A multi-objective optimized fuzzy back-stepping controller for a nonlinear unmanned aerial vehicle

Mohammad Javad Mahmoodabadi (mahmoodabadi@sirjantech.ac.ir)
Sirjan University of Technology

Nima Rezaee Babak
Sirjan University of Technology

Research Article

Keywords: Nonlinear quadcopter, Back-stepping control, Fuzzy system, Optimal control, Multi-objective grey wolf optimization

Posted Date: June 7th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2714558/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
A multi-objective optimized fuzzy back-stepping controller for a nonlinear unmanned aerial vehicle

M.J. Mahmoodabadi*, N. Rezaee Babak
Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran.
* Corresponding author; Email: mahmoodabadi@sirjantech.ac.ir

Abstract
This paper introduces a multi-objective optimized fuzzy back-stepping (FBS) controller to stabilize an unmanned aerial vehicle (UAV) having nonlinear dynamics. At first, the back-stepping (BS) technique is carried out as the main approach to determine the control efforts of the quadcopter. Next, fuzzy systems are appropriately designed to tune the gains of the BS and to reach robustness. Afterwards, to simultaneously minimize the control efforts and system errors, the parameters of the fuzzy formulations are ascertained via multi-objective grey wolf optimization (MOGWO). This strategy is eventually applied on a nonlinear UAV model designed and fabricated in the department of mechanical engineering at Sirjan university of technology. The obtained results are illustrated to prove the effectiveness of the introduced strategy for stabilization of the considered UAV system.

Keywords: Nonlinear quadcopter; Back-stepping control; Fuzzy system; Optimal control; Multi-objective grey wolf optimization.

1. Introduction
The back-stepping control scheme, which has been utilized in various fields of the science and engineering, is known as one of the best control approaches with the acceptable computational cost. It combines the Lyapunov function with a feedback control scheme in a recursive procedure which has been broadly employed in scientific researches. By means of illustration, Zhou and Wen have introduced an adaptive back-stepping feedback scheme for stabilization of nonlinear uncertain systems with quantized states [1]. An active full-vehicle suspension controller has been developed through a cloud-aided adaptive back-stepping procedure by Zheng and Zhang [2]. Zhao and Cheng have presented an angle-tracking back-stepping controller for pneumatic muscle actuators through an adaptive extended state observer [3]. Further, the back-stepping control of a nonlinear quadcopter slung load system has been developed by Yu and Cabecinhas [4]. Yin and Wu have employed an adaptive robust back-stepping controller for a speed regulating differential mechanism of a turbine [5]. Wang and Xie have designed an adaptive back-stepping controller to solve the position regulation problem of an uncertain gear transmission servo system having asymmetric dead-zone nonlinearities [6]. Furthermore, a fractional-order back-stepping sliding mode controller has been suggested for an unidentified aerial vehicle quadcopter by Shi and Cheng [7]. Poultney and Gong have studied an integral back-stepping controller for trajectory and yaw motion tracking of quadcopters [8]. Hajji and Ayadi have utilized back-stepping and high gain stabilizers in a pneumatic system to accomplish the position control [9]. Moreover, Hu and Meng have studied a longitudinal flight controller for air-breathing hypersonic vehicles carrying aerodynamic uncertainties, input nonlinearities and flexible modes, through applying an inverse dynamic controller as well as an adaptive back-stepping technique [10].

