustrates with dashed lines the region in which Theorem 4 locates the poles through the relation $\lambda = e^{-\alpha T_{a,b}}$, being $-\alpha$ the position on the real axis in the S-plane and $T_{a,b}$ the sampling time.

3.3.1 Attitude control

The attitude controller is defined as follows:

$$U_a(h) = [K_a(h)x_a(k) + \Gamma(h)\Omega_r(k)] \quad (3.21)$$

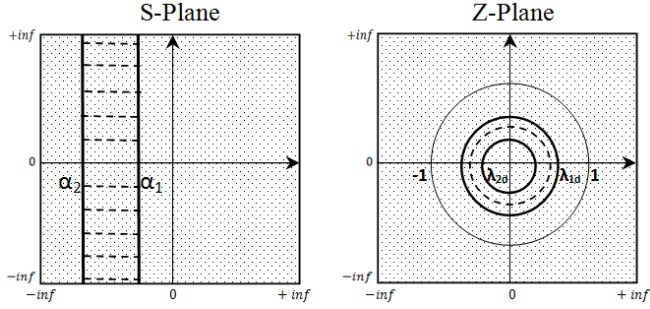


Figura 15 – Poles placement regions

with:

$$\Gamma(h) = \begin{bmatrix} \frac{a_2 f(h)}{b_1} & -\frac{a_4 q(h)}{b_2} & 0 \end{bmatrix}^T$$

In (3.21), $\Gamma(h)$ compensates dynamically the rotor relative speed (Ω_r). Thereby, replacing (3.21) in (3.4) in closed loop we have:

$$x_a(k+1) = [A_a(h(k)) + B_a K_a(h(k))] x_a(k) \quad (3.22)$$

Note that now the number of rules in the TS fuzzy model is reduced from eight to four. Theorems 4 and 5 can be used to compute the fuzzy gains $K_a(h_k)$ such that (3.22) is robust asymptotically stable.

3.3.2 Altitude control

Similar to continuous time, in this case to avoid steady state error, we add an integral action: $\xi(k) = \xi(k+1) - y(k)$ in the PDC. Accordingly, the control law is defined as:

$$U_1(k) = [K_b(\bar{h})x_b(k) - F(\bar{h})\xi(k) - m_q g_r] \quad (3.23)$$

So, the augmented altitude system in closed loop is:

$$\tilde{x}_b(k+1) = [\tilde{A}_b + \tilde{B}_b(\bar{h}(k))\tilde{K}(\bar{h}(k))]\tilde{x}_b(k) \quad (3.24)$$

with:

$$\tilde{x}_b = \begin{bmatrix} x_b \\ \xi \end{bmatrix}, \tilde{A}_b = \begin{bmatrix} A_b & 0 \\ C & I \end{bmatrix}, \tilde{B}_b = \begin{bmatrix} B_b \\ 0 \end{bmatrix}, \tilde{K} = [K_b \quad -F]$$

Now, the system (3.24) provides zero steady state error. Moreover, the system is robust to uncertainties in the model of the plant.

3.3.3 Generic algorithm

Taking advantage of the analysis done in this section, we propose a generic algorithm to embed in the quadrotor processor. The stabilizing control laws, obtained under the conditions predefined guarantee stability with an efficient processing. Aiming to reduce the number of rows in the writing of this algorithm, it is used an indented code structure.

To complement algorithm 1, it is possible to use (3.25) (see Annex A)

Algorithm 1 Generic algorithm structure

```

1: procedure WEIGHT GENERATOR(Attitude Control)
2:   for  $t = 1$  to 2 do
3:     for  $j = 1$  to 2 do
4:        $\ell = j + 2(t - 1)$ ;
5:        $aux1_{[\ell]} = V_{[t]}W_{[j]}$ ;
6:        $aux2 = 0$ ;
7:       for  $\ell = 1$  to 4 do
8:          $aux2 = aux2 + aux1_{[\ell]}$ ;
9:        $iaux2 = 1/aux2$ ;
10:      for  $\ell = 1$  to 4 do
11:         $h_{[\ell]} = aux1_{[\ell]}iaux2$ ;
12: procedure CONTROL LAW(Attitude Control)
13:   Using theorem 5 calculate  $K_a$ 
14:    $U_a = 0$ ;
15:   for  $i = 1$  to 4 do
16:      $aux3_{[i]} = h_{[i]}K_{a[i]}x_a$ ;
17:      $aux4_{[i]} = h_{[i]}\Gamma_{[i]}\Omega_r$ ;
18:      $U_a = U_a + aux3_{[i]} + aux4_{[i]}$ ;
19: procedure WEIGHT GENERATOR(Altitude Control)
20:    $\bar{h}_{[1]} = J_{[1]}$ ;  $\bar{h}_{[2]} = J_{[2]}$ ;
21: procedure CONTROL LAW(Altitude Control)
22:   Using theorem 5 calculate  $K_b$  and  $F_i$ 
23:    $U_1 = \bar{h}_{[1]}K_{b[1]}x_b + \bar{h}_{[2]}K_{b[2]}x_b - \bar{h}_{[1]}F_{[1]}\xi - \bar{h}_{[2]}F_{[2]}\xi - m_q g_r$ ;

```

to compute the commanded rotor speed corresponding to the inputs of

the ESC of each DC motor.

$$\begin{aligned}
 \Omega_1 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\
 \Omega_2 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4} \\
 \Omega_3 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\
 \Omega_4 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4}
 \end{aligned} \tag{3.25}$$

If it is considered the receding horizon, the calculations of the gains K_a and K_b through LMIs are performed online, otherwise the gains can be computed offline, then this algorithm could be easily embedded in any open source processor. In the first case it is possible to consider constraints over the control input adding (3.20) to Theorem 5.

Note in algorithm 1 are omitted the steps 13 and 22, if it is used the offline approach.

4 SIMULATION AND DISCUSSION

This section is divided in five parts. Firstly, the system parameters are described and justified. The steps followed to simulate the systems as in continuous as in discrete-time are presented in the second part. The third part shows simulations of the continuous-time controllers exposed in the second chapter without considering limitations and constraints. In the fourth part, it is presented the simulation of the discrete-time controllers, now considering limitations and constraints by the sensors and actuators. Also, the behavior of the actuators, control inputs and the weights of the TS fuzzy models are presented and compared. A brief discussion and pertinent commentaries are presented joining the various figures. Finally, to highlight the importance of choosing a correct T_s we present an example of an attitude system using an inappropriate T_s in the calculation of the feedback gains.

4.1 SYSTEM PARAMETERS DESCRIPTION

It is assumed the use of brushless DC motors that provide high torque and little friction. The parameters that describe the features of these motors are the gain and the time constant. The transfer function maps the desired propeller speed to the actual speed. The voltage supplied to the motors is directly proportional to the rad/s of its rotation. The constant of proportionality of this linear relationship appears as a gain in the transfer function (BOUABDALLAH, 2007) in (4.1)

$$\Lambda(s) = \frac{\textit{Actual rotor speed}}{\textit{Commanded rotor speed}} = \frac{0.936}{0.178s + 1} \quad (4.1)$$

The values of the parameters used in this simulation taken from Yacef e Boudjema (2011) are depicted in Table 3. In Tables 1 and 2 (Chapter 2) are specified the description of the symbols. The sampling times T_a and T_b were chosen according to the average of bandwidths in commercial IMU and ultrasonic sensors. Of course, the election of T_a and T_b could be arbitrary but, to guarantee expected performance and due to implementation requirements, we consider $T_a = 0.05s$ and $T_b = 0.1s$ with $T_b/T_a = \zeta \in \mathbb{Z}$.

