Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference
on Guidance, Navigation & Control, Delft University of Technology,
Delft, The Netherlands, April 10-12, 2013

UAV trajectory generation using fuzzy dynamic programming

Fatina Liliana Basmadji, Jan Gruszecki

Abstract This paper presents an algorithm based on fuzzy dynamic programming to generate UAV trajectory in the x-z plane. The dynamics of the UAV that had been written in a fuzzy form and the initial and final conditions relating to altitude and attitude had been considered.

1 Introduction

Unmanned Air Vehicles of a classical design have become more and more used for several purposes such as terrestrial scanning, making digital maps, terrain observation etc. One of the conditions while realizing these tasks is to fly at a constant altitude above the terrain. Aircraft control while performing such tasks that are considered as flights over configured terrain could be realized:
- using a variety of deterministic algorithms that use mathematical models of the objects in the form of differential equations,
- using discrete control algorithms,
- using control algorithms from the group of expert algorithms such as: fuzzy control, robust control or using neural networks.

The case where we are to achieve trajectory of given final conditions and meet certain conditions along the whole trajectory including initial conditions seems to be very interesting from a theoretical point of view. In this paper, the possibility of solving such task using dynamic programming and multidimensional fuzzy logic is presented.

2 Rewriting UAV equations of motion in a fuzzy form

The purpose of rewriting UAV equations of motion in a fuzzy form (as fuzzy system) was to replace trajectory generation by solving a set of differential equations

Fatina Liliana Basmadji
Rzeszow University of Technology, email: basmadji@prz.edu.pl
Jan Gruszecki
Rzeszow University of Technology, email: awionjgr@prz.edu.pl

with a simple fuzzy inference. The normal acceleration a_z was assumed to be the signal generated by the guidance system and the input of the control system.
Let us start with the concept of the deterministic control system as a system for which the state change is determined by the following function

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where, $x_k, x_{k+1} \in X$ and X is a set of states $X = \{s_1, \cdots s_n\}$, while $u_k \in U$ where U is a set of controls $U = \{c_1, \cdots, c_m\}$.
However, the system considered in this work is not a deterministic system because the altitude of the UAV and its attitude (pitch angle) and flight path angle can have any value, and in this case the set X would be infinitely large. The same thing is true of control signals. Therefore, this system was assumed to be a fuzzy system. Fuzzy control system is a system for which the change of state variables is determined using the following function

$$X_{k+1} = f(X_k, U_k) \tag{2}$$

As before, we assume that X is a set of states $X = \{s_1, \cdots s_n\}$, whereas U is a set of controls $U = \{c_1, \cdots, c_m\}$. Fuzzy state in the k-th stage is defined as a fuzzy set X_k defined in X and its membership function is $\mu_{X_k}(x_k)$. In the case of fuzzy control, U_k is defined in U, and its membership function is $\mu_{U_k}(u_k)$. More details about fuzzy systems can be found in [2].
In our previous work [1], equations of state variables were derived. For the flight path angle

$$\mu_{\gamma_{k+1}}(\gamma_{k+1}) = \max_{\theta_k \in \theta^*}[\mu_{\theta_k}(\theta_k) \wedge \mu_{\gamma_{k+1}}(\gamma_{k+1}, \theta)] \tag{3}$$

where:

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

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

While $\mu_{\gamma_{k+1}}(\gamma_{k+1} \mid \gamma_k, a_k)$ represents the transformation matrix of the flight path angle γ from γ_k to γ_{k+1}, for a given pitch angle $\theta_k = \theta_i$ and a given acceleration a_z^j. The membership function of the pitch angle θ is

$$\mu_{\theta_{k+1}}(\theta_{k+1}) = \max_{\gamma_k \in \gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\theta_{k+1}}(\theta_{k+1}, \gamma)] \tag{4}$$

respectively: for altitude

$$\mu_{\Delta h_{k+1}}(\Delta h_{k+1}) = \max_{\gamma_k \in \gamma^*}[\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\Delta h_{k+1}}(\Delta h_{k+1}, \gamma)] \tag{5}$$

and the pitch rate

$$\mu_{q_{k+1}}(q_{k+1}) = \max_{\theta_k \in \theta^*}[\mu_{\theta_k}(\theta_k) \wedge \mu_{q_{k+1}}(q_{k+1}, \theta)] \tag{6}$$

