Fuzzy Logic Control of a 1-DOF Helicopter System using The Bees Algorithm

Yunus Karaoğlu, Abdullah Çakan* and Mete Kalyoncu

Department of Mechanical Engineering, Konya Technical University, Konya, Türkiye

Abstract: The propeller-driven arm is required to achieve the desired angle while the thrust balances the varying gravitational load. The positioning control problem of a single-axis helicopter modeled as a nonlinear aeropendulum is addressed in this paper. Position error and angular velocity determine the thrust adjustment using a Sugeno fuzzy logic controller with gravitational load compensation as a feedforward input. The Bees Algorithm tunes the scaling factors and input membership-function positions using a combined objective of tracking error, actuator effort and command variations. The comparison uses a traditional proportional-integral-derivative controller and the initially designed fuzzy controller with a common thrust limit on the same plant. Further reference commands are applied to the selected controller without retuning. At larger reference commands, the tuned fuzzy controller settles faster with reduced transient oscillations compared with the initial design. The initially designed fuzzy controller gives lower tracking error at smaller commands.

Keywords: Aeropendulum, single-axis helicopter, fuzzy logic control, The Bees Algorithm, parameter optimization, reference tracking, nonlinear dynamics.

1. INTRODUCTION

An aeropendulum is a propeller-driven arm mounted on a fixed pivot. It provides a single-axis helicopter configuration in which thrust controls the angular position and supports the arm against gravity. This arrangement is also useful in control teaching because measured responses can be related directly to a mechanical model, feedback control and state estimation [1, 2]. The arm has only one rotational coordinate, although the control problem retains a nonlinear gravitational torque and thrust relation. The actuator limit restricts the available motion.

Job and Jose compare PID and LQR-based control using a mathematical aeropendulum model [3]. Lucena et al. describe the arm, motor and propeller together and estimate their parameters experimentally before designing local linear controllers [4]. Their steady-state thrust relation gives the actuator input for a desired force. Sampling introduces a further consideration: Habib et al. examine parameter estimation and stability under digital sampling and delayed feedback [5]. The model and feedback assumptions thus form part of the conditions for each control design.

Robust control based on identified models and second-order sliding mode control provide other approaches to angular regulation [6, 7]. Rafiuddin and Khan use data from real-time PID operation to develop a neural-network-based nonlinear controller [8]. Uyar determines the input through a gradient-based predictive controller for a single-axis copter [9]. In these applications, model information supports controller synthesis, supplies training data or predicts the effect of an input. The positioning task remains the same, but the design depends on how that information enters the feedback law.

Fuzzy logic describes the feedback action through rules. In Zadeh's fuzzy-set formulation, graded membership allows an input to activate neighboring linguistic regions [10]. Mamdani and Assilian use linguistic rules in an automatic controller [11]. The Takagi-Sugeno formulation combines numerical rule outputs into an input-output relation [12]. For position control, the rule conditions describe the current error and motion, with their outputs specifying the corresponding correction.

Fuzzy rules can define the feedback law or describe the plant used in controller design. In the hardware-in-the-loop helicopter study of Jancskar et al., phase-plane information guides the tuning of a fuzzy cascade controller [13]. Silva et al. use a Takagi-Sugeno representation to identify local aeropendulum models and design switched controllers [14]. The first approach places fuzzy rules in the feedback law. The second uses them to describe the plant for controller synthesis. These choices lead to different tuning problems even when both studies concern the same type of mechanical system.

*Address correspondence to this author at the Department of Mechanical Engineering, Konya Technical University, Konya, Türkiye;
E-mail: acakan@ktun.edu.tr

Received on 07-10-2026 Accepted on 19-11-2026 Published on 26-11-2026

E-ISSN: 3083-3450/2026

The performance of a fuzzy feedback law depends on the gains as well as the rule base. Carvajal et al. examine the performance and stability of a fuzzy PID structure for linear and nonlinear systems [15]. Jang's adaptive-network-based inference system learns a fuzzy input-output relation from data [16]. With prescribed rules, input scaling determines which memberships are active, their overlap determines the relative rule contributions and output scaling determines the size of the control action. These parameters act together, so changing one alters the effect of the others.

Optimization provides a means of choosing these interacting parameters. For helicopter control, Erdoğan et al. combine real-time identification with PID tuning by the Big Bang-Big Crunch algorithm to account for parametric uncertainty [17]. Rojas-Galván et al. use Hunting Search to tune membership parameters and output singletons in an adaptive fuzzy PID controller for an aeropendulum, with experimental implementation of the design [18]. The objective and test commands define what the search favors. A parameter set therefore needs evaluation over the command range of interest.

The Bees Algorithm searches the full parameter domain and recruits additional candidates around selected sites [19]. Promising neighborhoods receive more evaluations as random scouts explore elsewhere [20, 21]. Pham and Castellani examine the influence of algorithm settings and fitness landscapes through function-optimization comparisons [22]. In controller tuning, each candidate contains a parameter set and its closed-loop response determines the fitness.

Joint tuning of membership functions and scaling gains is an established application of The Bees Algorithm. Pham et al. use it for fuzzy control of an underactuated acrobatic robot [23]. Pham and Kalyoncu apply the algorithm to a flexible single-link robot arm [24]. Other gain-selection applications concern hierarchical PID control of a flexible manipulator [25], PID control of a robot arm with two degrees of freedom [26] and cascade proportional-derivative control of a flexible-link manipulator [27]. Each application gives the search a different plant and response objective.

This work considers one fuzzy parameter set for several reference angles. A fast approach reduces the early error but sometimes leaves a longer settling tail. The gravitational load changes with angle as well, which affects the response of a fixed controller. Tracking error, settling and actuator demand are therefore considered together across the reference commands.

The plant follows the nonlinear mechanical equations and steady-state thrust relation of Lucena et al. A zero-order Sugeno controller calculates the thrust correction from position error and angular rate. The Bees Algorithm selects the scaling gains and input membership locations while the rule consequents stay fixed. Gravity compensation supplies the equilibrium thrust at each reference angle. A weighted objective combines time-weighted error with actuator effort and command variation. The PID and initial fuzzy controllers provide comparisons on the same plant with a common thrust limit. Additional references test the selected settings between the angles that enter the search.

At larger commands, the optimized fuzzy controller has a shorter settling interval and less residual oscillation. The initial fuzzy controller gives lower accumulated tracking error at smaller commands. The comparison concerns nominal reference tracking with quasi-static thrust allocation.

The contribution is a joint tuning study with a fixed Sugeno rule base on the nonlinear aeropendulum model. Scaling gains and input membership locations are selected together, while gravity feedforward remains separate from the feedback correction. The objective combines tracking error, actuator effort and command variation under common response limits. Tests at the tuning angles and additional references then identify how the same parameter set changes the balance between tracking and actuator demand. The numerical records link the selected settings to their tracking and actuator histories.