Tabela 3 – Quadrotor parameters values

Symbol	Values and units	Symbol	Values and units
m_q	0.486 Kg	L	0.255 m
g_r	9.81 m/s ²	J_r	3.357×10^{-5} Kg.m ²
I_{xx}	3.82×10^{-3} Kg.m ²	I_{yy}	3.82×10^{-3} Kg.m ²
I_{zz}	7.65×10^{-3} Kg.m ²	K_{fax}	5.567×10^{-4}
K_{fay}	5.567×10^{-4}	K_{faz}	6.354×10^{-4}
K_{frz}	0.048	K_m	1.12×10^{-6}
K_f	2.923×10^{-5}	M_A	$\pi/3$ rad
M_{RP}	5 rad/s	M_{RT}	5 rad/s
T_a	0.05 s	T_b	0.1 s
Q_a, Q_b	$\sqrt{20}, \sqrt{10}$	R_a, R_b	1

4.2 STEPS TO APPLY A TS-FC WITH PDC

Once the TS fuzzy model of a system is obtained, no matter if it is an online or an offline controller implementation, to achieve successful results we propose a procedure to implement a continuous or discrete time TS-FC with PDC, described as follows:

1. Firstly, from the model calculate the numerical values of the non-linear submodels.
2. Elaborate a controllability analysis of the submodels of the TS fuzzy system. If any sub-model is uncontrollable, it is necessary to propose modification in the TS fuzzy model to obtain controllability in the system, being able to find feasible solutions to the LMIs conditions.
3. Next, validate the TS fuzzy model. In this case, it should be exact to the original nonlinear model, guaranteeing convexity under the specified region.
4. Define limitations and constraints in the system.
5. Finally establish a control law and using a stability-performance (LMI or polynomial) theorem, if the solution is feasible, find the feedback of the system, then implement it.

4.3 CONTINUOUS TIME SIMULATION

In this sections is depicted in format of comparison the approaches described in Chapter 2. The simulation was done in twenty seconds according to an actual scale of time. In Figure 16 is showed the output states pitch, roll, yaw and throttle. In blue solid line is presented the approach of Lyapunov stability proposed by Tanaka. On the other hand, in black dashed line is presented the approach that use D-region to allocate the poles in an arbitrary region. According to Figure 11, in this case the poles are bounded in a circle with center in -3 with radius equal to 2 on the S-plane for the altitude controller; for the attitude controller the center of the circle was defined in -10 with radius equal to 5 . Finally, with red dot and dashed line is set the reference to be tracked.

Mostly, quadrotors joysticks have configurable buttons. The buttons of roll, pitch and yaw are set with steps with amplitude of ± 0.8 , ± 0.8 and -0.3 rad . respectively hovering in $-4m$. along the $Z - axis$. These, values were chosen to test the effectiveness of the nonlinear controller.

In Figure 17 it is possible to see the control inputs of the closed loop system. In black dashed line is presented the the D-region approach. In blue solid line Lyapunov stability approach. U_1 represents the thrust, in Newtons units, needed by the DC motors, U_2 and U_3 expressed in $N.m$. are the angular momentums in X an Y axes needed to get a slope in the quadrotor and U_4 is the angular momentum in Z axis to head the quadrotor in a desired position.

In Figure 18 is presented in rad/s the behavior of the four electric actuators with blue solid line D-region approach and using black dashed is showed the Lyapunov stability approach. Apparently there is not a big difference between these result, but in Figure 16 it is possible to see how a small variation of the RPM has a considerable influence in the performance of the system.

The weights of the TS fuzzy model are depicted in Figure 19. The first pair of pictures belong to the altitude subsystem and the others to the attitude subsystem. In magenta solid line are represented the weights that correspond to the D-region approach and the others to the Lyapunov stability approach.

In coclusion, despite both of the approaches have the same processing effort, the approach with D-region offers a better performance.

