Aircraft Dynamic Fuzzy Model Toolbox Documentation

Pablo Brusola

July 2017

Contents

1 Introduction

1.1 Introdution to Fuzzy Modeling

Nowadays, there is an important reliance on automatic control systems in aviation. Some of the critical phases of the flight, like the final approach, induce that many variables should be controlled simultaneously. Therefore, the improvement of control algorithms combined with navigation systems, directly contributes to have a secure flight. These results have a common source before being implemented: a reliable aircraft dynamic model.

The classic aircraft dynamic models are on the most part composed by 15 non-lineal differential equations, which are not the best option in order to adjust the control laws. The most common solution for this model identification, is by transfers functions methods, but there are also some inconvenient. These methods work due to a model linearisation from an equilibrium point, and for a normal flight, there are many different stages with one equilibrium point for each one. The outcome is a considerable number of different controllers.

This is the main fact for investigate a different dynamic model using a methodology based on fuzzy logic. These Fuzzy Models, have a considerable advantage in front of classic model, because it can be implemented one unified control system for all the model. They are composed by a combination of consequent and membership functions, which turn out into a sum of rules. Therefore this method is better than classic ones as it reduces computing costs.

The main objective will be the research of another version of conventional aircraft dynamic model, using Fuzzy Logic methods. For a better comprehension, the validation and examples of the model will be performed by the Airbus A310 aircraft values. This Toolbox is the result of another aircraft dynamic model-based on fuzzy logic.

1.2 About the Toolbox

This toolbox provide a general dynamic model based on fuzzy logic. There is two main fact on this:
Conversion to fuzzy model: With a FIS struct, the toolbox can calculate with the aircraft parameters, all the structure of the fuzzy logic. The main function to do this job is **NoLin2FIS**, where the model include all the non-lineal term of the classic aircraft dynamic model.
Defuzzyfication for Validation: This stage is important to compare the fuzzy model with the classic nonlineal model. The toolbox provide all the function to proliferate this.

1.3 Installation

The installation is straightforward and it does not require any changes to your system settings. Proceed along the following four steps:

Step 1: Create a subdirectory under your work directory, call it whatever you like.

Step 2: Copy the toolbox files to this directory. If the toolbox was provided as a zip file, use pkunzip to unpack it in this directory.

Step 3: Modify or create the Matlab startup.m file to include the Toolbox directory with the command addpath.

Step 4: Start Matlab and run your script working with the Toolbox.

2 Reference

2.1 Structures

2.1.1 param

The aerodynamic coefficients are governed by the following equations:

$$\begin{array}{rcl} C_L & = & C_{L0} + C_{L\alpha}\alpha(t) + \frac{\bar{c}}{V_a(t)}C_{Lq}q(t) + C_{L\delta_e}\delta_e + C_{LH}e^{-\lambda_L H_{LG}(t)} \\ C_Y & = & C_{Y\beta}\beta(t) + C_{Yr}\delta_r \\ C_D & = & C_{D0} + C_{D\alpha}\alpha(t) + C_{D\alpha_2}\alpha(t)^2 \end{array} \tag{1}$$

$$\begin{array}{rcl} C_l & = & C_{l\beta}\beta(t) + \frac{\bar{c}}{V_a(t)}(C_{lp}p(t) + (C_{lr0} + C_{lr\alpha}\alpha(t))r(t)) + C_{l\delta_a}\delta_a + C_{\delta_r}\delta_r \\ C_m & = & C_{m0} + C_{m\alpha}\alpha(t) + \frac{\bar{c}}{V_a(t)}C_{mq}q(t) + C_{m\delta_e}\delta_e + (C_{mH0} + C_{mH\alpha}\alpha(t))e^{-\lambda_m H_{LG}(t)} \\ C_n & = & (C_{n\beta_0} + C_{n\beta_\alpha}\alpha(t))\beta(t) + \frac{\bar{c}}{V_a(t)}(C_{nr}r + (C_{np0} + C_{np\alpha}\alpha(t))p(t)) + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \end{array} \tag{2}$$

