ISSN: 1231-4005
e-ISSN: 2354-0133
DOI: 10.5604/01.3001.0010.2810

INITIAL STATE EVALUATION ALGORITHM USING FUZZY BELLMAN METHOD

Jan Gruszecki, Fatina Liliana Basmadji

Rzeszow University of Technology
Department of Avionics and Control Systems
Powstancow Warszawy Av. 8, 35-959 Rzeszow, Poland
tel.: +48 17 8651783, fax: +48 17 8544319, e-mail: awionjgr@prz.edu.pl, basmadji@prz.edu.pl

Abstract

The problem of UAV control along a predefined trajectory is a well know problem and there are several methods that could accomplish this task. However, adding final conditions that the UAV have to meet at the end of the trajectory makes this problem more difficult. In this article, the authors present a fuzzy based method that is not time consuming and can solve this problem.

The algorithm for flying at low altitudes are aimed to ensure the maximum safety of the flight by considering only the acceleration values that are possible to achieve by the object. In particular, the article presents: flight trajectory over mountainous terrain, the structure of altitude control system, the structure of normal acceleration control system, exemplary calculations and an object with velocity equals, membership functions of angle of attack, solutions' illustration, flight path angle and pitch angle along the trajectory, the demanded and actual normal acceleration along the trajectory and flying object trajectory.

Keywords: *UAV, guidance, trajectory, mountainous terrain*

1. Introduction

The control of a flying object in different phases of flight and while flying over a mountainous terrain can be performed using different algorithms. The choice of algorithms depends on the type of task and its assumptions. Algorithms that use static optimization methods, variational methods or Bellman method usually require long execution time, especially when then final conditions imposed on the trajectory are defined. In the process of flying object control over a mountainous terrain it is demand to perform flight along the previously developed trajectory. This trajectory is generated by the pilot (pilot's assistant) of the manned aerial vehicle or the operator of the unmanned system in the Ground Control Station considering several aspects:

- constant flight altitude over the terrain,
- performing a flight along initially defined trajectory,
- performing a flight taking into consideration the following constraints: height above the terrain must be bigger than $\Delta H_{min} = z_0$, and the flight path $\gamma(x) \in [\gamma_{Terenu\ min}, \gamma_{Terenu\ max}]$.

Such a flight, as mentioned in [4], could be performed using various algorithms. Some of them are:

- classical algorithms of continuous minimization of the distance error from the trajectory or to minimize the sum of squared distances in the selected points of the trajectory,
- algorithm generated by the LQR method,
- Bellman discrete method,
- fuzzy algorithm [5].

Choosing the best control algorithm [1, 2, 4], especially when it is required to achieve defined final conditions at t_k, led the authors to develop algorithms that allow an effective estimation of control signals along the trajectory. This article presents the last two methods that are adapted to control the object while flying over mountainous terrain with a given final condition at the final time t_k (Fig. 1).

The control of various objects (such as aircrafts with classical control system, unmanned air vehicles) typically uses two control variables: altitude h(t) and normal acceleration $a_z(t)$. The difference is only in the different system structure (Fig. 2 and 3). In this article, the scheme shown in Fig. 3 will be used.