2. MATERIALS AND METHODS

2.1. Mathematical Model

The aeropendulum model follows Lucena et al. [4]. The motor and propeller apply a force perpendicular to a rigid arm rotating about a fixed pivot (Figure **1**). With the angle measured from the downward vertical, the moment balance is

$$J\ddot{\theta} + C\dot{\theta} + mgd\sin\theta = LF. \quad (1)$$

Table **1** lists the model coefficients. Taking angular position and angular velocity as the states gives

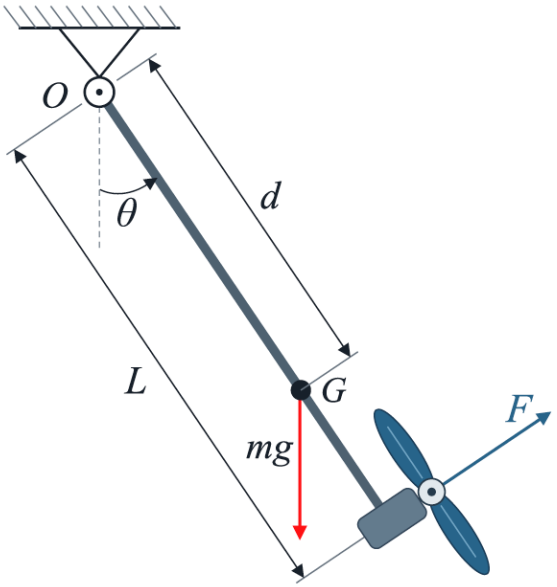

Figure 1: Schematic of the aeropendulum.

Table 1: Aeropendulum Model Parameters

Symbol	Description	Value
m	Effective pendulum mass	0.021 kg
d	Pivot-to-centre-of-mass distance	0.160 m
L	Thrust moment arm	0.230 m
J	Moment of inertia	5.481×10^{-4} kg m^2
C	Viscous damping coefficient	3.719×10^{-4} N m s/rad
g	Gravitational acceleration	9.806 m/s^2
K_0	Thrust-map coefficient	0.0072653
K_1	Thrust-map coefficient	0.5798
K_2	Thrust-map coefficient	0.3745
$u_{\max}$	Maximum duty cycle	60%
$F_{\max}$	Maximum applied thrust	0.1279025 N

$$\dot{\theta} = \omega. \tag{2}$$

$$\dot{\omega} = \frac{LF - C\omega - mgd\sin\theta}{J}. \tag{3}$$

The reduced source model assumes that motor and propeller transients are fast relative to the arm motion. Under this assumption, the steady-state thrust relation is

$$F = f_T(u) = K_0\left(\sqrt{K_1^2 + K_2 u} - K_1\right)^2. \tag{4}$$

Solving for the actuator command gives

$$u = f_T^{-1}(F) = \frac{\left(\sqrt{F/K_0} + K_1\right)^2 - K_1^2}{K_2}. \tag{5}$$

The input is the pulse-width-modulation duty cycle in percent. Its upper limit follows the calibration range of the source model, giving the thrust limit

$$F_{\max} = f_T(u_{\max}). \tag{6}$$

The controller requests thrust, which the inverse relation converts to an actuator command. The physical

coefficients come from the cited identification study and are not re-estimated here.

At a constant operating angle, equilibrium requires

$$\omega_0 = 0. \tag{7}$$

$$F_0 = \frac{mgd}{L}\sin\theta_0. \tag{8}$$

$$u_0 = f_T^{-1}(F_0). \tag{9}$$

Linearization about this equilibrium gives

$$J\Delta\ddot{\theta} + C\Delta\dot{\theta} + mgd\cos\theta_0\,\Delta\theta = L\Delta F. \tag{10}$$

The corresponding thrust-to-angle transfer function is

$$\frac{\Delta\Theta(s)}{\Delta F(s)} = \frac{L}{Js^2 + Cs + mgd\cos\theta_0}. \tag{11}$$

The operating angle enters the local restoring term through the linearization. The simulations use the original nonlinear moment balance throughout.

2.2. Fuzzy Logic Controller

Each fuzzy rule associates an input condition with a control action. More than one rule can be active because neighboring membership functions overlap. For the zero-order Sugeno controller, the rule outputs are constants and their firing strengths determine the weighted average [12]. The output changes continuously as the input moves from one membership region to the next.

Position error and angular rate form the controller inputs. The error determines the correction toward the reference and rate feedback moderates the arm motion. A separate feedforward input supplies the equilibrium thrust. Input scaling sets the membership regions reached by the feedback signals, while output scaling converts the fuzzy result into force. Figure **2** gives this structure and the parameters that The Bees Algorithm selects.

The position error is

$$e = r - \theta. \tag{12}$$

The equilibrium thrust at the reference angle provides the feedforward input:

$$F_{\mathrm{ff}} = \frac{mgd}{L}\sin r. \tag{13}$$

Scaling and clipping the error and negative angular rate gives the fuzzy inputs:

$$E = \mathrm{clip}(K_e e, -1, 1). \tag{14}$$

$$D = \mathrm{clip}(-K_{de}\omega, -1, 1). \tag{15}$$

Angular-rate feedback uses the arm motion directly and receives no derivative contribution from a reference step. Angle and rate measurements are ideal in the simulations.

The input labels are negative big, negative small, zero, positive small and positive big. The interior memberships are triangular and the end memberships have shoulder shapes. Their centres are

$$\mathbf{c}_E = (-1, -a_E, 0, a_E, 1). \tag{16}$$

$$\mathbf{c}_D = (-1, -a_D, 0, a_D, 1). \tag{17}$$

Linear interpolation between adjacent centres gives membership grades whose sum is unity across the clipped domain. Product conjunction defines the firing strength:

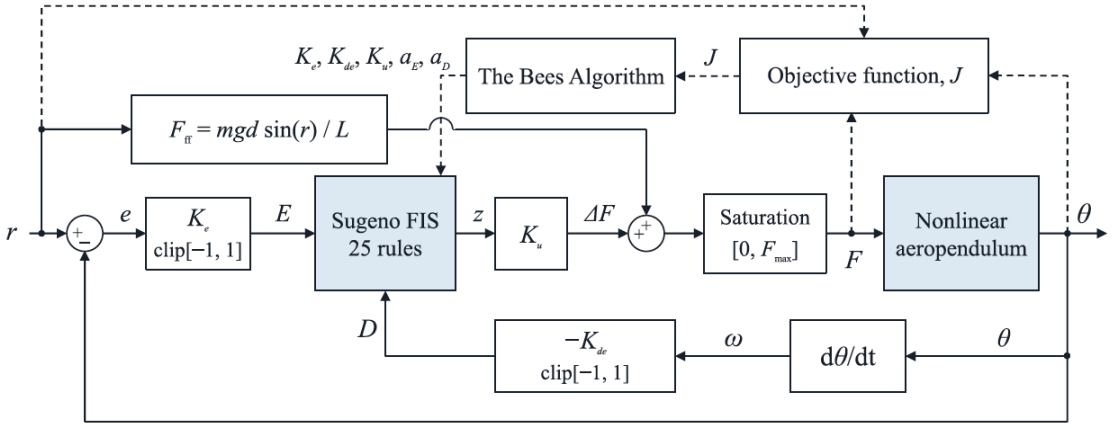

Figure 2: Control structure with The Bees Algorithm.

$$w_{ij} = \mu_i(E)\nu_j(D). \quad (18)$$

The constant rule consequents are

$$= q_{ij} = \mathrm{clip}\left(\frac{i+j-6}{2}, -1, 1\right). \quad (19)$$

The Sugeno weighted average [12] then gives

$$z = \frac{\sum_i \sum_j w_{ij} q_{ij}}{\sum_i \sum_j w_{ij}}. \quad (20)$$

Table **2** contains the fixed rule base. The search changes the gains and membership locations while leaving these consequents unchanged.