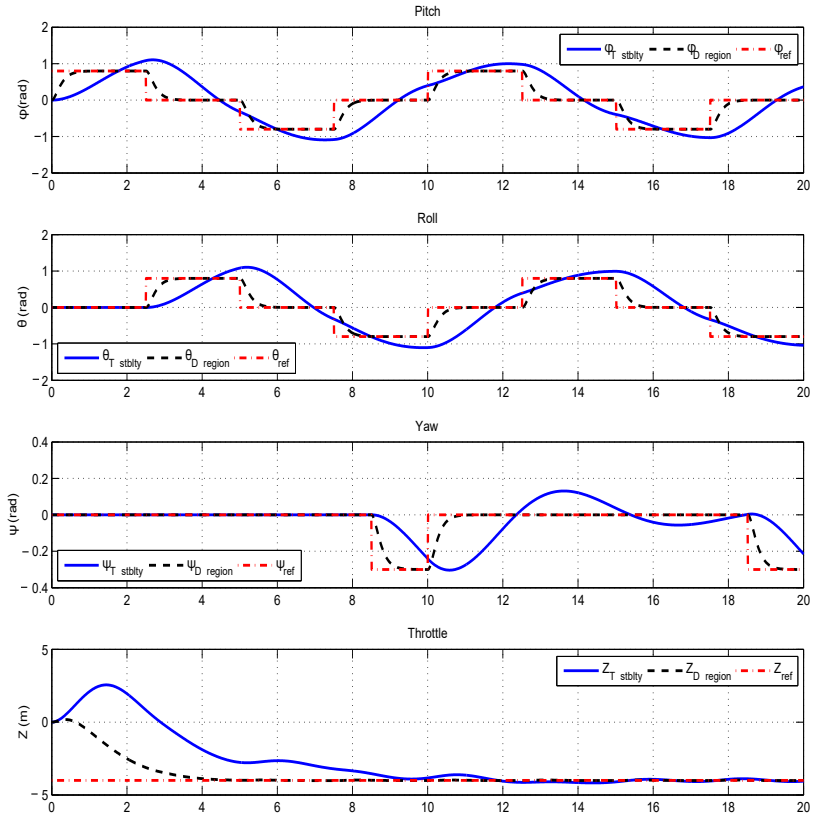


Figura 16 – Tracking performance in continuous-time

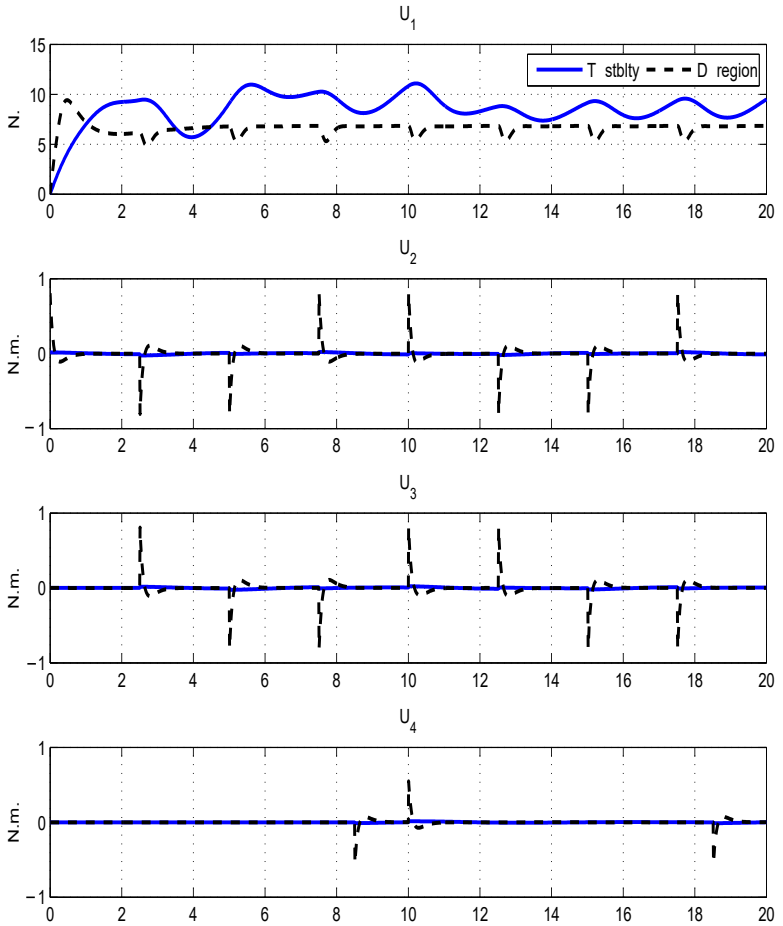


Figure 17 – Control inputs in continuous-time

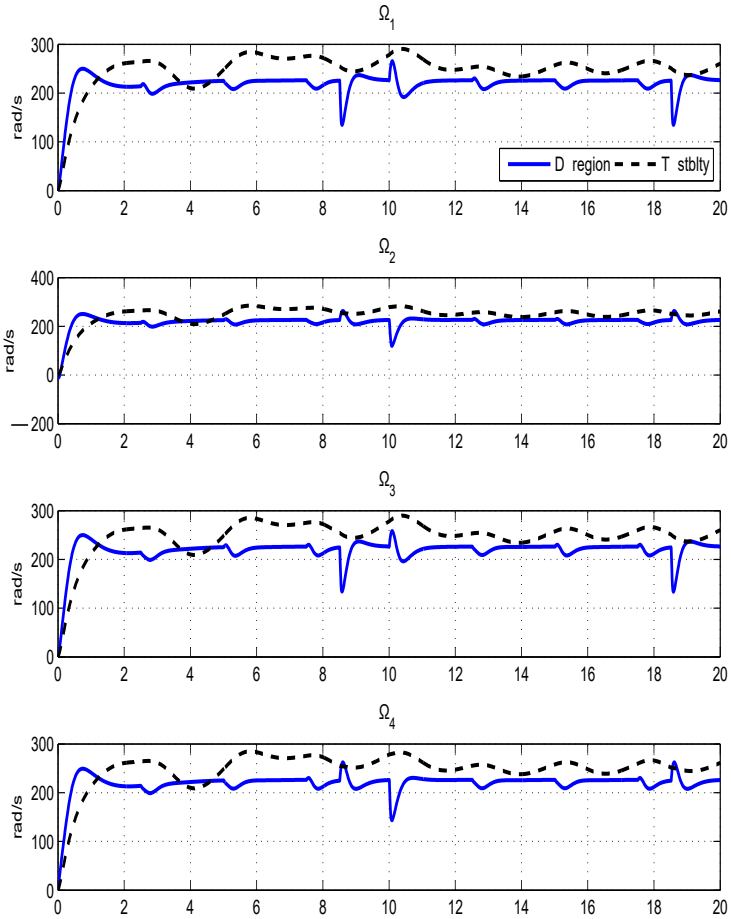


Figura 18 – DC motors revolutions in continuous-time

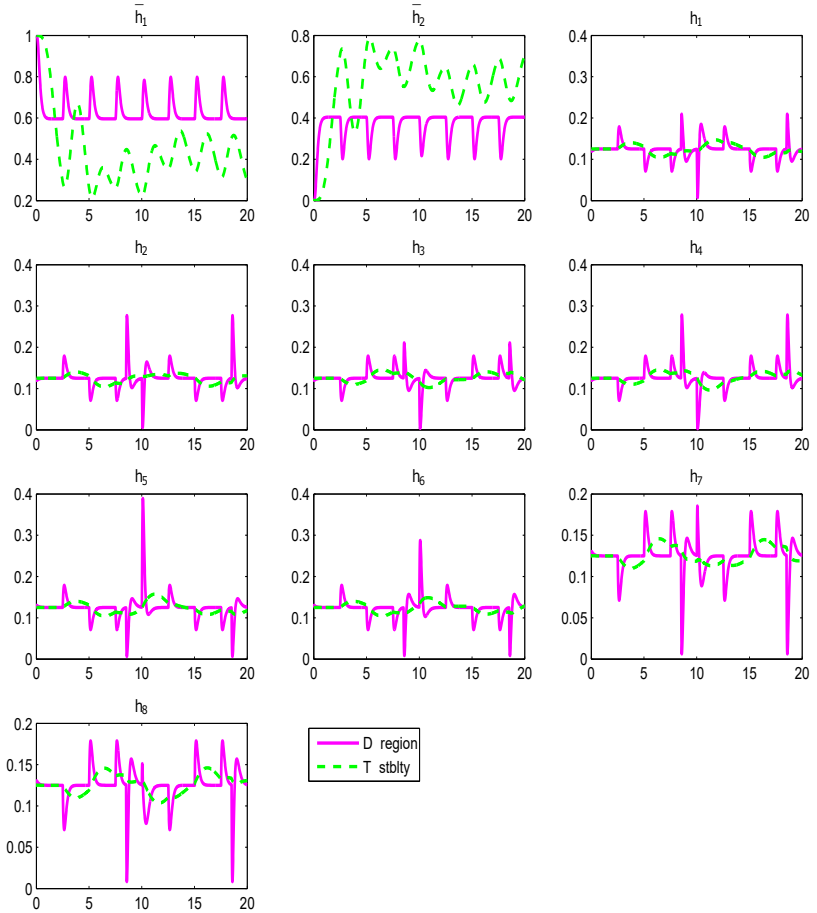


Figure 19 – Weights of the TS fuzzy model in continuous-time

4.4 DISCRETE TIME SIMULATION

In the last section was presented a simulation totally in continuous time. Now, in this section is presented a simulation considering the hybrid behavior of the system that is taking into account the continuous time behavior of the states of the quadrotor system, such as the position and speed, and the discrete time behavior of the sensors, processor and other electronic parts being this simulation closer to an actual plant. The simulation was done in ten seconds according to an actual scale of time. The selection of the sampling time in the loops of the control structure is done based on the data sheets of the sensors. It was possible to realize that the frequency of the sensors used in quadrotors work with a low frequency compared to the other electronic elements involved. Other important fact is to consider constraints in the system, in this case applied to the control inputs. It was chosen the constraints in the control input U_1 because it is in charge of the thrust of the quadrotor. The size and weight of the DC motors depend of the power required. For this reason, it was assumed that we have four DC motors that provide a maximum thrust of $10N$. Due to these limitations and constraints it was concluded about the importance to take them into account in the whole design of the discrete-time controller.

Due to cascade control strategy, it is necessary to have a inner loop faster than the outer loop. Accordingly, we choose $Q_a = \sqrt{20}.I(n_x)$, $Q_b = \sqrt{10}.I(n_x)$, $R_a = 1.I(n_u)$ and $R_b = 1.I(n_u)$. The comparison was carried out between three techniques presented in Chapter 3. Thus, in Figure 20 is presented with blue solid line the techniques that uses poles allocation, with black dashed lines the offline technique of optimal control, with magenta dot and dashed line the online optimal control approach and with red dotted line the reference. Because of the absence of feasibility in the solving of the online optimal control approach, it was removed the integral action in both optimal techniques causing steady state error in the altitude subsystem. The poles allocation technique looks a little better in performance comparing with the optimal control approaches in the attitude subsystem. On the other hand in the altitude subsystem the performance is better in the optimal control techniques, but these techniques present steady state error.

It is possible to see U_1 in Figure 21 that the poles allocation technique, in blue solid line, demands less than $10N$ of thrust, in the offline optimal control technique, in black dashed line, it was required almost $40N$, then applying the constraint LMI condition presented in the last chapter is obtained, in magenta dot and dashed line, the online