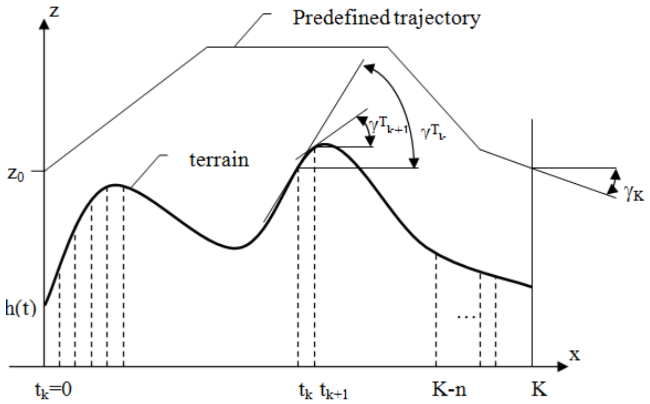

Fig. 1. Flight trajectory over mountainous terrain

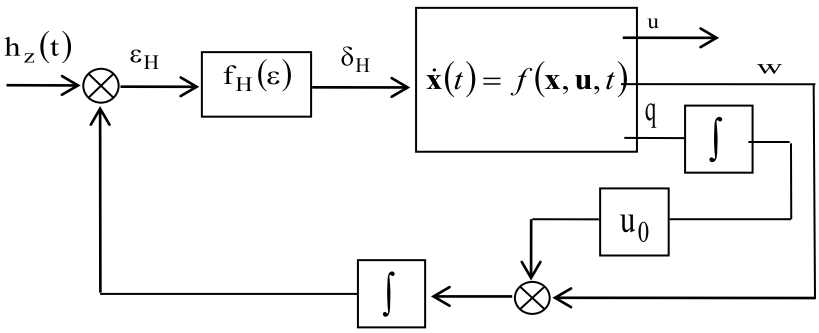

Fig. 2. The structure of altitude control system

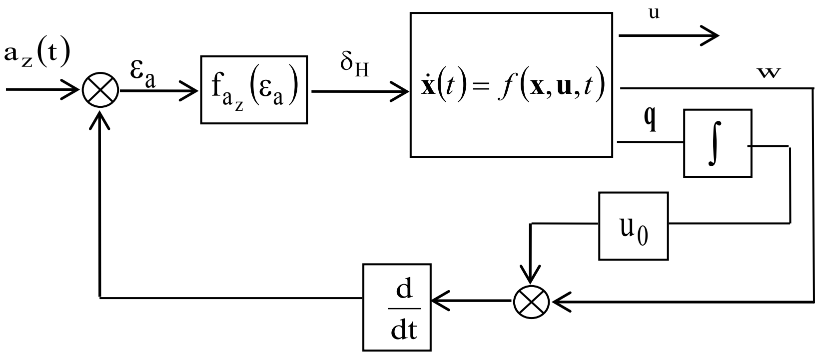

Fig. 3. The structure of normal acceleration control system

2. Discrete Bellman method based algorithm

Let the longitudinal motion of the flying object be described by the following equations:

$$x(t) = f\,(\,x(t),\, a_z(t),\, t\,),$$

$$x(t) = [\,u(t),\, w(t),\, q(t),\, \Theta(t)\,]^T, \tag{1}$$

where:

u (t) – is the change in horizontal velocity,

w (t) – is the change in vertical velocity,
q (t) – is the change in pitch angular velocity,
Θ (t) – is the change in pitch angle,
with the final condition:

$$x(t_k) = x_k. \tag{2}$$

Terrain configuration given as a numerical terrain model or as a set of height values along the flight path x_E in an inertial coordinate system could be defined as z(l(t)),

where: $l(t)=\int_{t_0}^{t_K} u(t)dt$;

U (t) is the flight velocity.

The most rational criterion function could be defined as following:

$$\mathcal{J} = \min_{\delta_H(t)\in\Delta} \int \left[\Gamma(l(t)) - \gamma(l(t))\right]^2 dt, \tag{3}$$

with the constraint:

$$h(t) \geq z(l(t) + z_0, \tag{4}$$

where:
l (t) – x-axis coordinate in the inertial coordinate system,
z (l (t)) – terrain elevation along the x-axis,
z0 – minimal height margin over terrain,
Γ – terrain slope along the x-axis,
γ (t) – flight path angle,
h (t) – x-axis coordinate of the centre of gravity in the inertial coordinate system.

Using Bellman's theorem [6] criterion function on the last trajectory stage l(t), i.e. $[l(t_{k-1}), l(t_k)]$ is equal to:

$$\mathcal{J}_{K-1}\left(\delta_{H_{K-1}}\right) = \left[\Gamma(l(t_{K-1})) - \gamma(l(t_{K-1}))\right]^2, \tag{5}$$

and reaches a minimum for $a_z(t_{K-1})$ those correspondents to δ_{HK-1}. In general, for the whole trajectory l(t), t є (t_0, t_K), we have 2K equations obtained from the condition of:

- criterion function minimization:

$$\mathcal{J} = \sum_{k=0}^{K-1}\left[\Gamma(l(t_k)) - \gamma(l(t_k))\right]^2, \tag{6}$$

thus:

$$\frac{\partial \mathcal{J}}{\partial \delta_H} = 0, \ \delta_H = \left[\delta_{H0},\ldots,\delta_{HK-1}\right]^T, \tag{7}$$

- Linearization of the state equation (1) to the form of:

$$\dot{\mathbf{x}}_{k+1} = \mathbf{x}_k + f(\mathbf{x}(t_k), \mathbf{u}(t_k), t_k) \cdot \Delta t_k, \tag{8}$$

where:
Δt_k – time of flight from the t_k to t_{k+1} with velocity equals to U_k.
This problem is usually solved making some simplifications resulting from the linearization of the equation (1) for a determined flight velocity U_0 and a constant average of height H_0.