Table 1: param.coef structure

Aerodynamic Coefficients (param.coef)					
.CL0	.CLa	.CLq	.CLh	.lambdal	.CLde
.CYb	.CYdr	.CD0	.CDa	.CDa2	.Clb
.Clp	.Clr0	.Clra	.Clda	.Cldr	.Cm0
.Cma	.Cmq	.Cmh0	.Cmha	.lambdam	.Cmde
.Cnb0	.Cnba	.Cnp0	.Cnpa	.Cnr	.Cnda
.Cndr					

Table 2: param.mig structure

Mass and Geometry (param.mig)	
.Sref	Reference surface, m^2
.Lref	Aircraft mean chord, m
.mass	aicraft mass, kg
.I	Aircraft matrix of inertia, kg/m^2
.J	Inverse of the matrix of inertia
.dxg	distance in x axis of he aerodynamic center, m
.dze	distance in z axis of the engines thrust, m

Table 3: param.atm struct

Mass and Geometry (param.mig)	
.To	Reference temperature, K
.rho	air density, kg/m^3

Table 4: param.eng structure

Engines (param.eng)	
.Ga	Coefficient of thrust equation
.Gb	Coefficient of thrust equation
.tau	time constant, s
.RL	rate limit

Table 5: param.act structure

Actuators (param.act)	
.tau	1x3 vector with time constants
.RL	1x3 vector with rate limits

Table 6: param.lim structure

Input limits (param.lim = [max,min])	
.a	angle-of-attack
.b	sideslip angle
.pVa	$\frac{p}{V_a}$
.qVa	$\frac{q}{V_a}$
.rVa	$\frac{r}{V_a}$
.Va2	V_a^2
.VazVax	$\frac{V_{az}}{V_{ax}}$
.VayVa	$\frac{V_{ay}}{V_a}$
.Hlg	Distance of the landing gear to the ground

2.1.2 FIS

This struct is composed by all the non-lineal terms of the classic aircraft dynamic model. The non-lineal terms are described on the following table 7. In each variable mentioned, there is always the same structure which define the fuzzy model.

Table 7: FIS. structure

Non-linear variables (FIS.)	
.a	angle-of-attack
.b	sideslip angle
.Va	airspeed
.CL1	first lift coefficient non-lineal term
.CL2	second lift coefficient non-lineal term
.CD2	first drag coefficient non-lineal term
.Cl1	first roll coefficient non-lineal term
.Cl2	second roll coefficient non-lineal term
.Cm1	first pitch coefficient non-lineal term
.Cm2	second pitch coefficient non-lineal term
.Cn1	first yaw coefficient non-lineal term
.Cn2	second yaw coefficient non-lineal term
.Cn3	third yaw coefficient non-lineal term

.mf: In this struct it will be defined the membership function where it gonna be an nx3 matrix where n are the different types of membership function matrix. The fisrt column of the matrix define the type of membership function. There are 5 types where:

$$\begin{array}{lcl} 1 & = & \text{atan(x) type} \\ 2 & = & \text{asin(x) type} \\ 3 & = & \sqrt{x} \text{ type} \\ 4 & = & \frac{x_1}{x_2} \\ 5 & = & e^{-x} \text{ type} \end{array}$$

The other two columns are the maximums and minimums of the membership function. If there are Global sector, this values are only use to plot the membsership function.

.c: This struct define the fuzzy model consequent for each rule and it is an rxm matrix where r are the number of rules. The firts column define if the consequent is 'constant' or if it is 'lineal'. On the first case, m=2 and the second column takes the constant value. For the second case, m=3 and the second and third column are a and b coefficients respectively of an $y = ax+b$ equation.

.r: This struct is an entire number which define the number of rules of the fuzzy model system.

2.2 Fuzzy Library

2.2.1 actuators.m

Purpose

This function returns the differential equation of the actuators dynamics of ailerons, elevators and rudder. They are been modelated as first-order differential equation.