The fuzzy controller requests the thrust

$$F_{\mathrm{req}} = F_{\mathrm{ff}} + K_u z + K_i I. \quad (21)$$

Saturation enforces the available thrust range:

$$F = \min\{F_{\max}, \max(0, F_{\mathrm{req}})\}. \quad (22)$$

The optimized fuzzy controller uses the fuzzy correction and gravity compensation without integral action. The initial fuzzy controller also includes integral correction. Where this correction is present, conditional integration prevents the integral term from increasing saturation.

For a constant reference in the nominal model, gravity feedforward supplies the equilibrium thrust without an integral contribution. The optimized design therefore fixes the integral gain at zero as a controller-structure choice; the two zero bounds in Table **3** encode this constraint, while The Bees Algorithm tunes the other five parameters.

The PID controller shares the feed forward input and thrust limit. Its thrust request is

$$F_{\mathrm{req,PID}} = F_{\mathrm{ff}} + K_p e + K_i I - K_d \omega. \quad (23)$$

The PID and initial fuzzy gains remain fixed throughout the comparison.

These baseline gains are hand-specified nominal design settings for this study, rather than controller settings supplied by the model reference. The initial fuzzy design uses symmetric input partitions with equally spaced centres and the fixed consequents in Table **2**. Both baseline parameter sets give valid trajectories within the simulation envelope at the tested reference angles. No formal tuning rule or optimization procedure determines their gains. They therefore provide comparisons with prescribed controllers, with no claim that they represent the best achievable PID or initial fuzzy response.

2.3. The Bees Algorithm

The Bees Algorithm takes its search pattern from the foraging behavior of honeybees. Scouts explore the domain and recruitment directs a larger search effort

Table 2: Fuzzy Rule Consequents

$E \setminus D$	NB	NS	ZE	PS	PB
NB	−1	−1	−1	−0.5	0
NS	−1	−1	−0.5	0	0.5
ZE	−1	−0.5	0	0.5	1
PS	−0.5	0	0.5	1	1
PB	0	0.5	1	1	1

Table 3: Search Bounds and Initial Candidate

	K_e	K_{de}	K_u	K_i	a_E	a_D
Lower bound	0.5	0.05	0.005	0	0.15	0.15
Upper bound	12	3	0.15	0	0.85	0.85
Initial candidate	1	0.8	0.12	0	0.25	0.5

toward promising sites [19, 21]. Each site corresponds to a candidate solution with a fitness determined by the objective function. Neighborhood searches refine selected candidates as the remaining scouts explore other parts of the domain.

The choice of The Bees Algorithm rests on its direct treatment of bounded real parameters and its combination of local recruitment with continued global exploration. The closed-loop objective includes clipping, saturation and response penalties, so the search uses simulation values without requiring objective derivatives. Recruitment also gives explicit control over the evaluations assigned to promising neighborhoods. PSO and GA provide alternative search procedures for this problem. The present study evaluates the resulting controller settings and does not establish a performance ranking among optimizers.

Each candidate contains the fuzzy scaling gains and input membership locations. Simulations at the prescribed reference angles determine its fitness. Pham and Kalyoncu use The Bees Algorithm for a related fuzzy tuning problem on a flexible single-link robot arm [24]. The present search retains the rule consequents and follows the bounds in Table **3**.

The search ranks the candidates and assigns more recruits to elite sites than to the other selected sites. Each neighborhood retains its best member, including the incumbent. The remaining scouts explore the full bounded domain. This cycle repeats for the specified number of iterations (Figure **3**). Table **3** also gives the initial candidate.

The neighborhood radius contracts with iteration relative to the range of each parameter:

$$\rho_\ell = \max\{\rho_{\min}, \rho_0 \gamma^{\ell-1}\}. \tag{24}$$

$$h_{\ell,d} = \rho_\ell (p_d^{\max} - p_d^{\min}). \tag{25}$$

Local sampling is uniform within this half-width. Reflection returns coordinates that cross a bound to the search domain. The settings in Table **5** include the

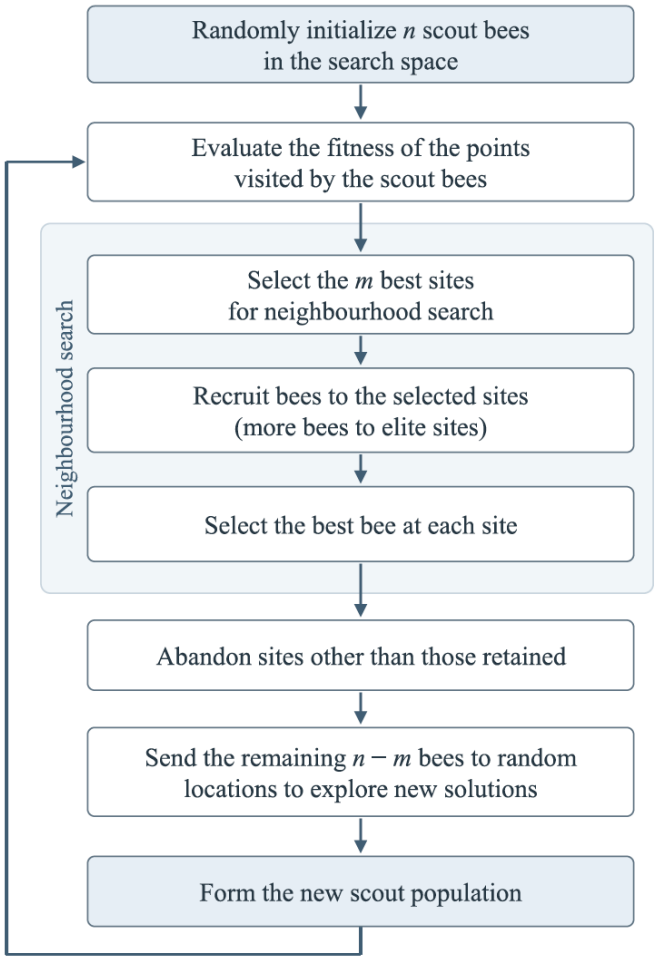

Figure 3: Flowchart of The Bees Algorithm.

incumbent in each recruitment count; its deterministic fitness is retained without another evaluation.

