I. Interval Type-2 FLS

A general Type-2 FLS is shown at Figure 1. It is composed of fuzzifier, rules, inference, type-reducer and defuzzifier, being necessary to have the presence of, at least, one Type-2 Fuzzy Set in one of the antecedents or in the consequent, which compose one of the rules that form the system. The specific case of Interval Type-2 FLS happens when all of the antecedents and consequent are Type-2 Fuzzy Sets, being this approach used in this work.

A. Input Variables

It was adopted for the controller the horizontal distance (dX and dY) between the UAV and the landing spot in the X axis and Y axis of UAV body frame, respectively. Figure 2 shows the membership functions for dX (dY have similar membership functions) which are divided into three fuzzy trapezoidal groups: negative (red), close (blue) and positive (green). The variables indicate velocity values between -100% and 100%, relative to the field of view of the camera used. The Equations (1) - (6) represent them listed above, where NS means negative superior, NI means negative inferior, CS means close superior, CI means close inferior, PS means positive superior and PI means positive inferior. The other shapes in fuzzy sets are outside the scope of this paper.

$$\mu_{NS}(dX) = \begin{cases} 1 & \text{if } dX \leq -10, \\ -0.1dX & \text{if } -10 \leq dX \leq 0, \\ 0 & \text{if } dX \geq 0, \end{cases} \tag{1}$$

$$\mu_{CS}(dX) = \begin{cases} 0 & \text{if } dX \leq -15, \\ 0.1dX + 1.5 & \text{if } -15 \leq dX \leq -5, \\ 1 & \text{if } -5 \leq dX \leq 5, \\ -0.1dX + 1.5 & \text{if } 5 \leq dX \leq 15, \\ 0 & \text{if } dX \geq 15, \end{cases} \tag{2}$$

$$\mu_{PS}(dX) = \begin{cases} 0 & \text{if } dX \leq 0, \\ 0.1dX & \text{if } 0 \leq dX \leq 10, \\ 1 & \text{if } dX \geq 10, \end{cases} \tag{3}$$

$$\mu_{NI}(dX) = \begin{cases} 0.5 & \text{if } dX \leq -15, \\ -0.1dX - 1 & \text{if } -15 \leq dX \leq -10, \\ 0 & \text{if } dX \geq -10, \end{cases} \tag{4}$$

$$\mu_{CI}(dX) = \begin{cases} 0 & \text{if } dX \leq -5, \\ 0.1dX + 0.5 & \text{if } -5 \leq dX \leq 0, \\ -0.1dX + 0.5 & \text{if } 0 \leq dX \leq 5, \\ 0 & \text{if } dX \geq 5, \end{cases} \tag{5}$$

$$\mu_{PI}(dX) = \begin{cases} 0 & \text{if } dX \leq 10, \\ 0.1dX - 1 & \text{if } 10 \leq dX \leq 15, \\ 0.5 & \text{if } dX \geq 15, \end{cases} \tag{6}$$

B. Output Variables

It was adopted for the controller the linear velocity (vX, vY and vZ) in the UAV frame. Figure 3 shows the membership functions for vX (vY and vZ have similar membership functions) which are divided into three fuzzy trapezoidal groups: negative (red), zero (blue) and positive (green). The variables indicate velocity values between -100% and 100%, which will later be denormalized for actuation on the plant, according to an established maximum velocity of the UAV. The Equations (7) - (12) represent them listed above, where NS means negative superior, NI means negative inferior, ZS means zero superior, ZI means zero inferior, PS means positive superior and PI means positive inferior. The other shapes in fuzzy sets are outside the scope of this paper.