Basically, a controller would be effective for producing desired performance, if its design parameters are appropriately determined. Calculations of such parameters for complex or multistate systems by trial and error procedures can be challenging or even impossible. To solve this problem, many schemes such as predictive methods [11-14], neural networks [15-18], evolutionary approaches [19-25] and fuzzy systems [26-33] have been widely employed to improve the performance of designed controllers through tuning their coefficients. In this regard, fuzzy systems would be designed
to be combined with the back-stepping controller to adaptively and robustly tune the gains, in this research. In fact, the fuzzy systems, as a tool against uncertainties and vagueness, is broadly applied in combination with control methods. Further, fuzzy systems are basically robust and able to regulate the relation between system properties and expert knowledge. A number of noticeable research works done in this field could be cited as follows. An adaptive back-stepping sliding mode controller joined with a fuzzy system has been employed to stabilize a cantilever having unknown dynamics, external disturbance and model uncertainties [34]. The speed of a linear induction motor with external load disturbances and uncertain parameters has been controlled through combining an adaptive fuzzy system with a command-filtered back-stepping controller by Xu and Huang [35]. Moreover, a class of uncertain nonlinear strict-feedback systems has been controlled using an adaptive fuzzy back-stepping method in accordance with a dynamic surface control approach by Peng and Dubay [36]. Chen and Huang have succeeded to control nonlinear bilateral teleoperation manipulators and enhanced the related transparency performance by applying an adaptive fuzzy back-stepping controller [37]. Wu and Jin have presented a trajectory tracking controller based on fuzzy sliding modes along with kinematic back-stepping approaches for differential mobile robots [38]. El-samahy and Shamseldin have utilized self-tuning fuzzy PID and model reference adaptive control methods to solve the tracking problem for a brushless DC-motor [39]. Additionally, Eltag and Aslamx have used fuzzy PID control schemes optimized by employing particle swarm optimization to improve the dynamical stability of a power system during low frequency oscillations [40]. A fuzzy adaptive back-stepping controller has been introduced for a class of active suspension systems, while the stability of the closed-loop system and the convergence of the observer have been proved [41]. Finally, to handle discrete-time uncertain nonlinear systems, Ji and Qiu have proposed a fuzzy-model-based output feedback sliding mode controller [42].

On the other hand, gain determination (gain scheduling) constitutes a significant affair in the design process of the controllers. Over the past years, researchers have developed meta-heuristic optimization algorithms to deal with this matter. A number of popular examples of such algorithms could be listed as follows: genetic algorithm by Holland [43], ant colony optimization by Colorni and Dorigo [44], particle swarm optimization by Eberhart and Kennedy [45], team game algorithm by Mahmoodabadi et. al. [46], converged teacher-learning-based optimization by Mahmoodabadi and Ostadzadeh [47], dragonfly algorithm by Mafarja and Heidari [48], lion optimizer by Heidari and Faris [49], grasshopper optimization by Saremi and Mirjalili [50], multi-verse optimizer by Aljarah and Mafarja [51], grey wolf optimizer by Mirjalili and Aljarah [52], whale optimization by Mirjalili and Mirjalili [53], and tree optimization algorithm by Mahmoodabadi et al. [54].

Besides, balancing control efforts and output errors for controller design is of high priority and achievable through the multi-objective optimization processes. In this work, the fuzzy parameters, as the design variables, are regulated via a multi-objective grey wolf optimization (MOGWO) algorithm. In this way, two conflicting objective functions would be considered as summation of absolute control effort values and summation of absolute output error values that aimed to be minimized, simultaneously. It is noteworthy that the past researches have suggested the multi-objective methods more efficient than the single-objective ones for solving these types of optimization problems and making the trade-off between the objective functions. For instance, Nisi and Nagaraj have regulated the gains of a PID controller by combining multi-objective evolutionary optimization strategies such as the genetic algorithm, particle swarm optimization and bacterial foraging approach and compared the results with the conventional methodologies [55]. Karmellos and Mavrotas have presented a multi-objective optimization methodology with a comparison framework for design of distributed energy systems in order to satisfy local needs in heating, cooling and electricity used to promote energy efficiency and reduce carbon emissions [56]. Yang and Wang have studied a hybrid method based on a dual decomposition strategy and a multi-objective optimization algorithm for electricity
price forecasting [57]. Nagarkar and Bhalerao have developed a multi-objective genetic algorithm for an active nonlinear quarter-car suspension-seat-driver model with cubic nonlinearity terms in the spring and quadratic ones in the tire [58]. Bui and Tran have introduced multi-objective optimal design of a fuzzy system to control the structural vibration with actuator saturation through a hedge-algebras approach [59]. Tian and Hao have suggested a wind speed forecasting system through a data preprocessing algorithm and Elman neural network models improved by a multi-objective satin bowerbird optimizer [60]. Liu and Li have developed a model-based strategy for multi-objective optimization of charging patterns for lithium-ion batteries by considering various key parameters such as the charging speed, energy conversion efficiency as well as temperature variations [61]. Beykal and Boukouvala have succeeded optimal design of energy systems via a grey-box multi-objective optimization scheme and tested the performance of the framework in terms of accuracy and consistency, under many equality constraints [62].