Each reference step starts from rest and gives a separate response for evaluating the candidate. The objective combines time-weighted tracking error, actuator effort and command variation:

$$J_0 = \frac{1}{n_r}\sum_{j=1}^{n_r}(w_1 A_j + w_2 B_j + w_3 H_j). \quad (26)$$

$$A_j = \frac{\int_0^T t|e_j(t)|\,dt}{r_j T^2/2}. \quad (27)$$

$$B_j = \frac{1}{T}\int_0^T \left(\frac{u_j(t)}{u_{\max}}\right)^2 dt. \quad (28)$$

$$H_j = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{u_j(k)-u_j(k-1)}{u_{\max}}\right)^2. \quad (29)$$

The time factor increases the cost of an error that persists late in the response. Normalization uses the error of an arm that remains at rest throughout the reference step. The other terms account for actuator effort and successive command changes. The inverse thrust relation converts each thrust history to actuator input, with the evaluation interval and sampling time setting the number of increments.

The response limits in Table **4** impose additional conditions on each candidate. The peak reference excess is

$$\delta_j = \max\{0, \theta_{\max,j} - r_j\}. \quad (30)$$

Adding the normalized limit violations gives

$$V = \sum_{j=1}^{n_r}(v_{1,j} + v_{2,j} + v_{3,j}). \quad (31)$$

The fitness used to rank the candidates is

$$J = \begin{cases} J_0, & V = 0, \\ J_0 + \lambda(1+V), & V > 0. \end{cases} \quad (32)$$

Fitness calculations use radians for angular quantities. For a response that remains unsettled at the end of the optimization interval, the settling penalty uses that interval as its settling time. Invalid trajectories receive the rejection cost in Table **5**. The penalty separates candidates that violate the limits from those whose response meets them; the weighted objective ranks candidates within the admissible set.

The simulations use fourth-order Runge-Kutta integration with constant thrust between controller updates. Each reference step starts from rest with zero integral state. MATLAB provides the numerical implementation. The search uses the optimization interval, while the performance tables use the shorter presentation interval.

Root-mean-square error measures tracking accuracy:

$$\mathrm{RMSE} = \frac{180}{\pi}\sqrt{\frac{1}{T_p}\int_0^{T_p} e\,(t)^2 dt}. \quad (33)$$

The percent-zovershoot measure is

$$M_p = 100\frac{\max(\theta_{\max} - r, 0)}{r}. \quad (34)$$

The RMS actuator command is

$$u_{\mathrm{rms}} = \sqrt{\frac{1}{T_p}\int_0^{T_p} u\,(t)^2 dt}. \quad (35)$$

Settling requires the sampled angle to remain within a two percent band around the reference until the end of the observation interval. Peak-angle evaluation includes integration points between controller updates. The integral measures use trapezoidal quadrature. Additional reference steps and a smaller integration step test the same selected parameters without retuning.

3. RESULTS AND DISCUSSION

Table **6** gives the three controller settings. Relative to the initial fuzzy controller, the optimized fuzzy

Table 4: Response Limits used During Parameter Selection

Response measure	Limit	Violation contribution
Peak reference excess	$\varepsilon_o = 10^{-8}$ deg	$v_{1,j} = \dfrac{\max(\delta_j - \varepsilon_o, 0)}{r_j}$
Absolute final error	$\varepsilon_e = 0.02$ deg	$v_{2,j} = \dfrac{\max(\lvert e_j(T)\rvert - \varepsilon_e, 0)}{r_j}$
Settling time	$t_{\lim} = 3$ s	$v_{3,j} = \dfrac{\max(t_{s,j} - t_{\lim}, 0)}{T}$

Table 5: Search and Simulation Settings

Setting	Value
Iterations	50
Scouts / selected sites / elite sites	40 / 12 / 4
Candidates per elite / other selected site	10 / 5
Initial / minimum relative radius	0.20 / 0.02
Radius reduction factor	0.95
Random-number seed	20261103
Objective evaluations	4840
Training references	20, 40, 60 deg
Additional references	30, 50 deg
Optimization / presentation interval	8 / 3 s
Controller sampling / integration step	0.01 / 0.005 s
Extended check interval / integration step	20 / 0.0005 s
Objective weights w_1 / w_2 / w_3	1 / 0.01 / 0.001
Penalty coefficient λ	1000
Invalid-trajectory cost	10^6
Angular-position / rate validity bounds	90 deg / 100 rad/s

Table 6: Controller Parameters used in the Comparison

Parameter	PID	FUZZY	FUZZY-BA
K_p	0.08	-	-
K_i	0.015	0.008	0
K_d	0.018	-	-
K_e	-	3	0.761011
K_{de}	-	0.6	0.489715
K_u	-	0.12	0.146650
a_E	-	0.5	0.151939
a_D	-	0.5	0.521227

controller has a smaller error scaling gain and a larger output gain. Its interior error membership centres lie closer to zero in normalized coordinates, with less change in rate spacing. Its equilibrium input comes from gravity compensation, since it has no integral contribution.

The fuzzy-controller comparison includes the removal of integral action as well as changes in gains and membership locations. The response differences therefore characterize the selected controller configurations, with no separate estimate of the effect of each change.

The best retained fitness decreases during the search in Figure **4**. The plotted quantity includes the penalty whenever response limits are violated. For feasible candidates, it is the weighted tracking and actuator cost. Retaining the incumbent at each selected site prevents the search from losing its current best solution.

The error and rate partitions change by different amounts (Figures **5** and **6**). Narrower central error regions alter the active rule weights near zero in the normalized domain. The singleton locations remain fixed in Figure **7**; their contribution to thrust depends on the output gain. The surface in Figure **8** contains the fuzzy correction, with gravity compensation added separately.

Figure **7** places the five consequent values on the vertical axis against their linguistic output labels. These values are the distinct levels appearing in the rule table. Figure **8** instead uses physical tracking error and error rate on its horizontal axes. For a constant reference, the error rate equals the negative arm angular rate. The plotted coordinates pass through the

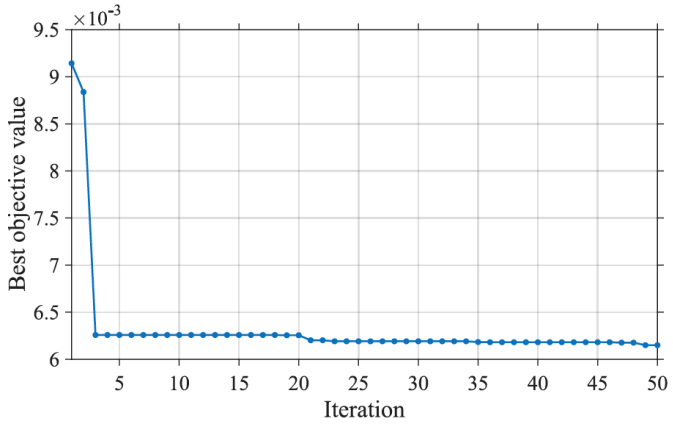

Figure 4: Convergence of The Bees Algorithm.