While generating transformation matrices it was assumed that the pitch rate is equal to zero. However, the pitch rate is not necessarily equal to zero. Therefore, the following correction was introduced to the pitch angle

$$q = \dot{\theta} \quad \rightarrow \quad q = \frac{\theta_{k+1} - \theta_k}{\Delta t} \quad \rightarrow \quad \theta_{k+1} = q \cdot \Delta t + \theta_k \tag{7}$$

for small distance Δx the trajectory could be assumed to be linear, thus

$$\Delta t = \frac{\Delta x}{V \cdot \cos\gamma} \tag{8}$$

where V - is UAV velocity.

3 Fuzzy dynamic programming algorithm

3.1 Cost function

Fuzzy dynamic programming is a dynamic programming with some of the variables fuzzy [4]. In this work, the goal imposed on the final state concerns three state variables connected with each other: flight path angle, pitch angle and altitude. Equations (6), (7) and (8) represent the goals imposed on flight path angle, pitch angle and altitude respectively:

$$\mu_{\overline{G}\Gamma^N}(\Gamma_N) = \max_{\gamma_N \in \Gamma}\left[\mu_{\Gamma_N}(\gamma_N) \wedge \mu_{G\Gamma^N}(\gamma_N)\right] \tag{6}$$

$$\mu_{\overline{G}\Theta^N}(\Theta_N) = \max_{\theta_N \in \Theta}\left[\mu_{\Theta_N}(\theta_N) \wedge \mu_{G\Theta^N}(\theta_N)\right] \tag{7}$$

$$\mu_{\overline{G}H^N}(H_N) = \max_{h_N \in H}\left[\mu_{H_N}(h_N) \wedge \mu_{GH^N}(h_N)\right] \tag{8}$$

The overall goal imposed on the final state which is the cost function that should be maximized could be written as following

$$I = \mu_{\overline{G}^N}(X_N) = \mu_{\overline{G}\Gamma^N}(\Gamma_N) \wedge \mu_{\overline{G}\Theta^N}(\Theta_N) \wedge \mu_{\overline{G}H^N}(H_N) \tag{9}$$

3.2 Solution for one stage

The developed algorithm take into account the relationship between angle of attack α and normal acceleration a_z. Considering the forces acting on the unmanned aerial vehicle during flight in x-z plane, from Newton second law

$$F_z = m \cdot a_z \tag{10}$$

where

$$F_z = -L \cdot \cos\alpha - D \cdot \sin\alpha + W \cdot \cos\theta \tag{11}$$

thus

$$a_z = \frac{1}{m}\left(-L \cdot \cos\alpha - D \cdot \sin\alpha + W \cdot \cos\theta\right) \tag{12}$$

$L -$ lift force,
$D -$ drag force,
$W -$ weight.

Lift force and drag force are related to angle of attack, thus, normal acceleration could be written as a function of angle of attack. Each small range of angle of attack values corresponds to a certain normal acceleration value. Hence, knowing the desired attitude angle and flight path angle, the desired angle of attack is known. However, there is an infinite number of angle of attack values between the minimum and the maximum values and it is impossible or very difficult to consider all allowable angle of attack values. Therefore, a number of reference normal acceleration values that corresponds to different ranges of angle of attack values were considered. The membership functions of angle of attack are presented in Fig. 1.