The contributions of this study could be briefly described as follows: An effective back-stepping control approach is implemented as the main controller for a nonlinear UAV flying machine. To tune the constant control gains, the controller is combined with fuzzy logic-based system. The state errors constitute as the fuzzy inputs, while the centers of the output membership function are acquired using the multi-objective grey wolf optimization algorithm. Finally, one of non-inferior solutions exhibited in the Pareto optimal front is chosen to depict the time responses of the system states.

The rest of the paper can be represented as follows: Section 2 elaborates the dynamical equations of the considered UAV. Section 3 concisely introduces the back-stepping control design and discusses its combination with the fuzzy logic-based systems. Subsequently, Section 4 explains how to optimize the implemented fuzzy systems by the MOGWO algorithm. Section 5 displays the results and comparisons with other works in order to reveal the capabilities of the presented strategy. Conclusions are also given in Section 6.

2. Dynamical equations of the UAV

The unmanned aerial vehicle studied in this work has been designed and fabricated in the mechanical engineering department at Sirjan university of technology for monitoring the Sirjan salt playa [63]. Figure 1 depicts a general view of this versatile machine patented in Iran which is able to move in the air, on the ground and on the water [63]. The dynamical equations of the machine in an earth-fixed coordinate system could be given based on the work of Parhizkar and Naghash [64] as follows.

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \theta_1 x_4 x_6 - \theta_4 (x_4 + x_1 x_6) \Omega + b_2 U_4 x_3 + b_1 U_2 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \theta_2 x_2 x_6 + \theta_5 (x_2 - x_3 x_6) \Omega - b_3 U_4 x_1 + b_2 U_3 \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \theta_3 x_2 x_4 + \theta_5 x_1 x_2 \Omega + b_2 U_3 x_1 + b_3 U_4 \\
\dot{x}_7 &= x_8 \\
\dot{x}_8 &= -g + \left( \frac{1}{m} \right) (\cos(x_3) \cos(x_1))(U_1 + \Delta + mg)
\end{align*}
\]

in which,
\[
\begin{align*}
\chi_1(t) &= \phi(t) \\
\chi_2(t) &= \phi(t) \\
\chi_3(t) &= \theta(t) \\
\chi_4(t) &= \theta(t) \\
\chi_5(t) &= \psi(t) \\
\chi_6(t) &= \psi(t) \\
\chi_7(t) &= Z(t) \\
\chi_8(t) &= Z(t)
\end{align*}
\]

(2)

while, \(\phi, \theta\) and \(\psi\) respectively denote the rotations about \(x, y\) and \(z\) axes. Further, \(Z\) represents the quadcopter height, \(\Delta\) stands for the disturbance in \(z\) direction and \(\hat{m}\) is the uncertain mass of the whole system. In the absence of any uncertainties, \(\hat{m} = m\) and for the system without any disturbance \(\Delta = 0\). Furthermore, the other parameters are defined as follows:

\[
q_1 = \frac{l_y - l_z}{l_x} + 1, q_2 = \frac{l_x - l_y}{l_x} - 1, q_3 = \frac{l_x - l_y}{l_x} + 1, q_4 = \frac{J_r}{l_x}, q_5 = \frac{J_r}{l_y}
\]

(3)

\[
b_1 = \frac{1}{l_x}, b_2 = \frac{1}{l_x}, b_3 = \frac{1}{l_x}
\]

(4)

\[
\Omega = \omega_1 - \omega_2 + \omega_3 - \omega_4
\]

(5)

\[
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} =
\begin{bmatrix}
T_1 + T_2 + T_3 + T_4 \\
-(T_2 - T_4)L \\
(T_1 - T_3)L \\
D_B(-T_1 - T_3 + T_2 + T_4)
\end{bmatrix}
\]

(6)

\[
T_i = B\omega_i^2, \quad i = 1, 2, 3, 4
\]

(7)

where, \(D\) and \(B\) respectively refer to drag and thrust coefficients. \(U_i\) and \(T_i\) \((i = 1, 2, 3, 4)\) correspondingly illustrate the control effort and torque of the system. The descriptions and numerical values of the coefficients of the studied UAV are listed in Table 1.