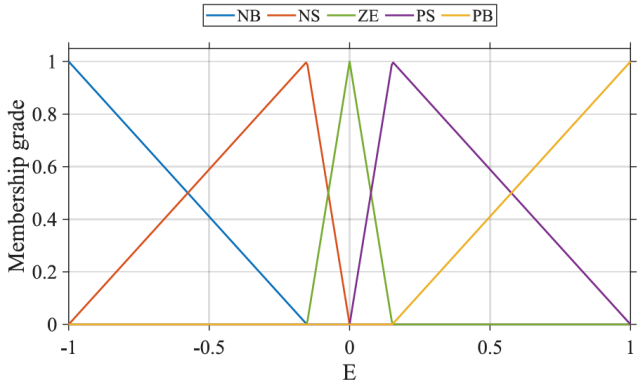

Figure 5: Input membership functions for tracking error.

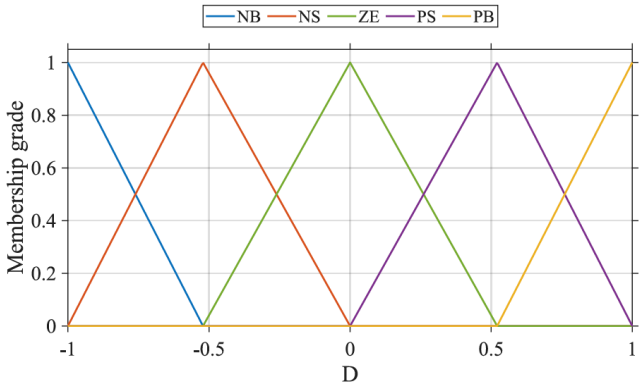

Figure 6: Input membership functions for angular-rate feedback.

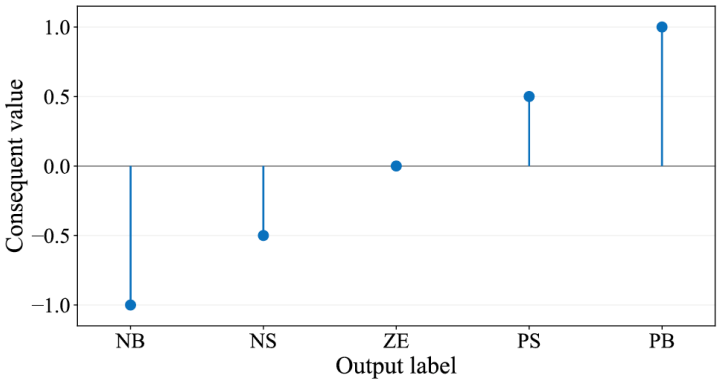

Figure 7: Output singletons of the fuzzy logic controller.

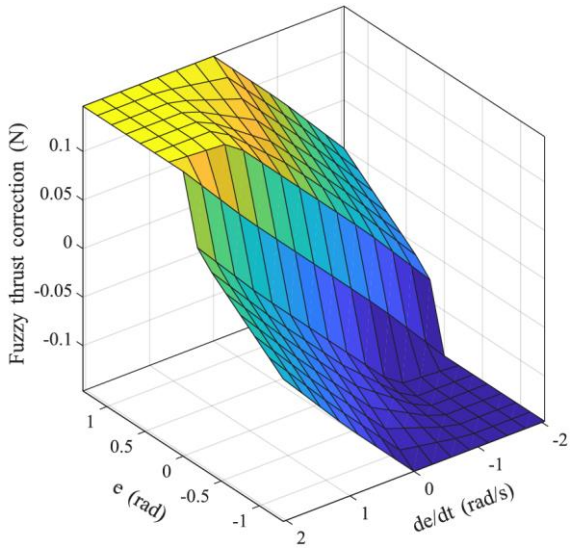

Figure 8: Control surface of the optimized fuzzy controller.

same input gains and clipping operations as the controller, giving the normalized inputs defined in Eqs. (14) and (15). The vertical coordinate is the fuzzy output after output scaling, in units of thrust. This surface excludes the separate feedforward contribution.

Reaching the reference first does not always give the shortest settling time in Figure **9** to **11**. The initial motion and the later oscillation contribute differently to the measures in Table **7**. All three controllers use the same observation interval for this comparison.

RMSE evaluates the complete error history but is not the quantity minimized during tuning. The objective weights absolute error by time and also accounts for actuator demand and command changes. Response-limit penalties further restrict the selected transient. Error near the start therefore has less weight in the tracking objective than error of the same magnitude later in the response. A more gradual approach can increase RMSE while reducing the settling tail. The trade-off is visible at the intermediate command, where shorter settling accompanies a larger RMSE.

Settling time uses a $\pm 2\%$ band around the reference and requires the sampled response to remain inside it until the end of the observation interval. All measures use the common three-second interval.

The initial fuzzy controller gives the lowest tracking error at the smallest command (Figure **9**). Its response reaches the reference slightly sooner and needs little correction afterward. The optimized fuzzy controller takes a more gradual path with slightly lower actuator effort. The PID response retains a longer oscillatory tail.

At the intermediate command, the initial fuzzy controller reaches the reference earlier but then crosses it before settling (Figure **10**). The optimized

Table 7: Tracking Performance for the Principal Reference Steps

Reference (deg)	Controller	RMSE (deg)	M_p (%)	t_s (s)	u_{rms} (%)
20	PID	3.730	23.650	0.85	25.738
20	FUZZY	3.376	0.198	0.44	25.546
20	FUZZY-BA	3.617	0.000	0.46	25.255
40	PID	7.609	22.551	0.90	44.192
40	FUZZY	7.354	15.504	0.59	44.367
40	FUZZY-BA	8.110	0.000	0.53	43.054
60	PID	12.299	20.008	1.16	56.784
60	FUZZY	12.344	24.401	1.12	56.912
60	FUZZY-BA	12.167	0.000	0.34	56.444