3. Bellman method based fuzzy algorithm

Procedure given in section 2 gives good results, but requires complex calculations related to the problem of solving 2K equations for state variables at t_k; k = 0,1, ... K-1, and controls at the same points of time. Another attempt to solve this problem, assuming that this problem is not deterministic, is to use fuzzy algorithm based on Bellman method.
Like in the case of deterministic problem, we assume that the state of the object is given by a discrete equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{9}$$

where:
x is the state space, $x = \{x_1, \ldots, x_N\} \subset X$,
a_z is the controls space, $a_z = \{s_1, \ldots sM\} \subset \mathcal{A}_z$

Fuzzy state in k-stage is defined as a fuzzy set x_k in X whose membership function is equal to $\mu_{\mathbf{x}_k}(\mathbf{x}_k)$. In the case of fuzzy control a_{zk} (vector a_{zk} was limited to a single scalar value a_z) will be determined in A_z. Its membership function is equal to $\mu_{az_k}(az_k)$. The problem of flying along predefined trajectory from t_0 to t_K had been solved in [1].

In [1] the equations determining the change of flying object state variables were derived, respectively:

- Flight path angle membership:

$$\mu_{\gamma_{k+1}}(\gamma_{k+1}) = \max_{\theta_k \in \theta^*}[\mu_{\theta_k}(\theta_k) \wedge \mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta)], \tag{10}$$

where:

$$\mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta) = \begin{bmatrix} \mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta_1) \\ \mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta_2) \\ \cdots \\ \mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta_n) \end{bmatrix},$$

$$\mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta_i) = \max_{\gamma_k \in \gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\gamma_{k+1}}(\gamma_{k+1} \mid \gamma_k, a_k)].$$

Whereas $\mu_{\gamma_{k+1}}(\gamma_{k+1} \mid \gamma_k, a_k)$ presents flight path angle transformation matrix from γ_k to γ_{k+1} for a given pitch angle $\theta_k = \theta_i$ and for given control value a_z.

- The membership function of the pitch angle is equal to:

$$\mu_{\theta_{k+1}}(\theta_{k+1}) = \max_{\gamma_k \in \gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\theta_{k+1}}(\theta_{k+1}, \gamma)]. \tag{11}$$

- Altitude membership function:

$$\mu_{\Delta h_{k+1}}(\Delta h_{k+1}) = \max_{\gamma_k \in \gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\Delta h_{k+1}}(\Delta h_{k+1}, \gamma)]. \tag{12}$$

- Pitch rate membership function:

$$\mu_{q_{k+1}}(q_{k+1}) = \max_{\theta_k \in \theta^*}[\mu_{\theta_k}(\theta_k) \wedge \mu_{q_{k+1}}(q_{k+1}, \theta)]. \tag{13}$$

When generating a transformation matrix it was assumed that the pitch rate is zero. This is true for the first stage where the object is in steady flight. However, for the n-stage trajectory the angular velocity at the beginning of each next stage is not necessarily equal to zero. Therefore, the following correction for the angular velocity was used:

$$q = \dot{\theta} \quad \rightarrow \quad q = \frac{\theta_{k+1} - \theta_k}{\Delta t} \quad \rightarrow \quad \theta_{k+1} = q \cdot \Delta t + \theta_k\,, \tag{14}$$

assuming that for small distances Δx the flight path is a line and the time in which the object perform this flight could be calculated from:

$$\Delta t = \frac{\Delta x}{Uo \cdot \cos\gamma}\,, \tag{15}$$

where Uo – object velocity.

Having Θ_k, γ_k i H_k it is easy to calculate their values in the next stage using the transformation matrices [1]. Whereas finding the value of γ_k and Θ_k having the values of γ_{k+1} and Θ_{k+1} with are no longer easy. This is due to the fact that the transformation matrix is singular. In this article, two possible algorithms will be presented: static optimization and algorithms that uses the properties of fuzzy logic.