![Figure 1: General view of the UAV manufactured in the mechanical engineering department at Sirjan university of technology, patented in Iran [63].](image)

<table>
<thead>
<tr>
<th>Description</th>
<th>Notation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>(m)</td>
<td>1.250 kg</td>
</tr>
<tr>
<td>Effective length</td>
<td>(l)</td>
<td>23 cm</td>
</tr>
<tr>
<td>Inertia about axis z</td>
<td>(I_z)</td>
<td>0.0036 kgm²</td>
</tr>
<tr>
<td>Inertia about axis x</td>
<td>(I_x)</td>
<td>0.00251 kgm²</td>
</tr>
<tr>
<td>Inertia about axis y</td>
<td>(I_y)</td>
<td>0.0021 kgm²</td>
</tr>
<tr>
<td>Thrust coefficient</td>
<td>(B)</td>
<td>645 Ns²</td>
</tr>
<tr>
<td>Drag coefficient</td>
<td>(D)</td>
<td>(0.08 \times 10^{-7}) Nm²</td>
</tr>
<tr>
<td>Rotor inertia</td>
<td>(J_r)</td>
<td>(1.808 \times 10^{-5}) kgm²</td>
</tr>
</tbody>
</table>
3. Proposed controller

3.1. Back-stepping control

The suggested back-stepping controller is elaborated here in details with respect to Equation (1). At first, a new form of the governing dynamical equations is considered as follow.

\[ \dot{\eta} = f(\eta) + G(\eta)\zeta, \]
\[ \dot{\zeta} = f_a(\eta, \zeta) + G_a(\eta, \zeta)u. \]

As it is clear, Equation 8 is affine to \( \zeta \) (\( f \) is dependent only to \( \eta \)). Therefore, the state-space equations would be rewritten as below:

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_3 \\
\dot{x}_5
\end{bmatrix} = \begin{bmatrix}
x_2 \\
x_4 \\
x_6
\end{bmatrix},
\]
\[
\begin{bmatrix}
\dot{x}_2 \\
\dot{x}_4 \\
\dot{x}_6
\end{bmatrix} = \begin{bmatrix}
\phi_1(x_4x_6) - \phi_4(x_4 + (x_1x_6))\Omega \\
\phi_2(x_2x_6) + \phi_5(x_2 - (x_3x_6))\Omega \\
\phi_3(x_2x_4) + \phi_5(x_1x_2)\Omega
\end{bmatrix} + \begin{bmatrix}
b_1 & 0 & b_3x_3 \\
0 & b_2 & -b_3x_1 \\
0 & b_2x_1 & b_3
\end{bmatrix} \begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix}.
\]

(10)

Moreover, the error vectors would be defined as:

\[
\eta = \begin{bmatrix}
z_1 \\
z_3 \\
z_5
\end{bmatrix} = \begin{bmatrix}
x_{1d} - x_1 \\
x_{3d} - x_3 \\
x_{5d} - x_5
\end{bmatrix},
\]
\[
\zeta = \begin{bmatrix}
z_2 \\
z_4 \\
z_6
\end{bmatrix} = \begin{bmatrix}
x_{2d} - x_2 \\
x_{4d} - x_4 \\
x_{6d} - x_6
\end{bmatrix},
\]

(11)

(12)

in which, \( x_{1d}, x_{2d}, ..., x_{6d} \) signify the desired inputs that must be reached, also

\[
f(\eta) = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}, G(\eta) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix},
\]
\[
f_a(\eta, \zeta) = -\begin{bmatrix}
\phi_1(x_4x_6) - \phi_4(x_4 + (x_1x_6))\Omega \\
\phi_2(x_2x_6) + \phi_5(x_2 - (x_3x_6))\Omega \\
\phi_3(x_2x_4) + \phi_5(x_1x_2)\Omega
\end{bmatrix},
\]
\[
G_a(\eta, \zeta) = -\begin{bmatrix}
b_1 & 0 & b_3x_3 \\
0 & b_2 & -b_3x_1 \\
0 & b_2x_1 & b_3
\end{bmatrix},
\]
\[
u = \begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix}.
\]

(13)

(14)

(15)

(16)

\( \zeta \) as the first set of the inputs could be rewritten as follows:

\[
\zeta = \phi(\eta) = \begin{bmatrix}
k_1 & 0 & 0 & z_1 \\
0 & k_3 & 0 & z_3 \\
0 & 0 & k_5 & z_5
\end{bmatrix} = \begin{bmatrix}
k_1z_1 \\
k_3z_3 \\
k_5z_5
\end{bmatrix}, \text{ and } k_1, k_3, k_5 > 0
\]

(17)

Figure 2: Input membership functions for the system states.
On the other hand, the Lyapunov function is defined as below:

\[ V(\eta) = \frac{1}{2} \eta^T \eta = \frac{1}{2} (z_1^2 + z_2^2 + z_3^2). \]  