$$NS = \int_{-15}^{0} \frac{vX * \mu_{NS}(dX)}{\mu_{NS}(dX)} \tag{7}$$

$$ZS = \int_{-15}^{15} \frac{vX * \mu_{CS}(dX)}{\mu_{CS}(dX)} \tag{8}$$

$$PS = \int_{0}^{15} \frac{vX * \mu_{PS}(dX)}{\mu_{PS}(dX)} \tag{9}$$

$$NI = \int_{-15}^{0} \frac{vX * \mu_{NI}(dX)}{\mu_{NI}(dX)} \tag{10}$$

$$ZI = \int_{-15}^{15} \frac{vX * \mu_{CI}(dX)}{\mu_{CI}(dX)} \tag{11}$$

$$PI = \int_{0}^{15} \frac{vX * \mu_{PI}(dX)}{\mu_{PI}(dX)} \tag{12}$$

C. Fuzzifier

This module has the function of transforming crisp inputs (dX and dY) in Type-2 Fuzzy Sets. For this work singleton fuzzifier was used.

D. Rules

This module has the function of describing the relationship between linguistic variables, defining system's performance and behavior. Four experts worked for the build of Type-2 fuzzy sets. Its format is "if dX is ... and dY is ..., so vX (or vY, or vZ) is ...". The Tables I, II and III represent rules for vX, vY and vZ, respectively. As the combination of dX and dY includes the descent movement, the variable dZ did not participate on the rule base, being responsible for forcing the landing in a predetermined height.

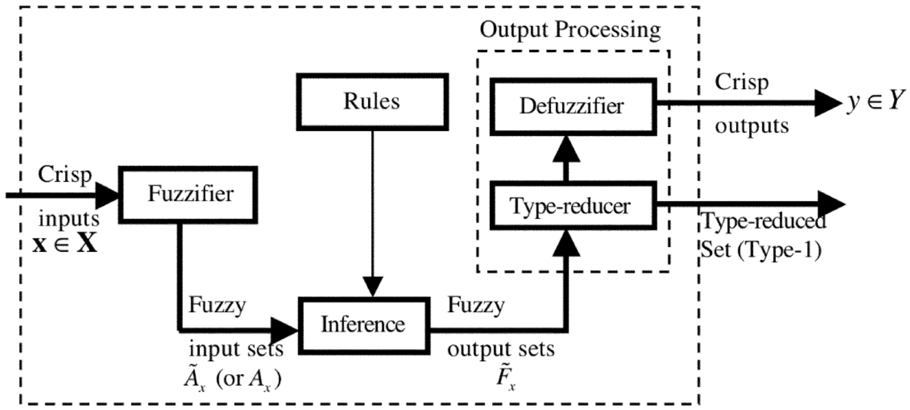

Fig. 1. Type-2 FLS [7]

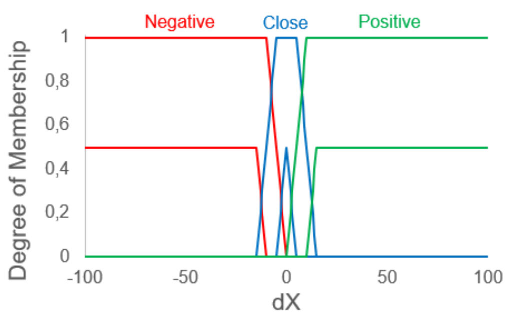

Fig. 2. Input variable

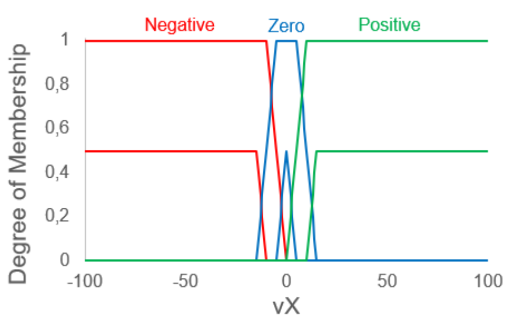

Fig. 3. Output Variable

E. Inference

This module has the function of processing mathematically each preposition of rules through approximate reasoning techniques, providing output from input combination. Considering the design of the FLS, it was adopted the minimum operator as the implication method. The Equations (13) - (39) represent the superior, inferior and medium activation degrees for each one of the nine rules referring to vX.