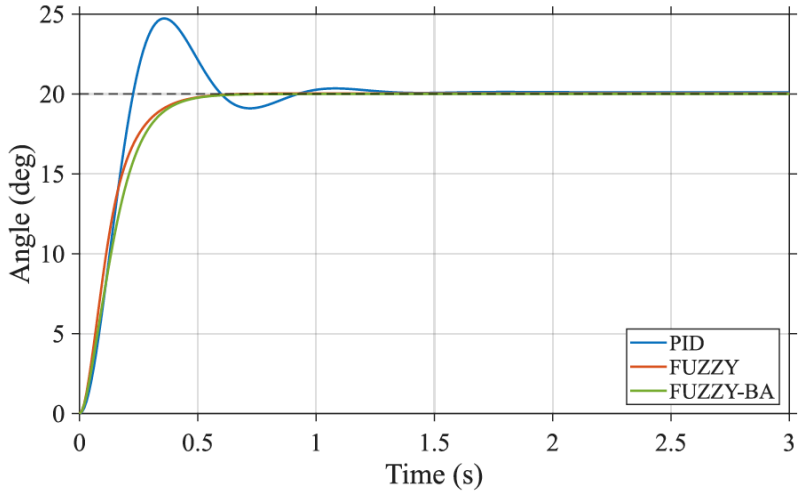

Figure 9: Tracking responses for the 20 degree reference

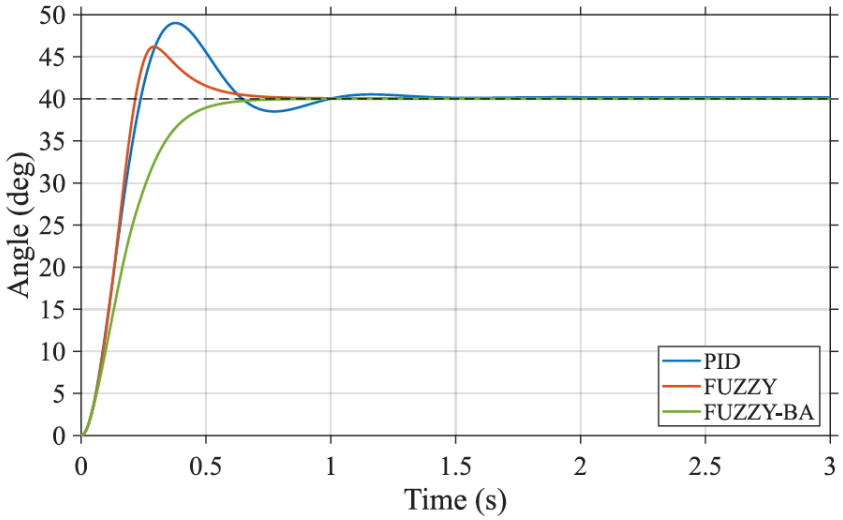

Figure 10: Tracking responses for the 40 degree reference.

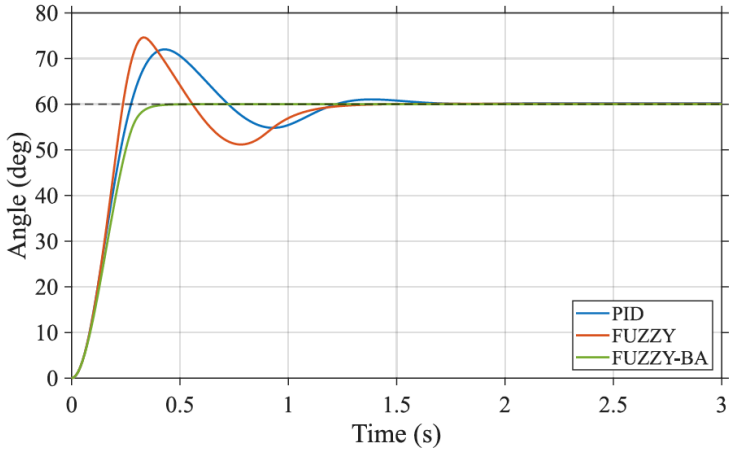

Figure 11: Tracking responses for the 60 degree reference.

fuzzy response approaches more slowly and stays close to the reference once it arrives. Its settling time and RMS actuator demand are lower, although its RMSE is higher. The additional error during the approach outweighs the benefit in the later part of the response for this error measure.

The optimized fuzzy controller has its clearest settling advantage at the largest command (Figure **11**). The PID and initial fuzzy responses continue to oscillate after first reaching the reference. The optimized response enters the settling band without that later oscillation. Its tracking error is also slightly lower than that of the initial fuzzy controller, with a modest reduction in actuator effort.

The error histories separate these later transients in Figure 12. Error approaches zero for the optimized fuzzy controller, while repeated zero crossings mark the decaying motion of the PID and initial fuzzy responses. These zero crossings accompany the longer settling tail. The accumulated-error measures in Table **7** also include the initial approach.

Both fuzzy controllers start at the upper duty-cycle limit in Figure **13** to **15**. Their peak demand is the same, so the lower RMS command of the optimized fuzzy controller comes from the subsequent input history. For the larger commands, the initial fuzzy controller reduces the input more deeply before returning toward equilibrium. The optimized controller returns with less oscillation. Table **7** accounts for the whole history, including the initial saturated input.

The load changes across these commands because the reference angle sets the equilibrium thrust and local gravitational stiffness. Feedforward supplies the equilibrium force, while the fuzzy correction drives the motion under the thrust limit. The same parameter set therefore acts on different operating conditions as the command changes.

The additional references in Table **8** lie between the training angles and do not enter the search. The parameters remain fixed for these tests. The initial fuzzy controller gives lower tracking error at both angles. It also settles sooner at the smaller additional command, while the optimized fuzzy controller settles sooner at the larger one.

Settling time uses a $\pm 2\%$ band around the reference and requires the sampled response to remain inside it until the end of the observation interval. All measures use the common three-second interval.

The extended reference sweep preserves the same qualitative behavior. A smaller integration step gives closely matching angular trajectories with the controller sampling time unchanged. Agreement between the two integrations supports the numerical consistency of the responses.

Fuzzy cascade and switched-model control provide related treatments of single-axis positioning [13, 14]. The Hunting Search aeropendulum study also uses search-based fuzzy tuning [18]. The present application keeps the rule base fixed and uses The Bees Algorithm to select gains and membership locations together. The resulting setting favors settling at larger commands, with a tracking-error cost at smaller and intermediate commands.

The scope of this comparison follows the settings in Table **6**. Integral action is present in the initial fuzzy controller and absent from the optimized fuzzy controller. The response difference includes that

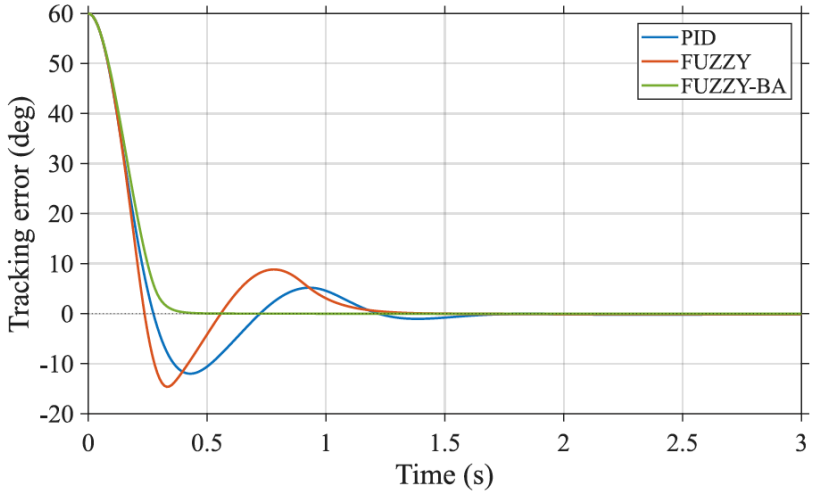

Figure 12: Tracking error for the 60 degree reference.