(18)

Calculating the derivative of the above equation and substituting Equations (11), (12) and (17) by this relation would give:

\[ \dot{V}(\eta) = z_1 \dot{z}_1 + z_2 \dot{z}_2 + z_3 \dot{z}_3 = z_1 z_2 + z_3 z_4 + z_5 z_6 = -k_1 z_1^2 - k_2 z_2^2 - k_3 z_3^2. \]

(19)

Hence,

\[ \dot{V}(\eta) < 0 \]

(20)

In addition, if a different articulation of the Lyapunov function is considered as follows:

\[ V_a = V(\eta) + \frac{1}{2} [\zeta - \phi(\eta)]^T [\zeta - \phi(\eta)] \]

(21)

\[ = \frac{1}{2} (z_1^2 + z_2^2 + z_3^2) + \frac{1}{2} ((z_2 + k_1 z_1)^2 + (z_4 + k_2 z_3)^2 + (z_6 + k_3 z_5)^2), \]

and if the control output is expressed as:

\[ u = G_a^{-1} \left[ \frac{\partial \phi}{\partial \eta} f + G \zeta - f_a - K (\zeta - \phi) \right] \text{ and } K = \begin{bmatrix} k_2 & 0 & 0 \\ 0 & k_4 & 0 \\ 0 & 0 & k_6 \end{bmatrix}. \]

(22)

then, the Lyapunov function’s derivative is negative definite, and the whole equation set would be stable. By substituting Equations (12) to (15), (17) and (19) in Equation (22), the control outputs would be acquired as follows:

\[ \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b_1 & 0 & b_3 \chi_3 \\ 0 & b_2 & -b_3 \chi_1 \\ 0 & b_2 \chi_1 & b_3 \end{bmatrix}^{-1} \begin{bmatrix} \dot{\chi}_{1d} + z_1 + k_1 z_2 + k_2 z_3 + k_3 z_4 + k_4 z_5 - \eta_1 x_4 + \eta_4 (x_4 + x_5 \chi_6) \Omega \\ \dot{\chi}_{3a} + z_3 + k_3 z_4 + k_4 z_5 + k_5 z_6 + k_6 z_7 - \eta_3 x_2 + \eta_5 (x_2 + x_3 \chi_6) \Omega \\ \dot{\chi}_{5a} + z_5 + k_5 z_6 + k_6 z_7 + k_7 z_8 - \eta_5 x_1 + \eta_7 (x_1 + x_2 \chi_4) \Omega \end{bmatrix}. \]

(23)

Furthermore, the above steps should be accomplished for the equations of the height position as well.

\[ \eta = \chi_7 d - \chi_7 \]

(24)

\[ \zeta = \chi_8 - \chi_8 \]

(25)

And

\[ \zeta = \phi(\eta) = -k_7 z_7, \quad k_7 > 0 \]

(26)

\[ V(\eta) = \frac{1}{2} z_7^2, \quad V_a = \frac{1}{2} z_7^2 + \frac{1}{2} (z_8 + k_8 z_8)^2 \]

(27)

\[ f = 0, \quad G = 1, \quad f_a = \chi_7 d + g, \quad G_a = -\frac{1}{m} \cos \chi_1 \cos \chi_3 \]

(28)

Finally, \( U_1 \) would be achieved as:

\[ U_1 = \frac{m}{\cos \chi_1 \cos \chi_3} (k_7 z_8 + z_7 + g \chi_7 d + k_8 z_8 + k_7 k_8 z_7) \]

(29)

### 3.2. Fuzzy tuning

Here, a fuzzy system is designed to tune the gains of the back-stepping controller, i.e., \( k_i \) (\( i = 1,2,3,\ldots,8 \)), to achieve better control performances in a variety of circumstances. It employs the singleton fuzzifier and the center average defuzzifier along with the product inference engine demonstrated in [65]. Thus, the following equation articulates fuzzy gains \( k_i^f \):

\[ k_i^f = \frac{\sum_{i=1}^{M} y_i^l \mu_{A_i^l}(|x_i|)}{\sum_{i=1}^{M} \mu_{A_i^l}(|x_i|)}, \quad i = 1,2,\ldots,8 \]

(30)