3.1. Static optimization method

Let us consider a task formulated in the same way as (1) to (5). However, using the discrete Bellman approach it is enough to solve the inverse problem from stage (k + 1) to k, assuming that at each segment (k, k + 1) the following minimum is achieved:

$$\min \mathcal{J}(s_m) = \min_{s_m \in \mathcal{A}_z} \{\Gamma(l(t_k)) - \gamma(l(t_k))\}\,;\ \mathrm{m} = 1,\ldots\mathrm{M}, \tag{16}$$

with the constraint:

$$\mathrm{h}(\mathrm{t_k}) \geq \mathrm{z}(\mathrm{l}(\mathrm{t_k})) + \mathrm{z_0}\,, \tag{17}$$

while as for t_K we have

$$\gamma(\mathrm{l}(\mathrm{t_K})) = \gamma_z(\mathrm{t_K})\,. \tag{18}$$

Finding $s_m(t_k)$ is difficult due to the fact that the function $\mathcal{J}(s_m)$ do not have continuous first derivatives, and the estimation of the value $s_m(t_k)$ is realized for discrete values of s_m adopted with a large discretization step in $\mathcal{A}_z$ (due to the rule base size constraint). From many methods of simple searching [3], good results were obtained by the authors by modifying the method of Hook and Jeeves.

While maintaining the principle of movement in each stage: trial step and work step, we analyse the behaviour of the function $\mathcal{J}(s_m)$ in a limited area of $\Delta\Theta_k$ and $\Delta\gamma_k$ at stage k.

The modification of the method at the work step will depend on taking into account the physical relationship that exists between Θ and γ:

$$\gamma(\mathrm{t_k}) = \chi\Theta(\mathrm{t_k}) \tag{19}$$

and considering the initial length of the step as τ, the values of the variables $\Theta(t_{k+1})$ and $\gamma(t_{k+1})$ will be equal to:

$$\gamma(\mathrm{t_{k+1}}) = \gamma(\mathrm{t_k}) + \tau\chi\nabla_\gamma\mathcal{J}(\mathrm{a_z}(\mathrm{t_k}))\,, \tag{20}$$

$$\Theta(\mathrm{t_{k+1}}) = \Theta(\mathrm{t_k}) + \tau\nabla_\Theta\mathcal{J}(\mathrm{a_z}(\mathrm{t_k}))\,. \tag{21}$$

The process of searching starts from γ^0 and θ^0 for possibly large of τ values. As the values of $\mathcal{J}(s_m)$ decrease, we decrease the value of $\tau = \beta\tau$. The values of $\gamma(t_k)$ and $\Theta(t_k)$ that minimize (6) correspond to the discrete values of $s_m \in \mathcal{A}_z$ for which the rules base was built.

Finding the value of $s_m(t_k)$ from a set of real numbers for which the condition (17) is achieved is possible only through approximation (usually a linear approximation) (Fig. 4):

$$s_m(t_k) = s_m^2 + \frac{\left[\gamma_k(s_m^2) - \Gamma(l(t_k))\right](s_m^2 - s_m^L) + \left[\gamma_k(s_m^2) - \gamma_k(s_m^2)\mathcal{J}s_m^2\right]}{\left[\gamma_k(s_m^2) - \gamma_k(s_m^2)\right]s_m^2}. \tag{22}$$

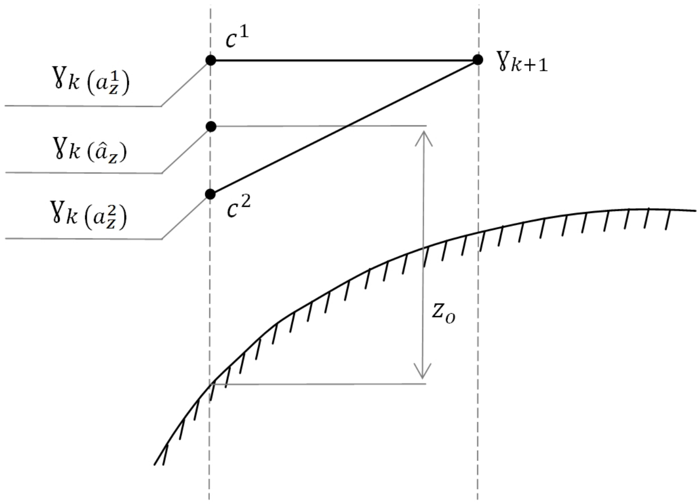

Fig. 4. Choosing s_m at the time k

An exemplary calculation using the inverse fuzzy algorithm is given in Tab. 1.

