Fuzzy Controller Optimized by the African Vultures Algorithm for Trajectory Tracking of a Two–Link Gripping Mechanism

Radiša Ž. Jovanović
Full Professor
University of Belgrade
Faculty of Mechanical Engineering

Uglješa S. Bugarić
Full Professor
University of Belgrade
Faculty of Mechanical Engineering

Mitra V. Vesović
Teaching and Research Assistant
University of Belgrade
Faculty of Mechanical Engineering

Natalija B. Perišić
Junior Research Assistant
University of Belgrade
Faculty of Mechanical Engineering

This paper presents the proportional–derivative fuzzy controller for trajectory tracking of the gripping mechanism with two degrees of freedom. Aiming to achieve movement of the gripping mechanism without sudden starting and stopping, a polynomial velocity profile is utilized. The African vultures optimization, as one of the latest metaheuristic algorithms, is used to obtain the optimal input/output scaling gains of the proposed fuzzy controller according to the selected fitness function. The results obtained by this algorithm are compared with the other three new and popular metaheuristic algorithms: the whale optimization, the ant lion optimization and the sine cosine algorithm. Moreover, a simulation study was done for the defined initial position and for the scenario where there is a certain deviation because the gripping mechanism is not at its original initial position. Finally, the robustness of the controller is tested for the case when the masses of the segments increase three times. The results revealed that the suggested controller was capable of dealing with nonlinearities of the gripping mechanism, initial position and parameter changes. The movement of the gripping mechanism is smooth and follows the defined trajectory.

Keywords: *gripping mechanism, trajectory tracking, fuzzy control, the African vultures optimization, metaheuristic optimization.*

1. INTRODUCTION

Intelligent mobile robots may be employed for a variety of tasks, including internal transport and material hand–ling in the manufacturing process. In such instances, an intelligent mobile robot can be thought of as a type of transportation equipment. Single or complex, the opera–ting cycle of all transportation devices is the most important feature. Furthermore, in this paper, intelligent mobile robots are regarded as "single-position mac–hines" with a discontinuous working regime. A single position machine is a machine that only handles one piece of product at a time and keeps it on the machine the entire time during handling [1] .

An intelligent mobile robot's single working cycle consists of the following steps: 1) Robot movement - from the starting point to a position in front of the production machine, within reach of the gripping mechanism. From this position the transportation unit can be caught; 2) Gripping mechanism movement - from the initial (transport) to the position required for capturing the transportation unit; 3) Transportation unit capture 4) Reverse movement of the gripping mechanism with the transportation unit – from the capturing position to the initial (transport) position; 5) Reverse robot movement - back to the starting point from the position in front of the manufacturing machine. 6) Repetition of activities 2), 3) and 4), but this time the transportation unit releases rather than captures. If the reverse robot movement does not end at the same position as it began, the working cycle is referred to as complex [1,2].

In general, the robot's mobility in the environment is realized using preprocessed images received by a stereo vision system, and the present state of the robot is determined using the specified optimal path based on a given criterion. This research considers only a segment of an intelligent machine's working cycle, namely the movement of the gripping mechanism.

A gripping mechanism is a device that allows an object to be captured, handled, released, and tightened by a robot. To imitate human hand movements, modern robotics and their grippers use integrated mechanisms and controls. Since the invention of the first Stanford controllable gripper, robotic arms have been one component of an automated system that has been around for more than fifty years.

Many of the gripper's design and control aspects are being employed today [3]. Traditional feedback controls (PID-like controls) [4], adaptive backstepping slide mode control [5], robust control [6] and [7], feedback linearization control [8] and many more techniques, as well as combinations of prior techniques, can and are used to control the movement of the gripping mechanism.

The fuzzy logic controller (FLC) is a widely used control approach for nonlinear and complicated systems due to its high performance. The Mamdani or Takagi-

Received: May 2022, Accepted: July 2022
Correspondence to: Dr Radiša Jovanović
Faculty of Mechanical Engineering,
Kraljice Marije 16, 11120 Belgrade 35, Serbia
E-mail: rjovanovic@mas.bg.ac.rs
doi:10.5937/fme2203491J

Sugeno fuzzy systems are common in variable fuzzy PID and proportional-derivative (PD) controllers [9]. The advantages of a fuzzy PID controller for trajectory tracking control of a mobile robot, and its gripping mechanism are paramount in its rapidity, stability, anti-interference and tracking precision [2].