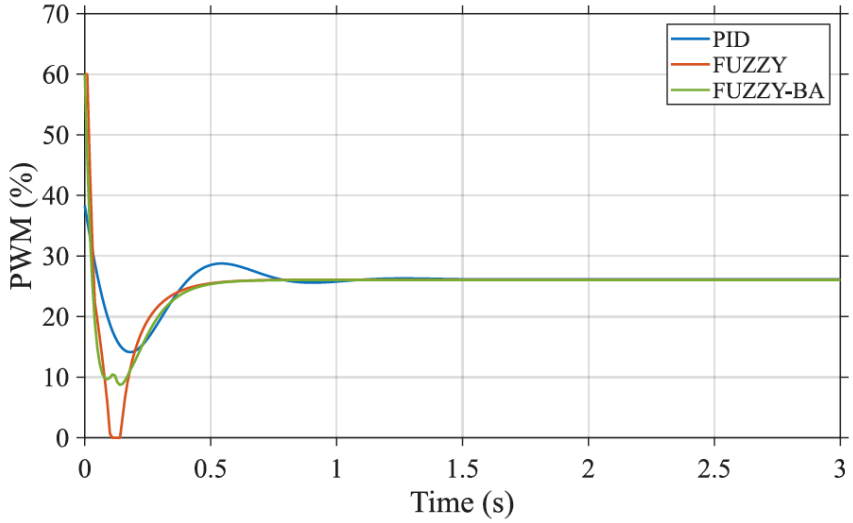

Figure 13: Actuator commands for the 20 degree reference.

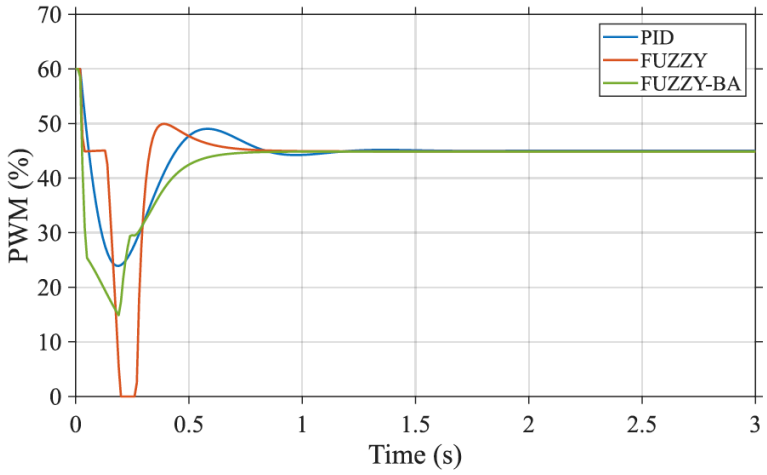

Figure 14: Actuator commands for the 40 degree reference.

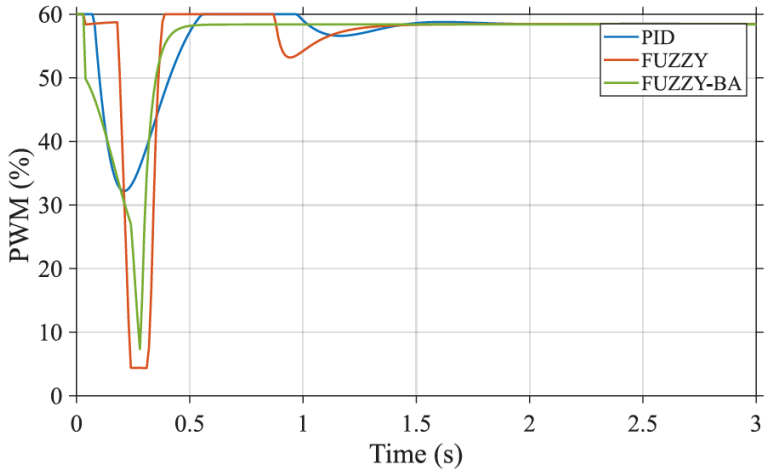

Figure 15: Actuator commands for the 60 degree reference.

Table 8: Performance at Reference Angles Excluded from Optimization

Reference (deg)	Controller	RMSE (deg)	M_p (%)	t_s (s)	u_{rms} (%)
30	PID	5.623	23.321	0.87	35.600
30	FUZZY	5.188	5.686	0.45	35.566
30	FUZZY-BA	5.915	0.000	0.51	34.678
50	PID	9.828	21.427	0.95	51.352
50	FUZZY	9.672	20.855	0.66	51.789
50	FUZZY-BA	10.076	0.000	0.48	50.424

change together with the gain and membership settings. The baseline gains also remain fixed without an equivalent global search. Ideal feedback and quasi-static thrust allocation define the model assumptions. Transfer to hardware requires assessment of motor transients, disturbances and model mismatch.

In a practical implementation, angle noise perturbs the error input and noise in the angular-rate signal can produce rapid changes in the thrust request. A rate estimate obtained by differentiation needs filtering, whose delay also affects the transient. Input clipping bounds the normalized signals but does not remove noise within that range. Errors in the mass distribution or thrust calibration also change the equilibrium input supplied by gravity compensation. Since the optimized design has no integral correction, a constant mismatch can leave a position offset. Changes in inertia and damping alter the transient as well. These are expected implementation effects under the model equations; the present simulations do not quantify their magnitude.

4. CONCLUSION

The Bees Algorithm selects scaling gains and input membership locations for a fixed-rule Sugeno controller on the nonlinear aeropendulum model. The optimized fuzzy controller settles sooner at the larger reference commands and uses slightly less RMS actuator effort at the principal references. The initial fuzzy controller retains lower tracking error at smaller commands because of its faster approach.

At the largest reference of 60 degrees, settling takes 0.34 s for the optimized fuzzy controller, compared with 1.12 s for the initial fuzzy controller and 1.16 s for PID under the same two percent criterion. This difference concerns the settling tail; the tracking-error advantage is smaller and does not extend to every reference angle.

One parameter set also controls the additional reference angles without retuning. The error advantage

of the optimized fuzzy controller over the initial fuzzy controller occurs only at the largest principal command, so the results support a command-dependent improvement in the transient. A finer integration step leaves the trajectories closely matched. These conclusions apply to nominal tracking with ideal feedback and a static thrust relation.

The next validation step concerns hardware operation with motor dynamics, measurement noise and external disturbances. Extending the search to the rule consequents also provides a separate design question: whether additional rule freedom improves the tracking and actuator balance beyond gain and membership tuning alone.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

Yunus Karaoğlu writes the original draft. Abdullah Çakan prepares the visualizations and reviews and edits the manuscript. Mete Kalyoncu contributes software and supervision and reviews and edits the manuscript.