Function

```
[ddeltadt] =actuators(t,delta,input_t,input_u,param)
```

Description

t	=	instant time value, sec
delta	=	1x3 vector with the actuators deflections as delta=[da,de,dr]
input_t	=	column time input vector, sec
input_u	=	column actuators deflection input vector
param	=	struct with actuators parameters

For this function, param should be defined by the time constant and rates limits of the actuators dynamics as param.act.tau and param.act.RL respectively. This function will return the derivative of the actuators with respect the time in 1x3 vector $[\frac{dda}{dt}, \frac{dde}{dt}, \frac{ddr}{dt}]$

See Also

param, engines.m, dFuzzyModeldt.m

2.2.2 calcc.m

Purpose

This function calculate the consequent of a fuzzy model with the function input. The consequents can be constants or lineal, it depends of the fuzzy model consequent matrix.

Function

```
[ C ] = calcc(c,u,n)
```

Description

c	=	Consequent Matrix of the FIS struct
u	=	input values of the model system, is a column vector
n	=	Number of rules (number of consequents), include in the FIS struct.

The function output is the calculated consequent of the model included in the FIS struct. The output C is a nxm matrix where n has the same length that the input vector u and m is the number of rules.

See Also

FIS, calcfuzzy.m, calcmf.m

2.2.3 calcfuzzy.m

Purpose

This function calculate the defuzyfication using the FIS struct model with the membership functions and consequent matrix information.

Function

```
[ yout ] = calcfuzzy(fisvar,x,u)
```

Description

fisvar	=	FIS struct of the variable model with mfs and c matrix struct
x	=	Input variables of the membership functions
u	=	Input variables of the consequent functions (optional).

The function output will be the defuzzyfication of the system model. It uses the functions calcc.m and calcmf to calculate consequent and membership function respectively. The input of the consequent are optional for the reason that in many times, there are constant.

See Also

FIS, calcc.m, calcmf.m

2.2.4 calcmf.m

Purpose

This function returns the value of each membership function of the model introducing the input variable of the membership function.

Function

```
[ M ] = calcmf(mf,x,n)
```

Description

mf	=	membership matrix of the FIS struct
x	=	membership input
n	=	number of membership types

The function output will be the membership function values where it is an struct for each membership function type. Each strut will have two columns for the reason of the pairs of the memberships function into each type.

See Also

FIS, calcc.m, calcfuzzy.m, plotmf.m

2.2.5 DefaultMaxMin.m

Purpose

With this function, the intention is to set defaults maximums and minimums values for the fuzzy model system when the user has not defined it.

Function

```
[ param ] = DefaultMaxMin( param )
```

Description

param = aircraft parameters values.

The output will be an actualisation of the input param. The function has the objective of read all the param struct searching the limits and if they are not defined, it set the default values.

See Also

param, NoLin2Fis.m

2.2.6 dFuzzyModeldt.m

Purpose

This function joins all the aircraft dynamic model with the differential equations based on fuzzy models of the non-lineal terms and based on quaternions.

Function

```
[ dYoutdt ] =dFuzzyModeldt(t,Y,input_t,input_u,param)
```

Description

t	=	instant time (seconds)
Y	=	initial conditions of the variables states [OM, V, Quat, X]
input_t	=	time vector column of system inputs, sec
input_u	=	values vector column of the system inputs
param	=	aircraft parameters values, on this case it should be included param.FIS with the FIS struct of the aircraft.
OM	=	angular rates vector, rad/s
V	=	aircraft velocity vector, m/s
Quat	=	quaternions vector
X	=	aircraft position vector, m

The output of this function will be the derivative of the states variables of the system as dYoutdt = $[\frac{dOM}{dt}, \frac{dV}{dt}, \frac{dQuat}{dt}, \frac{dX}{dt}]$, and it will be an 1x13 vector.

See Also

param, dNoLindt.m, NoLin2FIS

2.2.7 dNoLindt.m

Purpose

This function joins all the differential equations of the classic aircraft dynamic model. Its structure is similar to the dFuzzyModeldt.m structure.