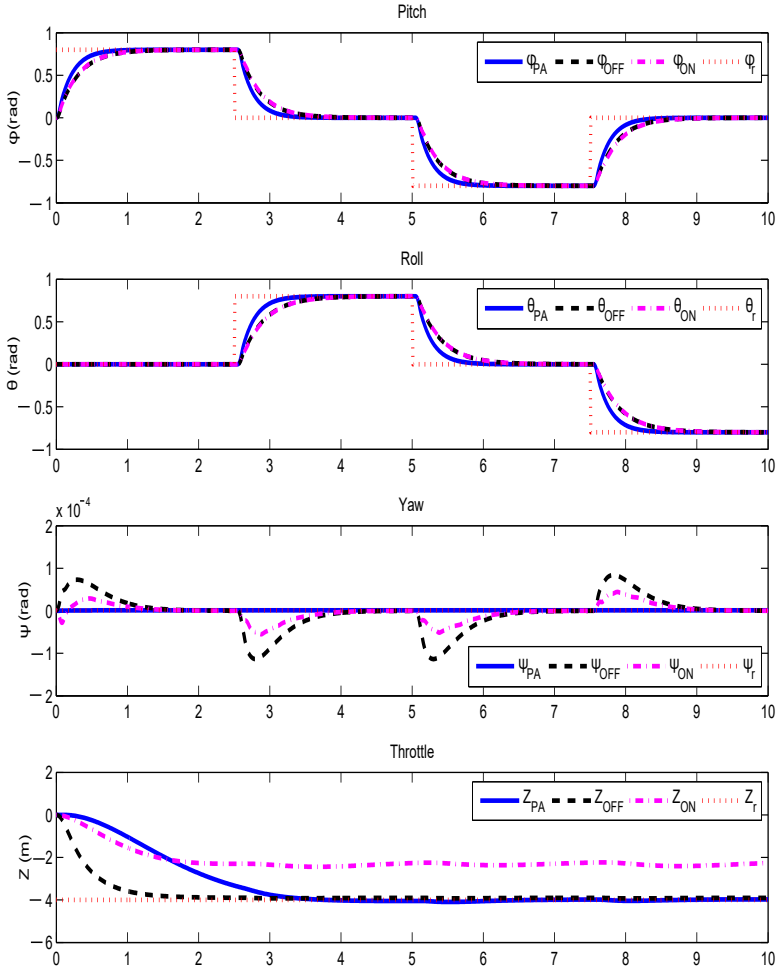


Figure 20 – Tracking performance comparison in discrete-time

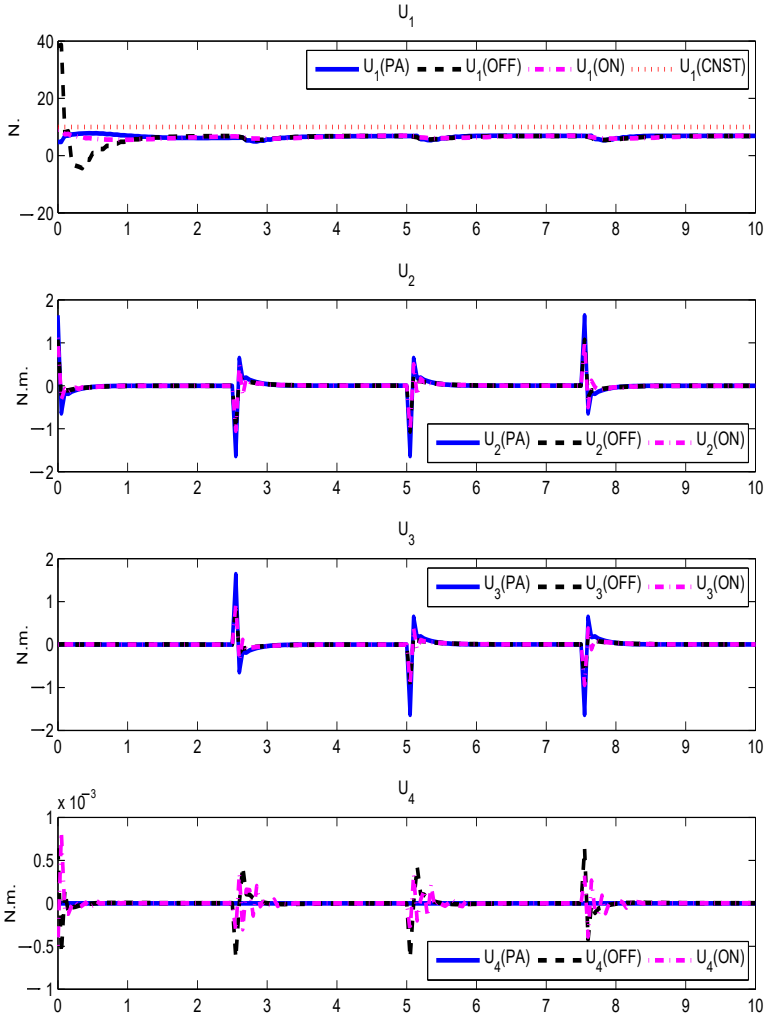


Figure 21 – Control inputs in discrete time for attitude subsystem

control input bounded to $10N$ of thrust. This constraint caused an increase in the steady state error as we can see in the fourth picture on the Figure 20. As was expected according to the performance U_2 , U_3 and U_4 do not present relevant difference in its values.

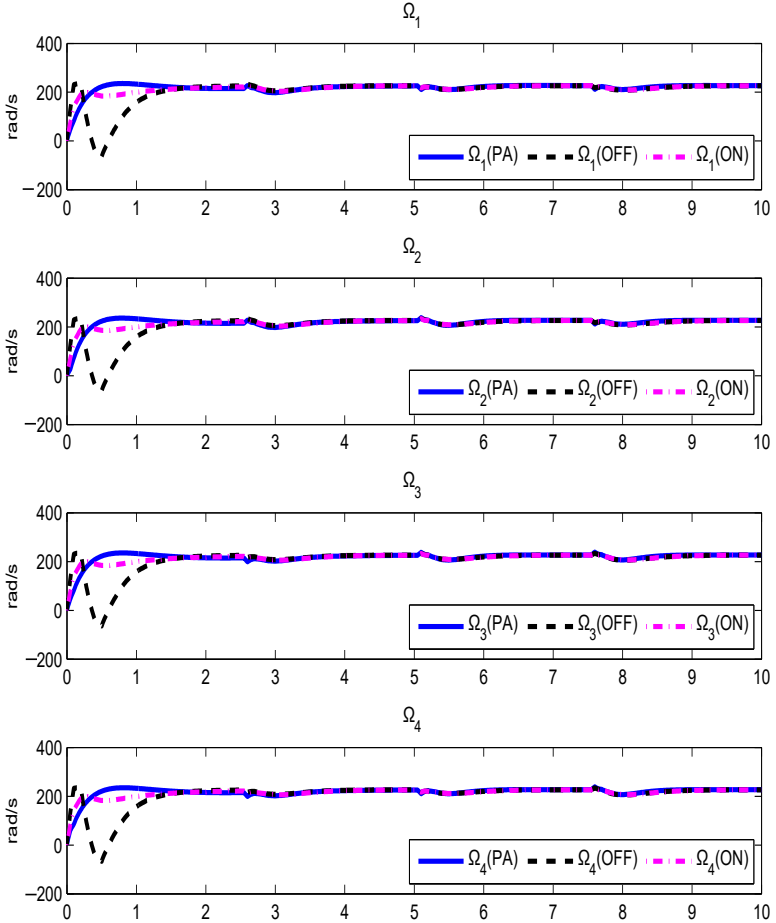


Figure 22 – DC motor revolutions in discrete-time

In Figure 22 is depicted, in black dashed line, a deplorable behavior in the offline optimal control approach without considering constraints by the DC motors, it is possible to see that is demanded negative

thrust which is impossible in an actual application highlighting the importance of constraints in the develop of the feedback control. The other two approaches have an acceptable developing in the required RPM.

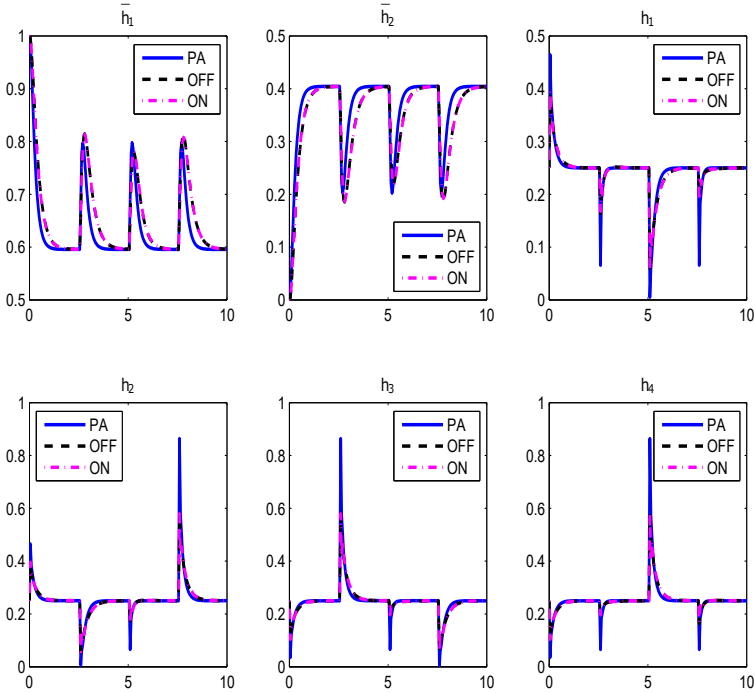


Figure 23 – Weights of the TS fuzzy model in discrete-time

Finally, in Figure 23 is showed the weights of the TS fuzzy models of the three approaches being very similar between each other. It happened because the similar tracking performance offered by the three techniques.

Remark 4.1.

To emphasize the correct selection of the T_s , we present an example of an attitude system with an inappropriate choice of T_a . Because of the reading frequency of the IMU, the minimum allowable T_a is $0.5s$. Let us consider an inadequate choice of a faster sampling time $T_a = 0.035s$, that was used to compute the gains using the Theorem 3 presented in

Chapter 3. The corresponding result is depicted in Figure 24. The system turns unstable, thus we can appreciate the importance of a correct T_s when we are working with discrete time systems.