REFERENCES

[1] Neto RC, Trindade FLA, Marques BRA, et al. An Aeropendulum-Based Didactic Platform for the Learning of Control Engineering. Journal of Control, Automation and Electrical Systems 2023; 34(3): 566-77. https://doi.org/10.1007/s40313-022-00981-4

[2] Enikov ET, Iriqui JA. From Step-Response to State-Space Controller-Observer Design in Twenty Minutes: A Hands-On Workshop on the Use of Matlab/Simulink to Control a Low-Cost Aerodynamic Pendulum. In: 2013 Pacific Southwest Section Meeting Proceedings. ASEE Conferences 2013; p. 14.

[3] Job MM, Jose PSH. Modeling and control of mechatronic aeropendulum. In: 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE 2015. p. 1-5. https://doi.org/10.1109/ICIIECS.2015.7192959

[4] Lucena ER, Luiz SOD, Lima AMN. Modeling, parameter estimation, and control of an aero-pendulum. In: Proceedings of the XV Simposio Brasileiro de Automacao Inteligente 2021. p. 1984-9.

[5] Habib G, Miklos A, Enikov ET, et al. Nonlinear model-based parameter estimation and stability analysis of an aero-pendulum subject to digital delayed control. International Journal of Dynamics and Control 2017; 5(3): 629-43. https://doi.org/10.1007/s40435-015-0203-0

[6] Breganon R, Alves UNLT, De Almeida JPLS, et al. Model Identification and H_∞ Control of an Aeropendulum. Journal of Applied Research and Technology 2023; 21(4): 526-34. https://doi.org/10.22201/icat.24486736e.2023.21.4.1741

[7] Tanveer MH, Koduru C, Roy RG, et al. A Robust Control Technique for Pitch Control of an Aeropendulum. In: 2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE). IEEE 2023. p. 257-61. https://doi.org/10.1109/RCAE59706.2023.10398789

[8] Rafiuddin N, Khan YU. Nonlinear controller design for mechatronic aeropendulum. International Journal of Dynamics and Control 2023; 11(4): 1662-70. https://doi.org/10.1007/s40435-022-01080-7

[9] Uyar O. Gradient-Based MPC Controller for One DOF Copter Control. International Journal of Applied Methods in Electronics and Computers 2024; 12(4): 97-103. https://doi.org/10.58190/ijamec.2024.110

[10] Zadeh LA. Fuzzy sets. Information and Control. 1965; 8(3): 338-53. https://doi.org/10.1016/S0019-9958(65)90241-X

[11] Mamdani EH, Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies. 1975; 7(1): 1-13. https://doi.org/10.1016/S0020-7373(75)80002-2

[12] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics. 1985; SMC-15(1): 116-32. https://doi.org/10.1109/TSMC.1985.6313399

[13] Jancskar I, Sári Z, Schiffer A, et al. Phase plane tuning of fuzzy controller for 1 DoF helicopter model. Pollack Periodica 2015; 10(2): 3-15. https://doi.org/10.1556/606.2015.10.2.1

[14] Silva HRM, Ramos ITM, Cardim R, et al. Identification and Switched Control of an Aeropendulum System. In: Anais do Congresso Brasileiro de Automatica 2020 2020; p. 6.

[15] Carvajal J, Chen G, Ogmen H. Fuzzy PID controller: Design, performance evaluation, and stability analysis. Information Sciences 2000; 123(3–4): 249-70. https://doi.org/10.1016/S0020-0255(99)00127-9

[16] Jang JSR. ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics. 1993; 23(3): 665-85. https://doi.org/10.1109/21.256541

[17] Erdoğan E, Soylu A, Şeker ŞS, et al. Real-Time Identification and Self-Tuning PID Controller for 1-DOF Helicopter System. In: 2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA). IEEE 2025; pp. 1-4. https://doi.org/10.1109/ICHORA65333.2025.11017316

[18] Rojas-Galván R, García-Martínez JR, Cruz-Miguel EE, et al. Hunting Search Algorithm-Based Adaptive Fuzzy Tracking Controller for an Aero-Pendulum. Technologies 2024; 12(5): 63. https://doi.org/10.3390/technologies12050063

[19] Pham DT, Ghanbarzadeh A, Koç E, et al. The Bees Algorithm: A Novel Tool for Complex Optimisation Problems. In: Pham DT, Eldukhri EE, Soroka AJ, editors. Intelligent Production Machines and Systems. Elsevier 2006; pp. 454-9. https://doi.org/10.1016/B978-008045157-2/50081-X

[20] Pham DT, Castellani M. The Bees Algorithm: Modelling foraging behaviour to solve continuous optimization problems. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2009; 223(12): 2919-38. https://doi.org/10.1243/09544062JMES1494

[21] Castellani M, Pham D. The Bees Algorithm: A Gentle Introduction. In: Pham DT, Hartono N, editors. Intelligent Manufacturing and Production Optimisation: The Bees Algorithm Approach. Cham: Springer 2022. p. 3-21. https://doi.org/10.1007/978-3-031-14537-7_1

[22] Pham DT, Castellani M. A comparative study of The Bees Algorithm as a tool for function optimisation. Cogent Engineering 2015; 2(1): 1091540. https://doi.org/10.1080/23311916.2015.1091540

[23] Pham DT, Darwish AH, Eldukhri EE. Optimisation of a fuzzy logic controller using The Bees Algorithm. International Journal of Computer Aided Engineering and Technology 2009; 1(2): 250-64. https://doi.org/10.1504/IJCAET.2009.022790

[24] Pham DT, Kalyoncu M. Optimisation of a fuzzy logic controller for a flexible single-link robot arm using The Bees Algorithm. In: 2009 7th IEEE International Conference on Industrial Informatics. IEEE 2009. p. 475-80. https://doi.org/10.1109/INDIN.2009.5195850

[25] Fahmy AA, Kalyoncu M, Castellani M. Automatic design of control systems for robot manipulators using The Bees Algorithm. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 2012; 226(4): 497-508. https://doi.org/10.1177/0959651811425312

[26] Karakoyun E, Çakan A, Kalyoncu M. Determination of PID control parameters for position control of a 2-DOF robot arm using The Bees Algorithm (BA). Journal of Balıkesir University Institute of Science and Technology 2022; 24(1): 111-24. In Turkish. https://doi.org/10.25092/baunfbed.885152

[27] Gümüş MS, Çakan A, Kalyoncu M. Cascade proportional derivative controller for a flexible link robot manipulator using The Bees Algorithm. Academic Platform Journal of Engineering and Smart Systems 2023; 11(1): 27-34. https://doi.org/10.21541/apjess.1084875

https://doi.org/10.65904/3083-3450.2026.02.05

© 2026 Karaoğlu *et al*.
This is an open access article licensed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution and reproduction in any medium, provided the work is properly cited.