Function

```
[ dYoutdt] = dNoLindt(t,Y,input_t,input_u,param)
```

Description

t	=	instant time (seconds)
Y	=	initial conditions of the variables states [OM, V, PHI, X]
input_t	=	time vector column of system inputs, sec
input_u	=	values vector column of the system inputs
param	=	aircraft parameters values, it is not necesary to include param.FIS
OM	=	angular rates vector, rad/s
V	=	aircraft velocity vector, m/s
PHI	=	euler angles, rad
X	=	aircraft position vector, m

The output of this function will be the derivative of the states variables of the system as dYoutdt = $[\frac{dOM}{dt}, \frac{dV}{dt}, \frac{dPHI}{dt}, \frac{dX}{dt}]$, and it will be an 1x12 vector.

See Also

param, dFuzzyModeldt.m, NoLin2FIS

2.2.8 engines.m

Purpose

This function returns the differential equation of the engines dynamics of the exhaust pressure ratio. It has been modelated as first-order differential equation.

Function

```
[ deprdt ] = engines( t,epr,input_t,input_u,param )
```

Description

t	=	instant time value, sec
epr	=	Exhaust pressure ratio, initial condition
input_t	=	column time input vector, sec
input_u	=	column EPR input vector
param	=	struct with engine parameters

For this function, param should be defined by the time constant and rates limits of the engines dynamics as param.eng.tau and param.eng.RL respectively. This function will return the derivative of the EPR with respect the time.

See Also

param, actuators.m, dFuzzyModeldt.m

2.2.9 mfplot.m

Purpose

This function plot the input membership function with the FIS.mfs matrix structure. It recognize the set limits in the conversion to FIS from non-linear.

Function

```
mfplot( mf,varname )
```

Description

mf	=	membership function matrix.
varname	=	name of the variable of the fuzzy model (optional)

Introducing the membership function matrix struct from a variable of the FIS, it returns one figure with one or two plots, it depends of the number of membership function types of the variable.

See Also

FIS, NoLin2FIS.m

2.2.10 NoLin2FIS.m

Purpose

This function gives the FIS struct including all the information of the fuzzy models merged from the aircraft specifications. FIS gives the information of the nonlinear terms of the classic model that it has been converted on fuzzy model.

Function

```
[ FIS ] = NoLin2FIS(param)
```

Description

param = aircraft parameters values.

As aircraft parameter values inputs, this function calculates al the membership functions and consequents of the fuzzy model for each non-lineal term of the aircraft dynamic system. It returns a FIS struct.

See Also

param, FIS, calcfuzzy.m

3 A310 example

For a better comprehension of the fuzzy model toolbox, it is proposed the fuzzy modelling, the defuzzyfication and the validation of the Airbus A310 aircraft dynamic model. For this reason, it gonna be include some functions to proliferate the demonstration. All of it is include in the script DemoA310.m.

In this demo, first it begins calculating the trim values for the initial conditions that they should be an input of the ode's solvers. There is a SolveODes path that includes some solvers for the differential equations. In this case it gonna be used ode1 that it is a simple solver using Euler's method. After calculate the initial conditions, it initialize the parameters as a function ParamsInput. This function only create the param struct explained before. Next, it is time to define the inputs values, the actuators and engines in front of the time. It is important for simulate to know the actuators of the aircraft. It follows as the fuzzy modelling, with all the aircraft parameters it creates the FIS struct including the fuzzy model of the non-lineal terms. Them it can be calculated the response of the non-lineal dynamic and the fuzzy model dynamic of the aircraft.

Them on the figure 1 it can be observed one of the validation variable state parameter. Also it has been ploted the membership functions of each variable as it shows the illustration 2.