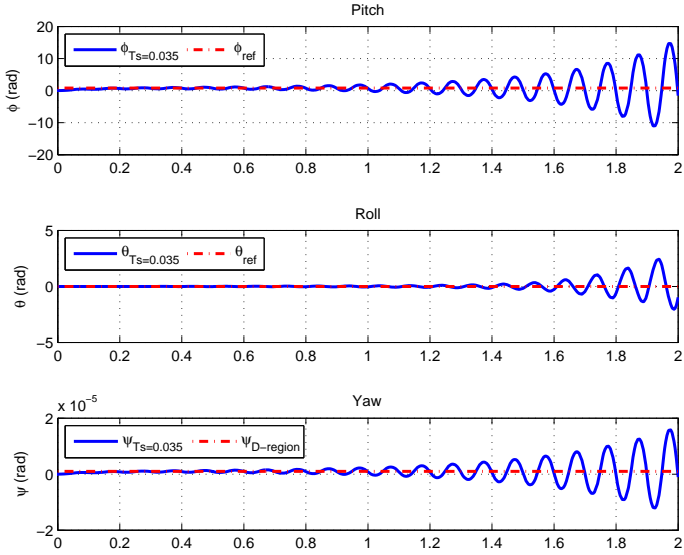


Figure 24 – Inappropriate sampling time test

5 CONCLUSION

In this chapter are detailed the conclusion and remarks obtained from all the chapters of this document based on the objectives detailed in chapter 1. As the objectives, the conclusions are divided in continuous and discrete time application as follows:

5.1 CONCLUSIONS ABOUT THE CONTINUOUS TIME APPLICATION

1. One of the most important challenges of this research was to obtain a TS fuzzy model of the plant with the lowest quantity of rules as possible. The result was more than satisfactory because it was generated a TS fuzzy model, that represents exactly similar the original model, with the lowest number of rules compared with all the referenced articles cited in this work without losing information about the dynamics of the system. Therefore, as consequence it was obtained an efficient processing.
2. Using random inputs, it was validated the TS fuzzy model which, as was expected, showed zero error compared to the original plant being trustworthy the fuzzy model.
3. Based in the result of the last chapter it was concluded that the only stability theorems, like the stability theorem proposed by Tanaka e Wang (2004), are not enough to get good results in simulation. So, it is important to complement these theorems with conditions to establish arbitrary features by the user, like poles allocation, which give the option to set a desired time of response and dumping coefficient in the output states. Thereby, the results using the theorem with D-region proposed by Hong e Nam (2003) were a sight better in all the aspects than the first ones.

5.2 CONCLUSIONS ABOUT THE DISCRETE TIME APPLICATION

1. The cascade control structure proposed was effective to obtain good performance results because, thanks to this structure, it is possible to set a faster dynamic in the attitude subsystem than the

altitude which can be interpreted like an extra degree of freedom to design the control algorithm.

2. It was presented and tested through simulation a generic algorithm. If it is used the offline approaches, the implementation can be easily done in any open source processor. If it is desired to implement the online approach, it has a drawback related to the computing of the feedback in each sampling time. The time of simulation of ten seconds about the online approach in a $2.7GHz$ laptop with $16Gb$ of RAM was more and less 40 minutes. For this reasons is not recommended to use this algorithm in quadrotors at least until it would be able more powerful processors for quadrotors in the future.
3. The performance in the results were similar in all the cases. There was feasibility problems in the online approach causing steady state error. Moreover, the main drawback is the processing effort demanded by the online optimal control technique. For these reasons it is suggested the controller using the theorem presented in Klug (2010) for be implemented in quadrotors.

5.3 PERSPECTIVES

Among some possible extensions to this work, it can be mentioned the following research directions:

- Research the use of the techniques presented in this document using non-PDC strategies applied in quadrotors and compare with the results obtained in this document.
- Looking to save energy in batteries, compare the processing effort of these techniques versus other nonlinear control strategies.
- The problem of feasibility in the online controllers maybe solved using Piecewise functions instead of common Lyapunov functions. It can be analyzed in future works too.
- Consider the employment of discretized TS fuzzy submodels in continuous time and compare with the results presented in this document.
- Implement the strategies and control algorithms in real plants.

REFERENCES

- AZZAM, A.; WANG, X. Quadrotor arial robot dynamic modeling and configuration stabilization. v. 1, p. 438–444, 2010.
- BINDER, Z. et al. About a multimodel control methodology, algorithm, multi-processors, implementation and application. p. 981–986, 1981.
- BLAŽIČ, S.; ŠKRJANC, I.; MATKO, D. Globally stable direct fuzzy model reference adaptive control. **Fuzzy Sets and Systems**, Elsevier, v. 139, n. 1, p. 3–33, 2003.
- BOUABDALLAH, S. **Design and control of quadrotors with application to autonomous flying**. Tese (Doutorado) — Ecole Polytechnique Federale de Lausanne, 2007.
- CAIRANO, S. D. et al. Model predictive control of magnetically actuated mass spring dampers for automotive applications. **International Journal of Control**, Taylor & Francis, v. 80, n. 11, p. 1701–1716, 2007.
- CAMACHO, E. F.; ALBA, C. B. **Model predictive control**. [S.l.]: Springer Science & Business Media, 2013.
- CARRILLO, L. R. G. et al. **Quad rotorcraft control: vision-based hovering and navigation**. [S.l.]: Springer Science & Business Media, 2012.
- CHOI, Y.-C.; AHN, H.-S. Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. **IEEE/ASME transactions on mechatronics**, IEEE, v. 20, n. 3, p. 1179–1192, 2015.
- CISNEROS, P. et al. Linear parameter-varying controller design for a nonlinear quad-rotor helicopter model for high speed trajectory tracking. p. 486–491, 2016.
- FENG, G. **Analysis and Synthesis of Fuzzy Control Systems: a model-based approach**. [S.l.]: CRC press, 2010.
- GAITAN, A. T.; BOLEA, Y. Modeling and robust attitude control of a quadrotor system. p. 7–12, 2013.

GAUTAM, D.; HA, C. Control of a quadrotor using a smart self-tuning fuzzy pid controller. **International Journal of Advanced Robotic Systems**, InTech, v. 10, 2013.

HO, W.-H.; CHOU, J.-H. Design of optimal controllers for takagi–sugeno fuzzy-model-based systems. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 37, n. 3, p. 329–339, 2007.

HONG, S. K.; NAM, Y. Stable fuzzy control system design with pole-placement constraint: an lmi approach. **Computers in Industry**, Elsevier, v. 51, n. 1, p. 1–11, 2003.

KLUG, M. **Realimentacao Dinamica de Saidas com Parametros Variantes e Aplicacao aos Sistemas Fuzzy Takagi-Sugeno**: Disertacao de mestrado. Florianópolis, 2010. Acesso em: 11 jun. 2011.

KLUG, M. **Control of nonlinear systems using N-fuzzy models**. Brazil: Ph.D. Thesis, Federal University of Santa Catarina, 2015.

KLUG, M.; CASTELAN, E. B.; COUTINHO, D. At–s fuzzy approach to the local stabilization of nonlinear discrete-time systems subject to energy-bounded disturbances. **Journal of Control, Automation and Electrical Systems**, Springer, v. 26, n. 3, p. 191–200, 2015.

KLUG, M. et al. Compensadores dinâmicos para sistemas nao lineares utilizando modelos fuzzy ts: Estudo comparativo e implementaçao hil. 2014.

KLUG, M. et al. Fuzzy dynamic output feedback control through nonlinear takagi–sugeno models. **Fuzzy Sets and Systems**, Elsevier, v. 263, p. 92–111, 2015.

LEE, H.; KIM, H. J. Robust control of a quadrotor using takagi-sugeno fuzzy model and an lmi approach. p. 370–374, 2014.

MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. **IEEE robotics & automation magazine**, IEEE, v. 19, n. 3, p. 20–32, 2012.

OHTAKE, H.; TANAKA, K.; WANG, H. O. Fuzzy modeling via sector nonlinearity concept. **Integrated Computer-Aided Engineering**, IOS Press, v. 10, n. 4, p. 333–341, 2003.