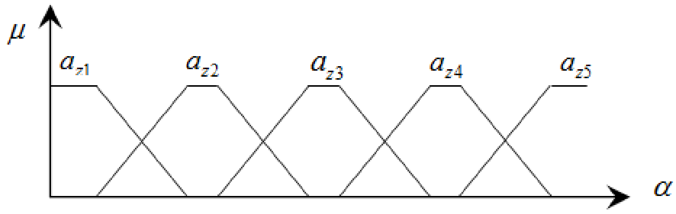

Fig. 1 Angle of attack membership functions

In order to use the principle of dynamic programming, the trajectory should be divided into stages, then, the normal acceleration is determined on each stage. In this work, stages of the same length on the x-axis had been considered. After determining the normal acceleration at k-th stage on the basis of the angle of attack value at the k+1-th stage using fuzzy inference, flight path angle and the pitch angle at k-th stage have to be determined. This could be done using the equations (3),(4),(5),(6) in the inverse form, for example, for the flight path angle

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

However, this approach could be used only if it is possible to determine $\mu_{\gamma_k}(\gamma_k, \theta_{k+1})$ for different flight path angle values. Since, as previously mentioned, the normal acceleration at k-th stage corresponds to a small range of angle of attack values, $\mu_{\gamma_k}(\gamma_k, \theta_{k+1})$ could not be determined for every possible pitch angle value. Matrix inversion algorithm also could not be used because in this case $\mu_{\gamma_k}(\gamma_k, \theta_k)$ would be determined. Therefore, a different approach was adopted. For any flight path angle and pitch angle values (unless the angle of attack is beyond the accepted limits), having the normal acceleration at k-th stage, the values of these angles in the k+1-th stage for which the angle of attack has a certain value depending on the acceleration given as input signal are determined. That is, the angle of attack at the beginning of each stage can have different values. Three different angle of attack values was considered to be the reference values at the beginning of each stage: maximum negative, zero and maximum positive. For each of the reference acceleration values and each of the reference angle

of attack values the transformation matrix for the flight path angle $\mu_{\gamma_k}(\gamma_k \mid \gamma_{k+1}, a_{k_i}, \alpha_{k_j})$ and for the pitch angle $\mu_{\theta_k}(\theta_k \mid \theta_{k+1}, a_{k_i}, \alpha_{k_j})$ were determined. Hence the angle values at the k-th stage could be calculated.
The membership function of the flight path angle

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_j}) = \max_{\gamma_{k+1} \in \gamma^*} [\mu_{\gamma_{k+1}}(\gamma_{k+1}) \wedge \mu_{\gamma_k}(\gamma_k \mid \gamma_{k+1}, a_{k_i}, \alpha_{k_j})] \tag{14}$$

respectively, the membership function of the pitch angle

$$\mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_j}) = \max_{\theta_{k+1} \in \theta^*} [\mu_{\theta_{k+1}}(\theta_{k+1}) \wedge \mu_{\theta_k}(\theta_k \mid \theta_{k+1}, a_{k_i}, \alpha_{k_j})] \tag{15}$$

However, altitude could be determined using the following membership function

$$\mu_{\Delta h_{k+1}}(\Delta h_{k+1}, a_{k_i}, \alpha_{k_j}) = \max_{\gamma_k \in \gamma^*} [\mu_{\gamma_k}(\gamma_k) \wedge \mu_{\Delta h_{k+1}}(\Delta h_{k+1} \mid a_{k_i}, \alpha_{k_j})] \tag{16}$$

For one reference control, and for any (not necessary one of the reference values) angle of attack at the k-th stage, the values of the angles could be determined as following, for the flight path angle

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_k) = \max_{\alpha_k \in \alpha_k^*} [\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\gamma_k}(\gamma_k, a_{k_i})] \tag{17}$$

where

$$\mu_{\gamma_k}(\gamma_k, a_{k_i}) = \begin{bmatrix} \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\gamma_k}(\gamma_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix} \tag{18}$$

- for the pitch angle

$$\mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_k) = \max_{\alpha_k \in \alpha_k^*} [\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\theta_k}(\theta_k, a_{k_i})] \tag{18}$$

where

$$\mu_{\theta_k}(\theta_k, a_{k_i}) = \begin{bmatrix} \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\theta_k}(\theta_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix} \tag{19}$$