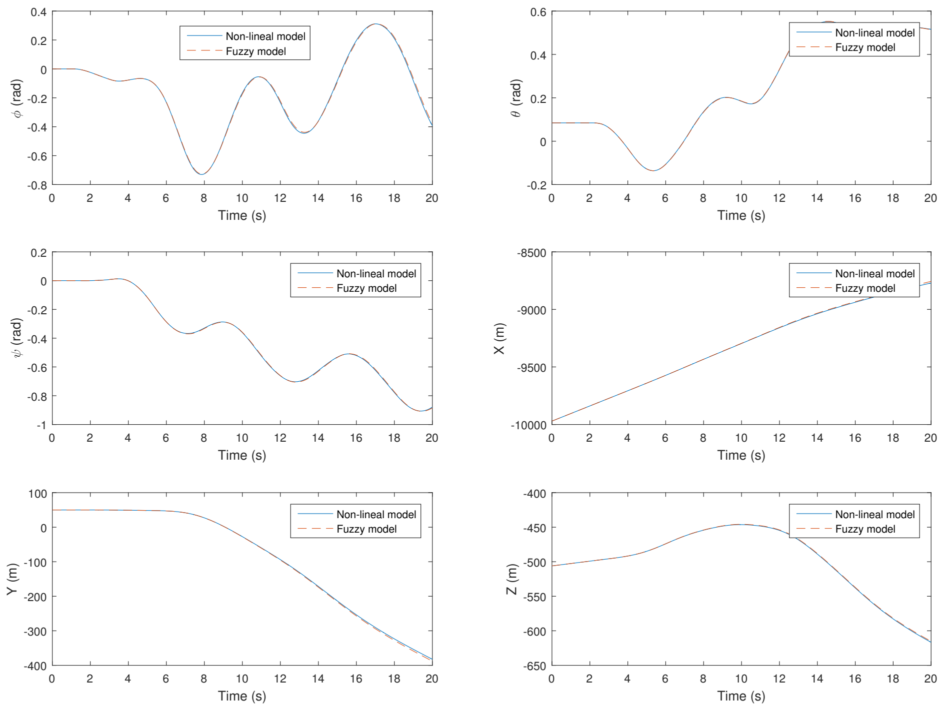

Figure 1: Validation of the system state variables

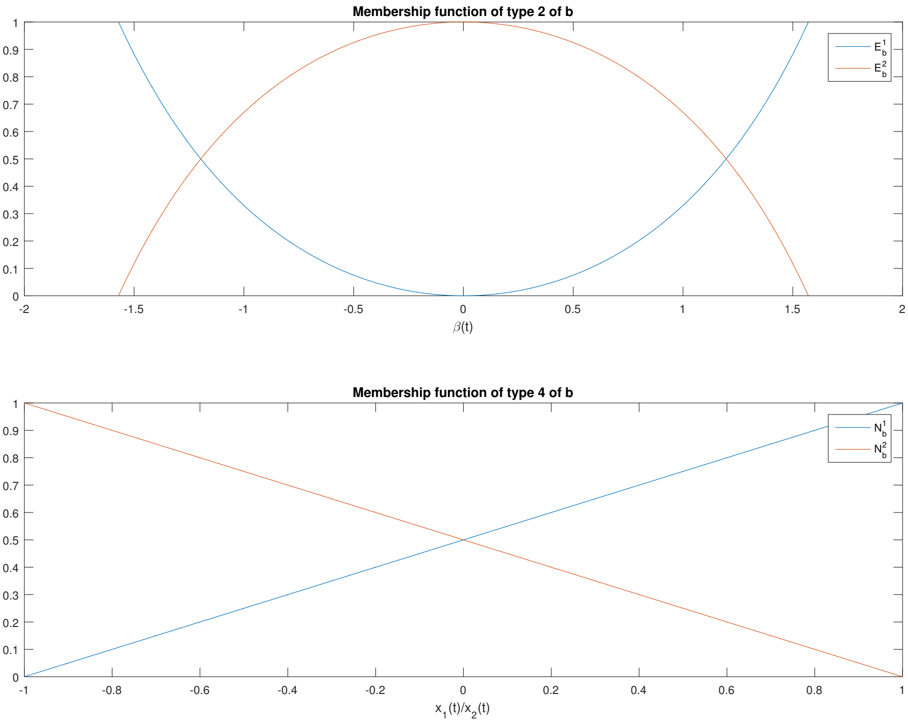

Figure 2: Example of the membership function plot