$$f_1^S = min[\mu_{NS}(dX), \mu_{NS}(dY)] \tag{13}$$

$$f_1^I = min[\mu_{NI}(dX), \mu_{NI}(dY)] \tag{14}$$

TABLE I
vX RULES

vX rules	Input		Output
Rule Number	dX	dY	vX
r1	Negative	Negative	Negative
r2	Negative	Close	Negative
r3	Negative	Positive	Negative
r4	Close	Negative	Zero
r5	Close	Close	Zero
r6	Close	Positive	Zero
r7	Positive	Negative	Positive
r8	Positive	Close	Positive
r9	Positive	Negative	Positive

TABLE II
vY RULES

vY rules	Input		Output
Rule Number	dX	dY	vY
r10	Positive	Negative	Negative
r11	Negative	Negative	Negative
r12	Close	Negative	Negative
r13	Positive	Close	Zero
r14	Negative	Close	Zero
r15	Close	Close	Zero
r16	Positive	Positive	Positive
r17	Negative	Positive	Positive
r18	Close	Positive	Positive

$$f_1 = \frac{f_1^S + f_1^I}{2} \tag{15}$$

$$f_2^S = min[\mu_{NS}(dX), \mu_{CS}(dY)] \tag{16}$$

$$f_2^I = min[\mu_{NI}(dX), \mu_{CI}(dY)] \tag{17}$$

$$f_2 = \frac{f_2^S + f_2^I}{2} \tag{18}$$

TABLE III
VZ RULES

vZ rules	Input		Output
Rule Number	dX	dY	vZ
r19	Close	Close	Negative
r20	Positive	Negative	Zero
r21	Positive	Close	Zero
r22	Positive	Positive	Zero
r23	Negative	Negative	Zero
r24	Negative	Close	Zero
r25	Negative	Positive	Zero
r26	Close	Negative	Zero
r27	Close	Positive	Zero