- for altitude

$$\mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_k) = \max_{\alpha_k \in \alpha_k *}[\mu_{\alpha_k}(\alpha_k) \wedge \mu_{\Delta h_k}(\Delta h_k, a_{k_i})] \tag{20}$$

where

$$\mu_{\Delta h_k}(\Delta h_k, a_{k_i}) = \begin{bmatrix} \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_1}) \\ \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_2}) \\ \mu_{\Delta h_k}(\Delta h_k, a_{k_i}, \alpha_{k_3}) \end{bmatrix} \tag{21}$$

In this way, γ_k, θ_k and Δh_k can be calculated for given $\gamma_{k+1}, \theta_{k+1}$ for which α_{k+1} belongs to a range that corresponds to one of the reference normal accelerations. However, for $\gamma_{k+1}, \theta_{k+1}$ for which α_{k+1} has another value, γ_k, θ_k and Δh_k are calculated as following:

$$\gamma_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \gamma_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)} \tag{22}$$

$$\theta_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \theta_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)} \tag{23}$$

$$\Delta h_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot \Delta h_k(a_{k_p})}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)} \tag{24}$$

$$a_k = \frac{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k) \cdot a_{k_p}}{\sum_{p=1}^{n} \mu_{\alpha_{k_p}}(\alpha_k)} \tag{25}$$

3.3 Determining guidance signals along the trajectory

After the first waypoint (final conditions) has been chosen, an area of available solutions for the next waypoint is generated. Each altitude that belongs to this area corresponds to particular values of attitude angle and flight path angle. By choosing the altitude of the waypoint, the corresponding control signals, attitude and flight path angles are been calculated using previously mentioned multidimensional fuzzy inference. The initial conditions are determined once the last waypoint is chosen. Fig. 2 shows the solution area for one stage. In Fig. 3 several stages have been taken into consideration and the solution areas were presented. Fig. 4 presents the whole trajectory where several waypoints had been determined.

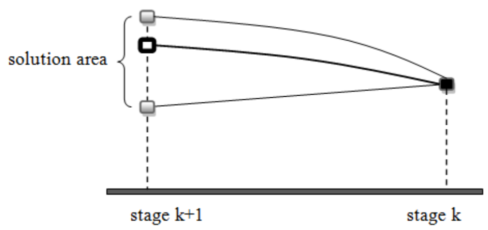

Fig. 2 Solution area for one stage

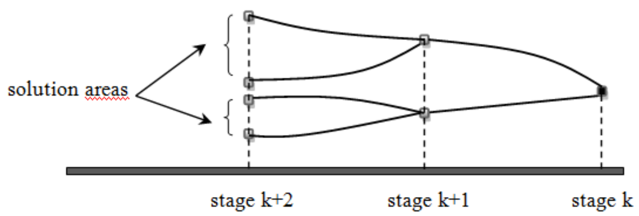

Fig. 3 Solution areas of the whole trajectory

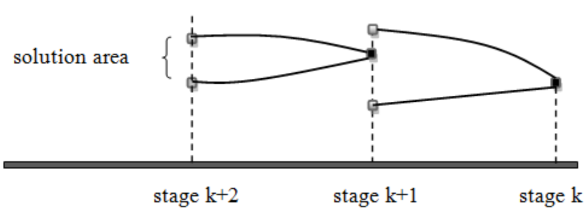

Fig. 4 Solution areas of the whole trajectory using waypoints

4 Simulation results

The algorithm had been tested on an example in which the final path angle and final pitch angle were equal to zero and the final altitude was about 2600 m.
Fig. 5 presents the trajectory that was generated using a set of waypoints. Fig. 6 shows the corresponding control signals. Fig. 7 show the corresponding values of flight path angle and pitch angle. In this example the horizontal distance between the waypoints is constant, however, the waypoints could be determined in such a way that the distance between each two waypoints could be different.