In which, \( M \) stands for the number of the fuzzy rules and is considered equal to 3, in this study. \( y_i^l \), which should be obtained by the optimization algorithm, refers to the centers of the output membership functions. Additionally, \( \mu_{A_i^l} \) denotes the input membership functions illustrated in Figure 2 for the rotational and translational states of the system. In this figure, \( A_i^l, i = 1,2,\ldots,8 \) and \( j = 1,2 \) are the centers of the membership functions determined in the optimization process for better tuning of the fuzzy system.
4. Optimization process

4.1. Grey wolf optimization

Inspired by social interactions, hunting mechanisms and leadership hierarchies of the grey wolves, the GWO method was initially presented by Mirjalili and Mirjalili [66]. In its mathematical model, the best solution is taken as alpha (α) grey wolf. Accordingly, the second and third fittest solutions are respectively called beta (β) and delta (δ) grey wolves, while the remaining solutions are considered as omega (ω) ones. In other words, after creation of a set of random positions within the search space, the three fittest positions are chosen as alpha, beta and delta wolves. Naturally, these three take the lead roles in the optimization process, while the omega wolves obey them. Other than the above mentioned leadership, their prey encirclement in hunting is modeled through the following equations suggested by Mirjalili et al. [66].

\[
\begin{align*}
\tilde{A} &= 2\tilde{a}r_1 - \tilde{a} \quad \text{(31)} \\
\tilde{C} &= 2\tilde{r}_2 \quad \text{(32)}
\end{align*}
\]

while, \(\tilde{A}\) and \(\tilde{C}\) denote the coefficient vectors, \(t\) signifies the current iteration, and \(\tilde{a}\) linearly reduces over the iterations from 2 to 0. At last, \(\tilde{r}_1\) and \(\tilde{r}_2\) represent two random vectors belong to [0, 1].

The simulation of the leadership and encircling processes could be formulated as follows:

\[
\begin{align*}
\tilde{D}_\alpha &= |\tilde{C}_1\tilde{X}_\alpha(t) - \tilde{X}(t)| \quad \text{(33)} \\
\tilde{D}_\beta &= |\tilde{C}_2\tilde{X}_\beta(t) - \tilde{X}(t)| \quad \text{(34)} \\
\tilde{D}_\delta &= |\tilde{C}_3\tilde{X}_\delta(t) - \tilde{X}(t)| \quad \text{(35)} \\
\tilde{X}_1 &= \tilde{X}_\alpha(t) - \tilde{A}_1\tilde{D}_\alpha \quad \text{(36)} \\
\tilde{X}_2 &= \tilde{X}_\beta(t) - \tilde{A}_2\tilde{D}_\beta \quad \text{(37)} \\
\tilde{X}_3 &= \tilde{X}_\delta(t) - \tilde{A}_3\tilde{D}_\delta \quad \text{(38)} \\
\tilde{X}(t + 1) &= \frac{\tilde{X}_1 + \tilde{X}_2 + \tilde{X}_3}{3} \quad \text{(39)}
\end{align*}
\]

in which, \(\tilde{X}(t)\) is the position vector belong to the regarded wolf at iteration \(t\). \(\tilde{X}_\alpha, \tilde{X}_\beta\) and \(\tilde{X}_\delta\) respectively denote the position vectors of the alpha, beta and delta wolves. Moreover, \(\tilde{A}\) and \(\tilde{C}\) are computed via Equations (31) and (32), respectively. The searching process executes with \(\tilde{A}\) which takes values smaller than -1 and larger than 1, to make the convergence or divergence operations, accordingly. Moreover, \(\tilde{C}\) selects random values within [0, 2], to emphasize \((\tilde{C} > 1)\) or deemphasize \((\tilde{C} < 1)\) the leaders by chance. At the beginning of the optimization process, a set of random positions is given to the search agents, and the three leaders are ascertained. Then, the positions of the omega wolves are updated via Equations (33) to (39) until the stopping criteria satisfied.