Tab. 1. Exemplary calculations of γ_k for γ_{k+1} and an object with velocity equals to $U_0 = 1000\, m/s$.

Step Number	γ_k	Θ_k	γ_{k+1}	Θ_{k+1}	$\mathcal{J}.10^{-4}$
0	**1.0000**	**3.0000**	**1.2221**	**2.5045**	-
1	1.2220	2.5165	1.3867	2.6754	-
2	1.2121	2.5345	1.3790	2.6676	1.2013
3	1.2021	2.5645	1.3722	2.6604	0.9808
4	1.1921	2.5945	1.3656	2.6533	0.9397
5	1.1821	2.6245	1.3588	2.6462	0.9685
6	1.1721	2.6545	1.3521	2.6391	0.9530
7	1.1621	2.6845	1.3453	2.6320	1.1125
8	1.1521	2.7145	1.3386	2.6249	0.9530
9	1.1421	2.7445	1.3310	2.6171	1.0960
10	1.1321	2.7745	1.3251	2.6107	0.9565
19	1.0421	2.9645	1.2579	2.5410	1.1866
20	1.0321	2.9745	1.2495	2.5324	1.4285
21	1.0222	2.9845	1.2412	2.5239	1.4625
22	1.0121	2.9945	1.2328	2.5152	1.4112
23	1.0021	3.0245	1.2244	2.5068	0.0200
24	1.0011	3.0035	1.2234	2.5058	0.0185
25	**1.0001**	**3.0005**	**1.2223**	**2.5046**	-

The search procedure usually ends when the next (for example) three steps give a result of $\left|\mathcal{J}_{j+1}(s_m) - \mathcal{J}_j(s_m)\right| < \varepsilon_0$, $\varepsilon_0 > 0$, j – iteration step.

Using this method, one can made a modification taking into account the change in control on the last (n) segments, while maintaining the control from t_0 to t_{K-n} calculated using Bellman method from the beginning to the end.

The results of control calculations $s_m \in \mathcal{A}_z$, assuming a weighted criterion

$$\mathcal{J}(s_m) = \sum_{k=0}^{K-1} \left[\kappa_1 (\gamma_k - \Gamma_k)^2 + \kappa_2 a_z^2 \right] \tag{23}$$

is given in [7].

3.2. Method for control estimation using the properties of fuzzy logic

In this method, the relationship between the angle of attack α and the normal acceleration a_z was used. Each angle of attack value corresponds to specific normal acceleration $a_z \in \mathcal{A}_z$. Knowing the desired pitch angle and flight path angle, we have
$\alpha = \theta - \gamma$; $\alpha \in (\alpha_{min}, \alpha_{max}) \subset X$.

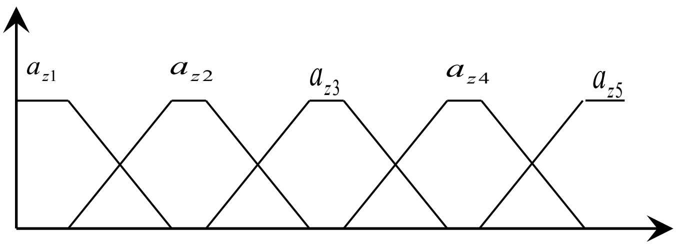

Fig. 5. Membership functions of angle of attack α for $a_z \in \mathcal{A}_z$

Between the minimum value and maximum value of the angle of attack, there is finite number of angles of attack. It would be impossible or at least very difficult it would be to consider all the possible values of this angle. Thus, only few reference normal acceleration values, which correspond to different ranges of angle of attack values. The intermediate values of angle of attack took a fuzzy form (Fig. 5). Using the dynamic programming principle, the trajectory was divided into permanent segments, determining the signals values for each segment of the x-axis.