- ORDAZ, J. et al. Predictor-based position control of a quad-rotor with delays in gps and vision measurements. **Journal of Intelligent & Robotic Systems**, Springer, v. 70, n. 1-4, p. 13–26, 2013.
- PATEL, A. R.; PATEL, M. A.; VYAS, D. R. Modeling and analysis of quadrotor using sliding mode control. p. 111–114, 2012.
- PRECUP, R.-E. et al. Design and experiments for a class of fuzzy controlled servo systems. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 13, n. 1, p. 22–35, 2008.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE transactions on systems, man, and cybernetics**, IEEE, n. 1, p. 116–132, 1985.
- TANAKA, K.; SUGENO, M. Stability analysis and design of fuzzy control systems. **Fuzzy sets and systems**, Elsevier, v. 45, n. 2, p. 135–156, 1992.
- TANAKA, K.; WANG, H. O. **Fuzzy control systems design and analysis: a linear matrix inequality approach**. [S.l.]: John Wiley & Sons, 2004.
- TORRES, F. et al. Fuzzy state feedback for attitude stabilization of quadrotor. **International Journal of Advanced Robotic Systems**, InTech, v. 13, 2016.
- WANG, H. O.; TANAKA, K.; GRIFFIN, M. F. An approach to fuzzy control of nonlinear systems: stability and design issues. **IEEE transactions on fuzzy systems**, IEEE, v. 4, n. 1, p. 14–23, 1996.
- YACEF, F.; BOUDJEMA, F. Local model network for non linear modelling and control of an uav quadrotor. p. 247–252, 2011.
- YACEF, F. et al. Takagi-sugeno model for quadrotor modelling and control using nonlinear state feedback controller. **International Journal of Control Theory and Computer Modelling (IJCTCM)**, v. 2, n. 3, p. 9–24, 2012.

ANNEX A – Summary - modeling of a quadrotor

In this appendix will be summarized the modeling of a UAV-quadrotor considering air friction. The kinematics and dynamics models will be derived based on a Newton-Euler formalism with the next presumptions:

- The quadrotor structure is rigid and symmetrical.
- The center of gravity coincides with the body fixed frame origin.
- The propellers are rigid.
- Thrust and drag are proportional to the square of the speed of the propellers.

Firstly, we denote the coordinate frames based in the Figure 1, which shows the earth references frames with N, E and D axes and the quadrotor frame with x, y and z axes. The distance between the Earth frame and the body frame describes the absolute postion of the center of mass through $\mathbf{r} = [x \ y \ z]'$. The rotation R is described using roll, pitch and yaw angles (ϕ , θ and ψ) representing rotations about the X,Y and Z-axes respectively. The rotation matrix R which is derived based on the sequence of principle rotation is:

$$\mathbf{R} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\theta c\psi & c\phi s\theta s\psi - s\theta c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}, \quad (\text{A.1})$$

where c and s denote \cos and \sin respectively. To obtain information about the angular velocity of the quadrotor, typically an on-board Inertial Measurement Unit (IMU) is used which will give it in the body coordinate frame. To relate the Euler rated $\dot{\boldsymbol{\eta}} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]'$ and the angular body rates $\boldsymbol{\omega} = [p \ q \ r]'$, a transformation $\boldsymbol{\omega} = \mathbf{R}_r \dot{\boldsymbol{\eta}}$ is needed, where:

$$\mathbf{R}_r = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}, \quad (\text{A.2})$$

Until this point was detailed the kinematics of the body, now we will expose the motion of the quadrotor using the dynamics model. The motion of the UAVs can be divided into two subsystem; rotational subsystem (roll, pitch and yaw) and translational (altitude or z and x-y postion). The rotational susbsystem is fully actuated while the

translational is underactuated.

The rotational equation of motion using the Newton-Euler method is:

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}_G = \mathbf{M}_B \quad (\text{A.3})$$

where:

\mathbf{J} : Quadrotor's diagonal inertia Matrix

$\boldsymbol{\omega}$: Angular body rates

\mathbf{M}_G : Gyroscopic moments due to the inertia of the rotor

\mathbf{M}_B : Moments acting on the quadrotor in the bbody frame

The gyroscopic moments are defined to be $\boldsymbol{\omega} \times [\mathbf{0} \ \mathbf{0} \ \mathbf{J}_r \boldsymbol{\omega}_r]'$, thus the rotational equation of the quadrotor can be rewritten as:

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\omega} \times [\mathbf{0} \ \mathbf{0} \ \mathbf{J}_r \boldsymbol{\Omega}_r]' = \mathbf{M}_B \quad (\text{A.4})$$

where:

\mathbf{J}_r : Inertia of the rotor

$\boldsymbol{\Omega}_r$: Relative speed of the rotor

Due to the symmetry of the quadrotor the off-diagonal elements of the inertia matrix are zero as follows:

$$\mathbf{J} = \begin{bmatrix} I_{xx} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{yy} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{zz} \end{bmatrix}, \quad (\text{A.5})$$

Where I_{xx} , I_{yy} and I_{zz} are the area moments of inertia about the principle axes in the body frame.

To define the moments acting on the quadrotor (\mathbf{M}_b) there is a need to define two physical effects which are the aerodynamic forces and moments produced by a rotor. As an effect of rotation, there is a generated force called the aerodynamic force or the lift force and there is a generated moment called the aerodynamic moment. Respectively represented by:

$$F_i = \frac{1}{2} \rho A C_T r^2 \Omega_i^2, \quad (\text{A.6})$$

$$M_i = \frac{1}{2} \rho A C_D r^2 \Omega_i^2, \quad (\text{A.7})$$

where:

ρ : Air density

A : Blade area

C_T, C_D : Aerodynamic coefficients

r : Radius of the blade

ω_i : Angular velocity of each i rotor

It is possible to appreciate that the aerodynamical forces and moments depend on the geometry of the propeller and the air density. The last two equations can be simplified respectively to:

$$F_i = K_f \Omega_i^2, \quad (\text{A.8})$$

$$M_i = K_m \Omega_i^2, \quad (\text{A.9})$$

Where K_f and K_m are the aerodynamic force and moment constants respectively and ω_i is the angular velocity of rotor i . The aerodynamic force and moment constants can be determined experimentally for each propeller type. Using the fig.1.1, it is possible to set the moments about the body frame's x-axis, by using the right-hand rule in association with the axes of the body frame, F_2 multiplied by the moment arm L generates a negative moment about the y-axis, while in the same manner, F_4 generates a positive moment. Thus the total moment about the x-axis can be expressed as:

$$\begin{aligned} M_x &= -F_2 L + F_4 L \\ &= -(K_f \Omega_2^2) L + (K_f \Omega_4^2) L \\ &= L K_f (-\Omega_2^2 + \Omega_4^2) \end{aligned} \quad (\text{A.10})$$

About the body frame's y-axis, also using the right-hand-rule, the thrust of rotor 1 generates a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis. Thus, total moment can be expressed as:

$$\begin{aligned} M_y &= F_1 L - F_3 L \\ &= (K_f \Omega_1^2) L - (K_f \Omega_3^2) L \\ &= L K_f (\Omega_1^2 - \Omega_3^2) \end{aligned} \quad (\text{A.11})$$

For the moments about the body frame's z-axis, the thrust of the rotors does not cause a moment. On the other hand, moment caused by the rotors' rotation as (A.7), by using the right-hand-rule, the moment

about the body frame of z-axis can be expressed as:

$$\begin{aligned}
 M_z &= M_1 - M_2 + M_3 - M_4 \\
 &= (K_m \Omega_1^2) - (K_m \Omega_2^2) + (K_m \Omega_3^2) - (K_m \Omega_4^2) \\
 &= K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (\text{A.12})
 \end{aligned}$$

In a vector form the last three equations can be expressed as:

$$M_B = \begin{bmatrix} LK_f (-\Omega_2^2 + \Omega_4^2) \\ LK_f (\Omega_1^2 - \Omega_3^2) \\ K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}, \quad (\text{A.13})$$

On the other hand the translational equations of motion are based on Newton's second law and they are derived in:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ mg \end{bmatrix} + \mathbf{R}F_B \quad (\text{A.14})$$

where:

$\mathbf{r} = [x \ y \ z]'$: Distance between the quadrotor and the inertial frame

m : Mass of the quadrotor

\mathbf{g} : Gravity

F_B : Nongravitational forces

The nongravitational forces are the ones that deal on the quadrotor in a horizontal orientation, it is no rolling or pitching, which can be expressed as:

$$F_B = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -K_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix}, \quad (\text{A.15})$$

The zeros in the firsts two rows are because there are not forces in the \mathbf{x} and \mathbf{y} directions, and the last row is an addition of the thrust forces produced by the four propellers. The negative sign is due to the fact that the thrust is upwards while the positive z-axis in the body framed is pointing downwards.

It is important to highlight that was included in our dynamic system aerodynamic effects, there are namely two types of aerodynamic effects,

drag forces and drag moments.