4.2. Multi-objective GWO

Generally, two ways are suggested for solving multi-objective problems. In the first way, a multi-criteria optimization problem is transformed into a single one, often by attaching weights to the objectives and summing them. Therefore, the considered single-objective solver must be run several times with various weights to obtain the corresponding design variables and objective functions. In the second way, there are approaches that address the problem by multi-objective algorithms and find a set of solutions via only one run of the algorithm. Here, by adding two operators to the GWO, it is extended as a multi-objective algorithm [67]. The first operator stores the non-inferior Pareto front solutions achieved in the optimization process in an external space, namely archive. The second one, however, helps us to select the leader wolves among the non-inferior solutions.
In fact, the concept of the archive is introduced to save the acquired non-inferior solutions as a simple storage module. Besides, the found non-inferior solutions would be compared with the archive members in each iteration. In this regard, one of the following cases may occur: (1) at least one of the archive members dominates the considered solution, which would not be permitted to come in; (2) at least one of the archive members is dominated by the new solution, which let it to replace the dominated solution(s) in the archive; (3) no inferiority is found in the comparison, and thus the new solution goes into the archive. In the case that the archive has the full capacity, the grid mechanism re-segments the search space to identify the most crowded segment and remove one of its solutions. Afterwards, it is possible to add the new solution to the archive.

As it was mentioned earlier, in order to converge the population to the global optimum solution, three leaders, i.e., $\alpha$, $\beta$ and $\delta$, should be assigned to guide other wolves in the searching domain. Nevertheless, in the multi-objective optimization process, the solutions are not simply comparable due to the concept of the Pareto optimality. To resolve this inconveniencing, a leader-selection mechanism based on the least crowded concept along with the roulette-wheel selection method is employed. In this regard, the relation for the probability of the roulette-wheel mechanism could be suggested as follows.

$$P_i = \frac{c}{N_i}$$  \hspace{1cm} (37)

in which, $c > 1$ stands a constant number, and $N_i$ signifies the number of the non-inferior solutions at segment $i$ of the Pareto front.

It is noteworthy that the most important benefits of the MOGWO are solution variety and strong convergence, respectively achieved through the proposed leader selection technique and the archive preservation mechanism [68-73]. However, if there exist more than two objectives for the optimization problem, similar to other Pareto-front-based algorithms, the efficiency of the MOGWO shrinks [67].

### 4.3. MOGWO for the controller design

In order to achieve the most suitable performance for the proposed back-stepping based fuzzy controller, the MOGWO is carried out to find the optimum values for the parameters of the fuzzy system ($\breve{Y}_i^l$ and $A_i^l$). The algorithm aims to minimize two objective functions, the system errors and control efforts, formulated as the sum of weighted integrals over time:

First objective $= w_1 \int |\phi|dt + w_2 \int |\psi|dt + w_3 \int |\theta|dt + w_4 \int |Z|dt$.  \hspace{1cm} (38)

Second objective $= w_5 \int |U_1|dt + w_6 \int |U_2|dt + w_7 \int |U_3|dt + w_8 \int |U_4|dt$  \hspace{1cm} (39)

while, $w_1, w_2, \ldots, w_8$ denote the weighting parameters used to close the effects of the different parts. Through the trial-and-error technique, $w_1, w_2$ and $w_3$ are assumed as 1; $w_4$ is considered as 0.3448; $w_5$ is set at $7.692 \times 10^{-4}$; and finally, $w_6, w_7$ and $w_8$ are fixed at 1.

### 5. Results and discussion

In the simulation process of behavior of the regarded UAV allocated with the introduced closed loop controller, the initial conditions of the system states are given according to Table 2, while their desired values are assumed to be zero. At first, the multi-objective grey wolf optimization algorithm is employed to determine the optimum values of design variables $\breve{Y}_i^l$ and $A_i^l$ considering two objective functions as the output errors and control efforts. Figure 3 depicts the acquired Pareto fronts corresponding to the fuzzy back-stepping (FBS) and back-stepping (BS) methods as well as the positions of the selected points with the design variables demonstrated in Table 3 and 4 for $\breve{Y}_i^l$ and $A_i^l$, respectively. The simulation results of the suggested fuzzy back-stepping (FBS) controller as well as back-stepping (BS) and fuzzy linear quadratic regulator (FLQR) suggested in [74] are exhibited in Figure 4. These graphs display the output variables of the UAV related to the suggested controller in
comparisons with those of the optimized simple back-stepping and FLQR [74] controllers. It is obvious that the controller of this study is preferred due to its rapid convergence with no over and undershoots. Figure 5 depicts the variations of the fuzzy gains in which all converge to constant values at about 0.25 (s).

To verify the robust and adaptive performance of the proposed method, the UAV is exposed to an external sinusoidal disturbance in $Z$ direction and a sinusoidal uncertainty on the overall mass as follows.