After determining the normal acceleration at the stage k using the fuzzy inference, on the basis of the angle of attack value at the k+1 stage, the values of the pitch angle and the flight path angle at k-stage were defined. Using equations (3), (4), (5), (6) in reverse form, e.g. for the flight path angle, it would give:

$$\mu_{\gamma_k}(\gamma_k) = \max_{\theta_{k+1} \in \theta^*} [\mu_{\theta_{k+1}}(\theta_{k+1}) \wedge \mu_{\gamma_k}(\gamma_k, \theta)]. \tag{24}$$

For arbitrary values of flight path angle and pitch angle (as long as the normal acceleration value is within the range [α_{min}, α_{max}]), giving the acceleration at time k, we obtain the values of these angles at stage k + 1, for which the angle of attack has the same value depending on the used control. The angle of attack at the beginning of each stage may have different values. Three reference values of angle of attack were used at the beginning of each stage: the maximum negative value, zero, the maximum positive value. For each reference acceleration value and for each reference value of the angle of attack the transformation matrices for the flight path angle $\mu_{\gamma_k}(\gamma_k \mid \gamma_{k+1}, a_{k_i}, \alpha_{k_j})$ and the pitch angle $\mu_{\theta_k}(\theta_k \mid \theta_{k+1}, a_{k_i}, \alpha_{k_j})$ were developed. For these values, the flight path angle values could be calculated.

The membership function of the flight path angle is:

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}\alpha_{k_j}) = \max_{\gamma_{k+1}\in\gamma^*}[\mu_{\gamma_{k+1}}(\gamma_{k+1}) \wedge \mu_{\gamma_k}(\gamma_k \mid \gamma_{k+1}, a_{k_i}, \alpha_{k_j})]\,(25). \tag{25}$$

Accordingly, the membership function of the pitch angle is:

$$\mu_{\theta_k}(\theta_k, a_{k_i}\alpha_{k_j}) = \max_{\theta_{k+1}\in\theta^*}[\mu_{\theta_{k+1}}(\theta_{k+1}) \wedge \mu_{\theta_k}(\theta_k \mid \theta_{k+1}, a_{k_i}, \alpha_{k_j})]. \tag{26}$$

The membership function of height is:

$$\mu_{\Delta h_{k+1}}(\Delta h_{k+1}, a_{k_i}, \alpha_{k_j}) = \max_{\gamma_k\in\gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\Delta h_{k+1}}(\Delta h_{k+1} \mid a_{k_i}, \alpha_{k_j})]. \tag{27}$$

For a specified control and for any (not necessarily reference) angle of attack at time k, the values of the angles can be determined appropriately by using:

- for the flight path angle:

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_k) = \max_{\alpha_k\in\alpha_k{}^*}[\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\gamma_k}(\gamma_k, a_{k_i})], \tag{28}$$

where:

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}) = \begin{bmatrix} \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix}, \tag{29}$$

- for the pitch angle:

$$\mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_k) = \max_{\alpha_k\in\alpha_k{}^*}[\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\theta_k}(\theta_k, a_{k_i})], \tag{30}$$

where:

$$\mu_{\theta_k}(\theta_k, a_{k_i}) = \begin{bmatrix} \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix}, \tag{31}$$

- For height:

$$\mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_k) = \max_{\alpha_k\in\alpha_k{}^*}[\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\Delta h_k}(\Delta h_k, a_{k_i})], \tag{32}$$

where:

$$\mu_{\Delta h_k}(\Delta h_k, a_{k_i}) = \begin{bmatrix} \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix}. \tag{33}$$

The calculated γ_k, θ_k and Δh_k (for given $\gamma_{k+1}, \theta_{k+1}$ for which α_{k+1} belongs to adopted range of angle of attack values) correspond to one of the reference acceleration values. While for $\gamma_{k+1}, \theta_{k+1}$ for which α_{k+1} has a different value, γ_k, θ_k and Δh_k are calculated as following:

$$\gamma_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \gamma_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)}, \tag{34}$$

$$\theta_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \theta_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)}, \tag{35}$$

$$\Delta h_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \Delta h_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)}, \tag{36}$$

$$a_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot a_{k_p}}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)}. \tag{37}$$

Assuming the angle of attack α_1 on the entire trajectory, for a given final conditions, one solution could be found $(\gamma_1, \theta_1, H_1)$. This means that the final conditions could be achieved only with the altitude H_1 and with the γ_1, θ_1 angles. Therefore, another solution is also determined, this time for the angle of attack $-\alpha_1$, which gives another solution with the initial conditions defined as $(\gamma_4, \theta_4, H_4)$. Therefore, reaching the final condition could be achieved in the solution area for angle of attack α_1 (dark grey area in Fig. 6). With the given initial values of the angles at which the object is flying, the solution could be determine using fuzzy inferences (dashed line 1). To meet the condition (17), other solutions are generated in the same way for a different angle of attack (dotted line 2). Thus, for any height that is between the two heights through another fuzzy inference, the normal acceleration could be determined along the whole trajectory.