Due to the air friction, a force acts on the body of the quadrotor resisting the motion. As the velocity of the body increase the same happen with the drag forces and moments. The drag forces \mathbf{F}_a and the drag moment \mathbf{M}_a can be approximated respectively by:

$$\mathbf{F}_A = \mathbf{K}_t \dot{\mathbf{r}}, \quad \mathbf{M}_a = \mathbf{K}_r \dot{\boldsymbol{\eta}}, \quad (\text{A.16})$$

Including the aerodynamic effects in (A.14) and (A.4) we have:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ mg \end{bmatrix} + \mathbf{R}\mathbf{F}_B - \mathbf{F}_a \quad (\text{A.17})$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\omega} \times [\mathbf{0} \quad \mathbf{0} \quad J_r \Omega_r]' = \mathbf{M}_B - \mathbf{M}_a \quad (\text{A.18})$$

Control input vector

In this section of this appendix is detailed the relation between the inputs U_i and the rotations speed of the rotors $\boldsymbol{\omega}_i$ as follows:

$$\begin{aligned} U_1 &= K_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\ U_2 &= K_f (-\Omega_2^2 + \Omega_4^2), \\ U_3 &= K_f (\Omega_1^2 - \Omega_3^2), \\ U_4 &= K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2), \end{aligned} \quad (\text{A.19})$$

Equations (A.19) can be arranged in a matrix as:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ \mathbf{0} & -K_f & \mathbf{0} & K_f \\ K_f & \mathbf{0} & -K_f & \mathbf{0} \\ K_m & -K_m & K_m & -K_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}, \quad (\text{A.20})$$

To determinate the rotor velocities, it is only necessary to obtain the inverse of the matrix that include the coefficients as follows:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & \mathbf{0} & \frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & \mathbf{0} & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & \mathbf{0} & -\frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & \mathbf{0} & -\frac{1}{4K_m} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}' \quad (\text{A.21})$$

Therefore, the rotor velocities can be calculated from the control inputs

as follows:

$$\begin{aligned}
\Omega_1 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\
\Omega_2 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4} \\
\Omega_3 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\
\Omega_4 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4} \quad (\text{A.22})
\end{aligned}$$

Now, replacing (A.19) in (A.13), the equation of the total moments acting on the quadrotor becomes:

$$M_B = \begin{bmatrix} LU_2 \\ LU_3 \\ U_4 \end{bmatrix}, \quad (\text{A.23})$$

Substituting (A.23) into the rotational equation of motion (A.4) the next relation can be derived:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} = \begin{bmatrix} LU_2 \\ LU_3 \\ U_4 \end{bmatrix}, \quad (\text{A.24})$$

From (A.24) and adding the aerodynamic effects due to air friction are obtained the rotational equation of motion (2.1).

For the translational equations of motion, it is necessary to replace (A.19) in (A.15) as follows:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix}, \quad (\text{A.25})$$

Next, substituting (A.25) into the translational equation of motion (A.14), we get:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\theta c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\theta \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix}, \quad (\text{A.26})$$

From (A.26) and using the aerodynamic effects due to air friction are obtained the translational equation of motion depicted in (2.11).

ANNEX B – Validation of the TS modeling

In the Chapter 4 of this documents were exposed the steps to implement a parallel distributed compensation controller, the third step is to validate the system. If the system is continuous through the time and the fuzzy TS is applied correctly, it is sure that the TS fuzzy model will represent exactly the original model under the region predefined by the user. The signal source was an uniform random number, which is a random signal with limited amplitude, it is important because if the signal is not limited in its amplitude, it is possible to quit the validity region of modeling. The strategy of modeling, since the beginning, was

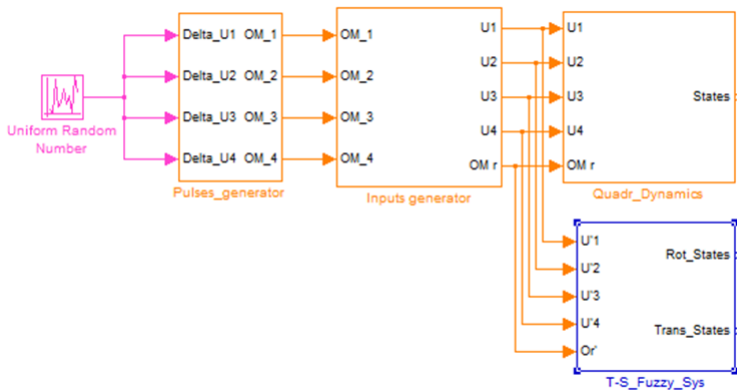


Figura 25 – Simulink validation scheme

to divide into two subsystem, as depicted in the Figure 3 we have the rotational subsystem and the translational subsystem.

The first attempt of modeling was including all the system, but the translational system in not fully actuated, therefore, it was modeled into two fully actuated systems, thus, it is possible to analyze easier the system. In Figure 25, it is showed that the original system states and the fuzzy TS modeling match exactly superimposed, which means error equivalent to zero. It is important to highlight that was taken care about the region of validity of the system through this procedure. Starting off this validation, then it is possible to apply any fuzzy control technique such as PDC, optimal or model predictive control over this fuzzy model.

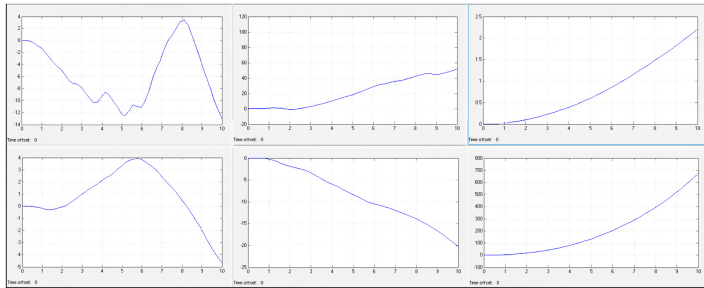
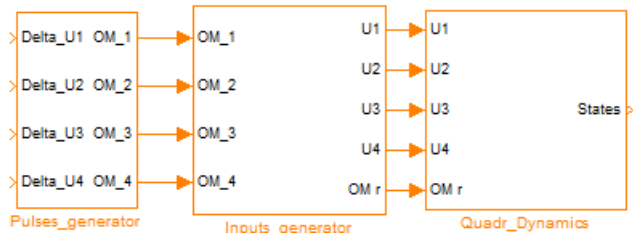


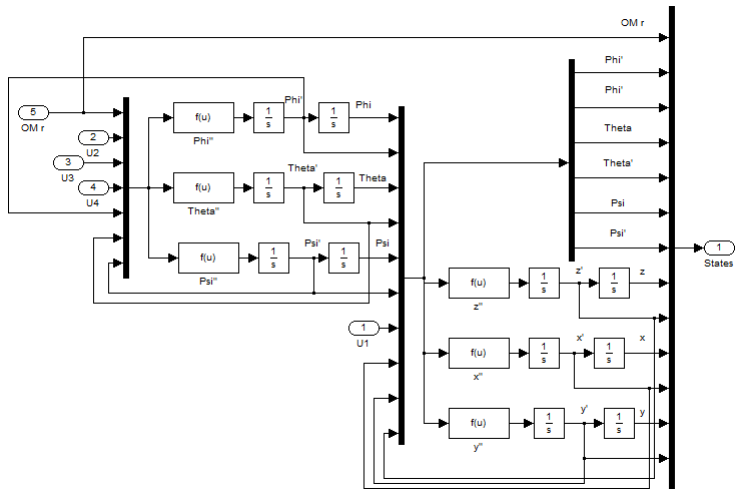
Figura 26 – Validation of ϕ , θ , ψ , x , y and z states