Sinusoidal disturbance: $\Delta = 20 \sin(6t)$  \hspace{1cm} (40)

Sinusoidal Uncertainty: $\dot{m} = m - 0.7 \sin(4t)$  \hspace{1cm} (41)

| Table 2: Initial conditions for the system states of the UAV. |
|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|
| System state    | $\varphi_0$  | $\varphi_0$  | $\theta_0$  | $\theta_0$  | $\psi_0$    | $\psi_0$    |
| Initial value   | $-\pi/9$     | 0            | $\pi/9$     | $\pi/9$     | 0            | $-1$        |

Figure 3: Pareto optimal fronts acquired through the optimization process based on the MOGWO algorithm.

| Table 3: Optimum values for centers of $\tilde{y}_{ij}$ determined by the selected point of the Pareto optimal front. |
|---|---|---|---|
| Index $i$ | $\tilde{y}_{1i}$ | $\tilde{y}_{2i}$ | $\tilde{y}_{3i}$ |
| 1  | 48.59129 | 46.3449 | 45.63673 |
| 2  | 40.83672 | 48.9994 | 48.8602 |
| 3  | 49.94538 | 41.74742 | 40.4601 |
| 4  | 49.16245 | 47.50848 | 47.18254 |
| 5  | 48.65325 | 47.20401 | 45.1284 |
| 6  | 49.02183 | 47.97964 | 47.85891 |
| 7  | 49.18363 | 42.93268 | 43.28514 |
| 8  | 49.98963 | 44.96797 | 45.26098 |
Table 4: Optimum values for $A_i^j$, determined by the selected point of the Pareto optimal front.

<table>
<thead>
<tr>
<th>Index $j$</th>
<th>$A_1^j$</th>
<th>$A_2^j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.24998</td>
<td>0.69439</td>
</tr>
<tr>
<td>2</td>
<td>1.378264</td>
<td>2.756528</td>
</tr>
<tr>
<td>3</td>
<td>0.269056</td>
<td>0.747378</td>
</tr>
<tr>
<td>4</td>
<td>1.445963</td>
<td>2.891927</td>
</tr>
<tr>
<td>5</td>
<td>0.260219</td>
<td>0.72283</td>
</tr>
<tr>
<td>6</td>
<td>1.482899</td>
<td>2.965798</td>
</tr>
<tr>
<td>7</td>
<td>0.433268</td>
<td>1.155382</td>
</tr>
<tr>
<td>8</td>
<td>2.956868</td>
<td>7.392169</td>
</tr>
</tbody>
</table>

While Figures 6 and 7 correspondingly illustrate the height of the quadcopter over time under the regarded disturbance and uncertainty, Figures 8, 10 and 12 show the changes of gains $k_7$ and $k_8$ and control effort $U_1$ over time for sinusoidal disturbance. Similarly, Figures 9, 11 and 13 illustrate the changes of $k_7$ and $k_8$ and $U_1$ for sinusoidal uncertainty of the overall mass over time. These figures clearly prove that the introduced back-stepping based optimal fuzzy controller is more successful, in the face of disturbances and uncertainties, in comparison with other methods. Generally, it could be said that coupling fuzzy logic based systems with back-stepping controller would result a more robust and adaptive technique in comparison with the simple back-stepping and FLQR reported in [74].

Figure 4: States of the simulated system with the suggested controller in comparison with the BS and FLQR [74].

Figure 5: variations of the fuzzy gains of the introduced controller over time.
6. Conclusion

In the present study, a novel optimal robust fuzzy back-stepping controller has been developed and executed on a nonlinear unmanned aerial vehicle system. First, the back-stepping approach has been adopted as the main method for ascertaining the control efforts of the system. Second, appropriate fuzzy systems have been utilized to tune the control gains and obtain robustness along with improved responses. Then, the control system has been optimized using the multi-objective grey wolf optimization algorithm to concurrently minimize the control efforts and system errors. Finally, simulation results have proved that the proposed method has a desirable capability with respect to the robustness and rapid convergence in the presence of disturbances and uncertainties when compared with previous research works. Besides, some future works could be suggested as follows.
(I) Considering more complete dynamical equations of the UAV having more degree-of-freedom.

(II) Employing the suggested optimal robust fuzzy back-stepping controller for other dynamical systems.

(III) Applying fractional-order calculus for derivative of the system states to make more flexibility on the controller.

(IV) Implementing structural and un-structural uncertainties on the UAV system to challenge the performance of the controller.

References