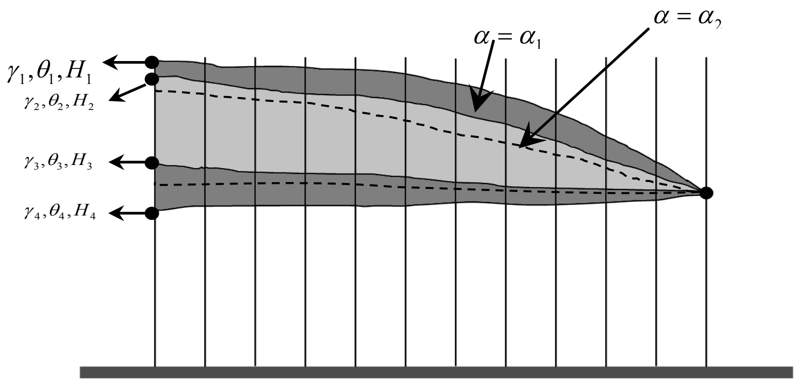

Fig. 6. Possible solutions illustration

4. Simulation Results

The algorithm was test for the following final conditions: $\gamma_{final} = -10$ [deg], $\theta_{final} = -10$ [deg], $H_{final} = 2600$ [m]. Fig. 7, 8 and 9 present the simulation results for the initial angles γ, θ at which the object starts its trajectory. Two solutions for two angles of attack were determined. For each solution, we received an initial altitude. As a result of the next fuzzy inferences, a set of guiding signals for the given initial conditions at which the object is flying were calculated. The following initial conditions were considered: $\gamma_{initial} = 5$ [deg], $\theta_{initial} = 8$ [deg], $H_{initial}$=2830 [m].

Both solutions generate guidance signals to (normal acceleration) for two different altitudes, leading the object to final conditions.