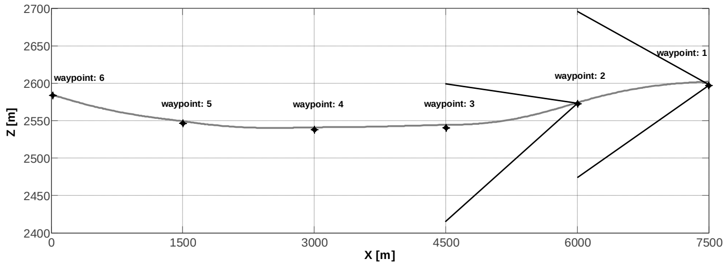

Fig. 5 Trajectory generated using fuzzy dynamic programming

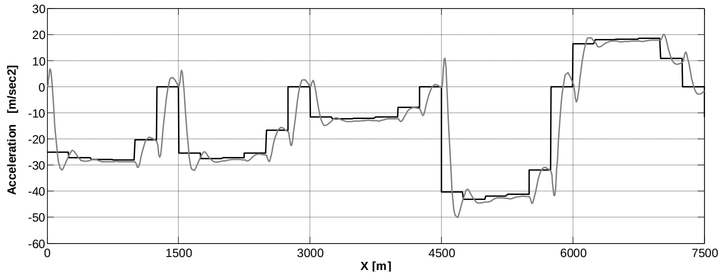

Fig. 6 Control signals (normal acceleration)

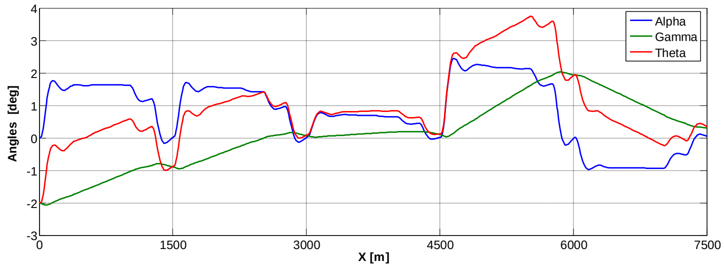

Fig. 7 Angle of attack, flight path angle and pitch angle

5 Conclusions

Analyzing the computational complexity of several algorithms that could be used to solve the considered task with given final conditions, the algorithm proposed in this paper has several advantages. The discrete approach of Bellman method for a fuzzy system is not as time consuming as QRT optimal control algorithms or Pontryagin method. It also allow o generate flight trajectory of a predefined profile. Currently, this algorithm is being developed for flying over configured terrain. While flying over configured terrain, the terrain configuration should be considered when determining the solution area and all the solutions that leads to a collision with the terrain are eliminated. Analysis of the possibility of realizing the planned mission is another facilitation for the person who is planning the trajectory of the UAV [3]. The presented algorithm could also be used as trajectory planning training tool. This method has also its disadvantages associated with not taking into account the continuous kinematic relationships of the UAV that must be met at each time. However, it seems that these problems can be eliminated by developing the proposed algorithm.

References

1. Basmadji F. L., Gruszecki J.: *"Sequential fuzzy missile guidance system while flying over configured terrain".* "Scientific Aspects of Unmanned Mobile Vehicle", p.335-348, ISBN 978-83-932107-0-1 Kielce 2010
2. Kacprzyk J.: Wieloetapowe sterowanie rozmyte. Wydawnictwo Naukowo-Techniczne, WNT, Warszawa 2001
3. Gruszecki J., Bociek S., Chłędowski M., Dołęga B., Grzybowski J., Pieniążek J., Rogalski T., Tomczyk A.: "Bezpilotowe aparaty latające". *"OFICYNA WYDAWNICZA POLITECHNIKI RZESZOWSKIEJ"*, 2002,1-308.
4. Buckley, James J., Jowers, Leonard J: Monte Carlo Methods in Fuzzy Optimization. Springer-Verlag 2008

Acknowledgments

"This scientific work has been financed from the Polish scientific funds for years 2010-2012 as a development project No. OR00011611"

Numerical experiments were conducted with the use of MATLAB application, purchased during the realization of project no. UDA-RPPK.01.03.00-18-003/10-00 "Construction, expansion and modernization of the scientific-research base at Rzeszów University of Technology" is co-financed by the European Union from the European Regional Development Fund within Regional Operational Programme for the Podkarpackie Region for the years 2007-2013, I. Competitive and innovative economy, 1.3 Regional innovation system.