ANNEX C – Simulink schemes

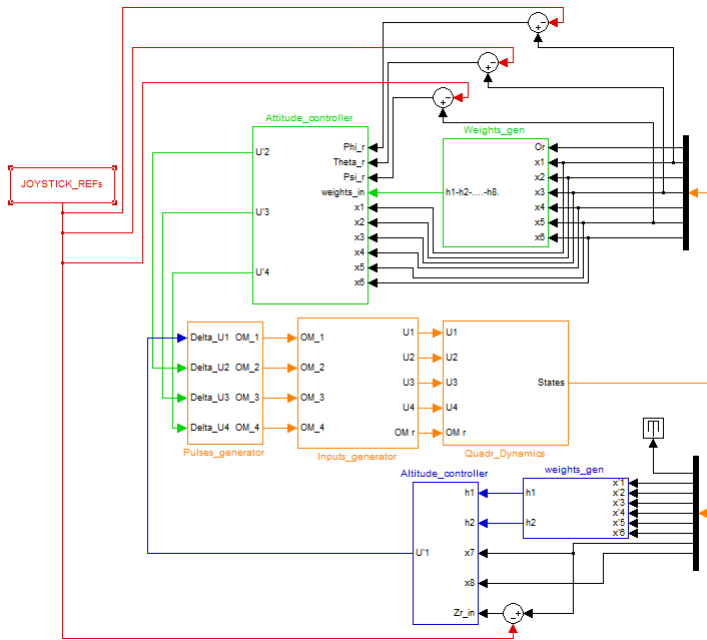
C.1 QUADROTOR MODEL



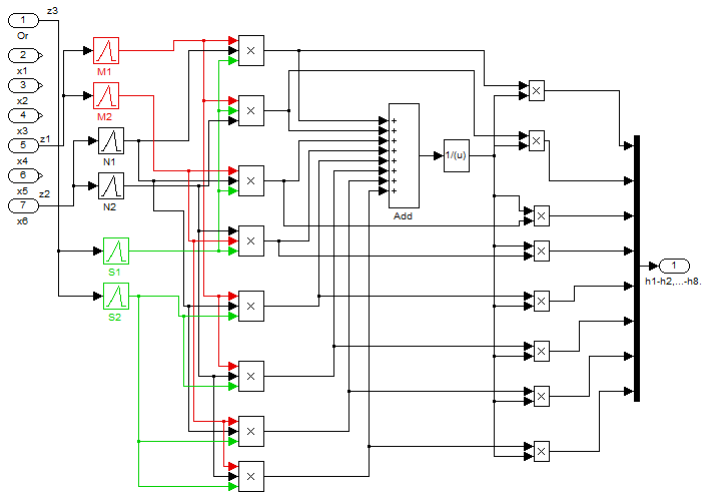
C.2 QUADROTOR DYNAMICS



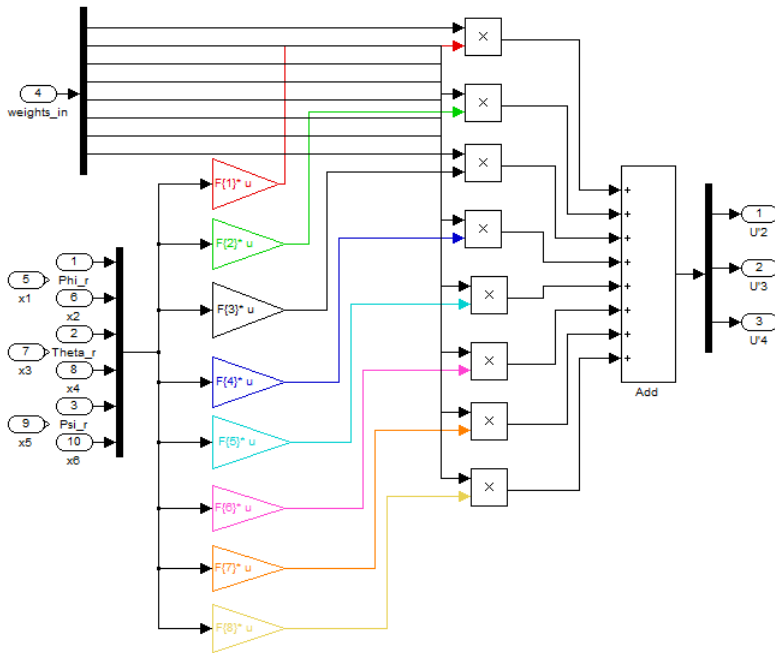
C.3 FUZZY TS - PDC



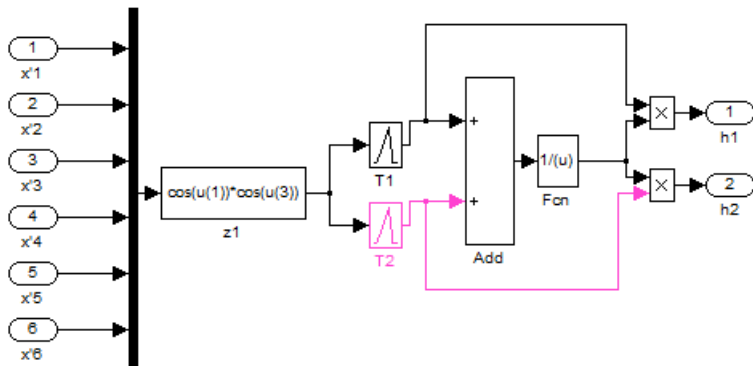
C.4 ATTITUDE CONTROL - WEIGHTS GENERATOR



C.5 ATTITUDE CONTROLLER



C.6 ALTITUDE CONTROL - WEIGHTS GENERATOR



C.7 ALTITUDE CONTROLLER



ANNEX D – Augmented PI dynamic in state equation

Consider the linear system:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \\ \mathbf{y} &= \mathbf{C}\mathbf{x},\end{aligned}\tag{D.1}$$

For convenience, the next assumption are done:

- The reference is denoted by \mathbf{r} , which is the desired constant value for the output \mathbf{y} to track asymptotically.
- The reference $\mathbf{r} \neq \mathbf{0}$.
- The number of inputs "n" is equal to the number of outputs "p".
- The state \mathbf{x} and the reference \mathbf{r} are available

With the aim to find a control law, which provide null error $\mathbf{e} = \mathbf{r} - \mathbf{y} = \mathbf{0}$ in infinite time $t \rightarrow \infty$, it is assumed that the state and the control converge to steady state values as $t \rightarrow \infty$, lets denote:

$$\begin{aligned}\hat{\mathbf{x}} &= \lim_{t \rightarrow \infty} \mathbf{x}, \\ \hat{\mathbf{u}} &= \lim_{t \rightarrow \infty} \mathbf{u},\end{aligned}\tag{D.2}$$

For asymptotic tracking, $\hat{\mathbf{x}}$ and $\hat{\mathbf{u}}$ should satisfy the equation:

$$\underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_E} \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{r},\tag{D.3}$$

where \mathbf{I} is a $p \times p$ identity matrix and $\mathbf{0}$ is a zero matrix of size $n \times p$. The matrix \mathbf{A}_E is square due to equal number of inputs and outputs. If \mathbf{A}_E is nonsingular, we can express:

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{M}_x \mathbf{r}, \\ \hat{\mathbf{u}} &= \mathbf{M}_u \mathbf{r},\end{aligned}\tag{D.4}$$

where:

$$\begin{bmatrix} \mathbf{M}_x \\ \mathbf{M}_u \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix},\tag{D.5}$$

Now, we augment the system (D.1) with a differential equation:

$$\dot{\boldsymbol{\xi}} = \mathbf{y} - \mathbf{r},\tag{D.6}$$

Since \mathbf{y} and \mathbf{r} are both measured, we can determinate $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \int_0^t [\mathbf{y} - \mathbf{r}] d\tau, \quad (\text{D.7})$$

The last expression is valid for $\boldsymbol{\xi}_0 = \mathbf{0}$. we can write the augmented system in the form:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{r},$$

Control laws are taken to be of the form:

$$\mathbf{u} = -(\mathbf{K}\mathbf{x} + \mathbf{F}\boldsymbol{\xi}),$$

The last equations are available for continuous time systems, if it is desired to apply this criteria in discrete time systems (D.6) turns in:

$$\boldsymbol{\xi}(k+1) = \boldsymbol{\xi}(k) + \mathbf{y}(k), \quad (\text{D.8})$$

Thus, the augmented system for discrete time systems is:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \boldsymbol{\xi}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u},$$

The scheme that represents this criteria is depicted in the next Simulink figure:

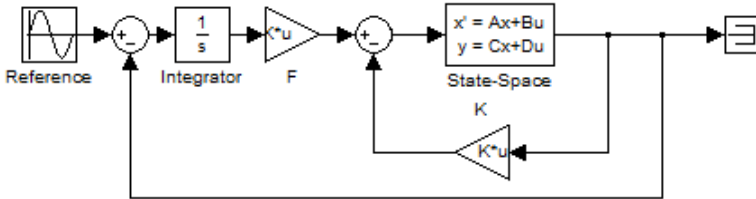


Figura 27 – Augmented PI scheme