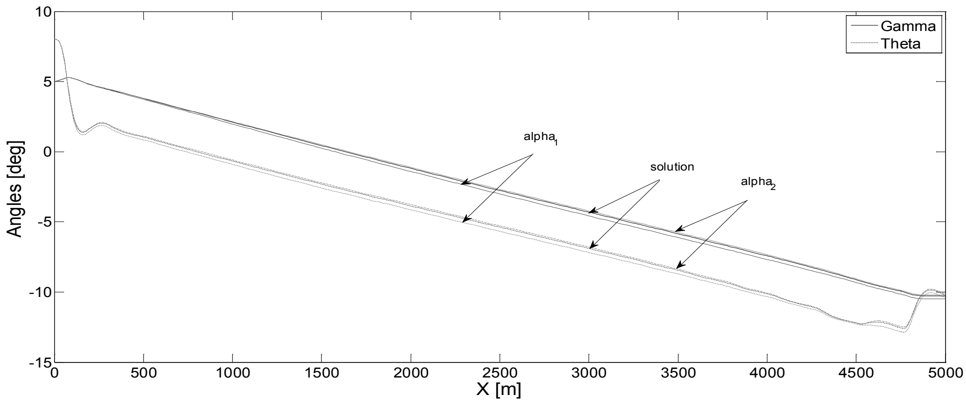

Fig. 7. Flight path angle and pitch angle along the trajectory

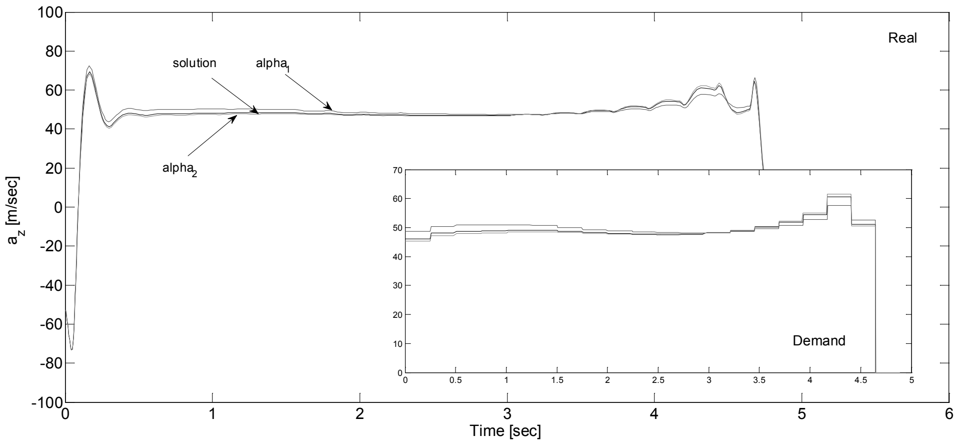

Fig. 8. The demanded and actual normal acceleration along the trajectory

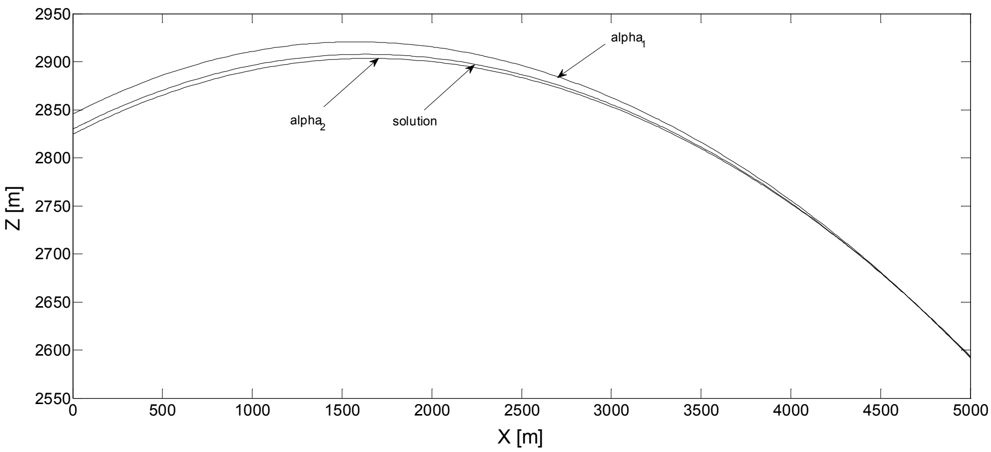

Fig. 9. Flying object trajectory

On the basis of these two solutions, for the altitude given in the initial conditions and using fuzzy inference, the solution for the given initial conditions was found.

5. Conclusions

Analysing the computational complexity of many algorithms that could be used to solve task with the given final conditions, the methods proposed in this article have many advantages. Particularly, Bellman method based fuzzy approach is not so time consuming as optimal control algorithms QRT or Pontriagin method. It also allows the object to flight along the predefined trajectory. Currently, further work is conducted on using this algorithm for flying at low altitudes, particularly over configured terrain (over the forests, mountainous terrain, etc.). These algorithms are aimed to ensure the maximum safety of the flight by considering only the acceleration values that are possible to achieve by the object.

References

[1] Basmadji, F. L., Gruszecki, J., *Sequential fuzzy missile guidance system while flying over configured terrain,* Scientyfic Aspects of Unmanned Mobile Vehicle, pp. 335-348, ISBN 978-83-932107-0-1, Kielce 2010.

[2] Buckley, J. J., Jowers, L. J., *Monte Carlo Methods in Fuzzy Optimization*, Springer-Verlag 2008.

[3] Findeisen W., Szymanowski J., Wierzbicki A.: *Teoria i metody obliczeń aproksymacji*, PWN, Warszawa 1980.

[4] Gruszecki, J., Bociek, S., Chłędowski, M., Dołęga, B., Grzybowski, J., Pieniążek, J., Rogalski, T., Tomczyk, A., *Bezpilotowe aparaty latające*, Oficyna Wydawnicza Politechniki Rzeszowskiej, 1-308, 2002.

[5] Kacprzyk, J., *Wieloetapowe sterowanie rozmyte,* Wydawnictwo Naukowo-Techniczne, WNT, Warszawa 2001.

[6] Roberts, S. M., *Dynamic Programming In Chemical Engineering and Process Control*, Academic Press, New York 1964.

[7] Szumała, O., *Sterowanie samolotem na trajektorii z zadanym warunkiem końcowym*, Praca dyplomowa, Wydz. BMiL PRz, Rzeszow 2016.