$$f_3^S = min[\mu_{NS}(dX), \mu_{PS}(dY)] \tag{19}$$

$$f_3^I = min[\mu_{NI}(dX), \mu_{PI}(dY)] \tag{20}$$

$$f_3 = \frac{f_3^S + f_3^I}{2} \tag{21}$$

$$f_4^S = min[\mu_{CS}(dX), \mu_{NS}(dY)] \tag{22}$$

$$f_4^I = min[\mu_{CI}(dX), \mu_{NI}(dY)] \tag{23}$$

$$f_4 = \frac{f_4^S + f_4^I}{2} \tag{24}$$

$$f_5^S = min[\mu_{CS}(dX), \mu_{CS}(dY)] \tag{25}$$

$$f_5^I = min[\mu_{CI}(dX), \mu_{CI}(dY)] \tag{26}$$

$$f_5 = \frac{f_5^S + f_5^I}{2} \tag{27}$$

$$f_6^S = min[\mu_{CS}(dX), \mu_{PS}(dY)] \tag{28}$$

$$f_6^I = min[\mu_{CI}(dX), \mu_{PI}(dY)] \tag{29}$$

$$f_6 = \frac{f_6^S + f_6^I}{2} \tag{30}$$

$$f_7^S = min[\mu_{PS}(dX), \mu_{NS}(dY)] \tag{31}$$

$$f_7^I = min[\mu_{PI}(dX), \mu_{NI}(dY)] \tag{32}$$

$$f_7 = \frac{f_7^S + f_7^I}{2} \tag{33}$$

$$f_8^S = min[\mu_{PS}(dX), \mu_{CS}(dY)] \tag{34}$$

$$f_8^I = min[\mu_{PI}(dX), \mu_{CI}(dY)] \tag{35}$$

$$f_8 = \frac{f_8^S + f_8^I}{2} \tag{36}$$

$$f_9^S = min[\mu_{PS}(dX), \mu_{NS}(dY)] \tag{37}$$

$$f_9^I = min[\mu_{PI}(dX), \mu_{NI}(dY)] \tag{38}$$

$$f_9 = \frac{f_9^S + f_9^I}{2} \tag{39}$$

F. Type-reducer

This module has the function of finding a Type-1 Fuzzy Set that better represents Type-2 Fuzzy Set, in such a way that the in the absense of uncertainty, the results of Type-2 FLS are reduced to Type-1 FLS [**?**]. For that, Karnik Mendel (KM) Algorithm [**?**] that can be executed on paralel and are monotonically and exponencially convergent is used.

The Equations (40) - (42) represents the KM Algorithm to obtaining the pertinence resultant functions to the left (L). The Equation (40) represents the vX^I initially estimated, whose result is refined through Equations (41) or (42), depending on the interval in which vX^I is placed. The variable vX^I receives this new value and this iterative process repeats until a new value of vX^I becomes equal to the previous one.

$$vX^I = \frac{NI\sum_{i=1}^3 f_i + ZI\sum_{j=4}^6 f_j + PI\sum_{k=7}^9 f_k}{\sum_{i=1}^3 f_i + \sum_{j=4}^6 f_j + \sum_{k=7}^9 f_k} \tag{40}$$

if NI $\leq vX^I \leq$ ZI:

$$vX^I = \frac{NI\sum_{i=1}^3 f_i^S + ZI\sum_{j=4}^6 f_j^I + PI\sum_{k=7}^9 f_k^I}{\sum_{i=1}^3 f_i^S + \sum_{j=4}^6 f_j^I + \sum_{k=7}^9 f_k^I} \tag{41}$$

if ZI $\leq vX^I \leq$ PI:

$$vX^I = \frac{NI\sum_{i=1}^3 f_i^S + ZI\sum_{j=4}^6 f_j^S + PI\sum_{k=7}^9 f_k^I}{\sum_{i=1}^3 f_i^S + \sum_{j=4}^6 f_j^S + \sum_{k=7}^9 f_k^I} \tag{42}$$

The Equations (43) - (45) represents the KM Algorithm to obtaining the pertinence resultant functions to the right (R). The Equation (43) represents the vX^S initially estimated, whose result is refined through Equations (44) or (45), depending on the interval in which vX^S is placed. The variable vX^S receives this new value and this iterative process repeats until a new value of vX^S becomes equal to the previous one.

$$vX^S = \frac{NS\sum_{i=1}^3 f_i + ZS\sum_{j=4}^6 f_j + PS\sum_{k=7}^9 f_k}{\sum_{i=1}^3 f_i + \sum_{j=4}^6 f_j + \sum_{k=7}^9 f_k} \tag{43}$$

if NS $\leq vX^S \leq$ ZS:

$$vX^S = \frac{NS\sum_{i=1}^3 f_i^I + ZS\sum_{j=4}^6 f_j^S + PS\sum_{k=7}^9 f_k^S}{\sum_{i=1}^3 f_i^I + \sum_{j=4}^6 f_j^S + \sum_{k=7}^9 f_k^S} \tag{44}$$

if ZS $\leq vX^S \leq$ PS:

$$vX^S = \frac{NS\sum_{i=1}^{3} f_i^I + ZS\sum_{j=4}^{6} f_j^I + PS\sum_{k=7}^{9} f_k^S}{\sum_{i=1}^{3} f_i^I + \sum_{j=4}^{6} f_j^I + \sum_{k=7}^{9} f_k^S} \quad (45)$$

G. Deffuzification

This module has the function of transforming Type-1 Fuzzy Sets in crisp output (vX, vY and vZ). For the present work, centroid defuzzification was used. The Equation (46) represents this transformation.

$$vX = \frac{vX^I + vX^S}{2} \quad (46)$$

The step by step for calculating vY and vZ is the same as for vX.