To project the fuzzy PID controller a trial-and-error approach can be used but, for the more complicated challenges, an optimization technique is required. As one of the modern optimization approaches, meta–heuristic algorithms are exploited to solve nonconvex, nonlinear, and multimodal problems with linear or nonlinear cons–traints and continuous or discrete vari–ables. For exam–ple, the Particle Swarm Optimization Algorithm has been effectively used to solve the wind farm layout design problem [11], or the Dingo Optimi–zation Algorithm for solving continuous engineering problems [12]. Further, an optimal load frequency control is optimally designed using the African Vultures Optimization Algorithm (AVOA) [12]. The optimum design for robot arm point-to-point trajectory planning and movement was achieved using the ant lion optimizer (ALO) [14].

The parameters of the FLC were optimized with particle swarm [15] and genetic [16] optimization algorithms for achieving a specific trajectory in a robot movement in terms of control precision and conver–gence speed. Optimal path search and control of mobile robot was also investigated using a hybridized sine-cosine algorithm (SCA) and ant colony optimization technique [17]. The whale optimization algorithm (WOA), as a novel optimization technique for solving optimization problems defined in [18] is utilized to find the parameters of FLC in the trajectory tracking control of a robot arm with two degrees of freedom [2].

Design and development of robot arm system using machine vision [19], or using the controller based on a programmable System-On-Chip device [20] was the primary topic in many studies.

The main goal of this paper is to design a fuzzy PD controller of a two-link gripping mechanism that ope–rates within a mobile robot working cycle. Proposed fuzzy controller is nonlinear, which makes it very sui–table for controlling complex and nonlinear systems, such as the gripping mechanism. In general, fuzzy controllers have multiple parameters that must be adjusted in order to obtain satisfactory dynamic beha–viour. This is often done by trial-and-error method. In this paper, the AVOA algorithm, as a novel technique for solving optimization problems, is used to determine the proper parameters of FLC in the polynomial trajectory tracking control. As practical applications re–quire the robot to move smoothly, the polynomial velocity profile is used to shift the gripping mechanism without abrupt starting and stopping. Obtained results are compared with three different optimization algo–rithms: the WOA, the ALO and the SCA. The suggested controller's robustness was further tested for altering initial conditions and changing link masses.

2. DYNAMICS OF A TWO-LINK GRIPPER

Figure 1 depicts the real object – a mobile robot with a gripper. A gripper on the robot can be described as a two-degree-of-freedom (2DOF) mechanism, which is approximated by the scheme portrayed in Figure 2.

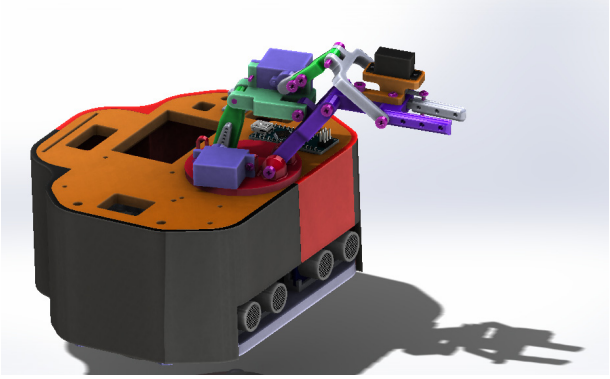

Figure 1. Mobile robot with gripping mechanism

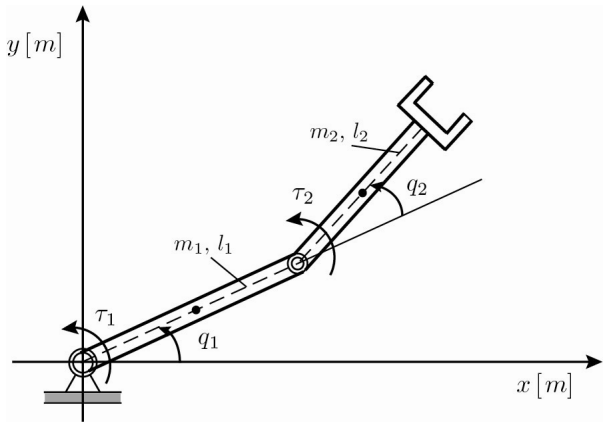

Figure 2. Scheme of the robotic gripper

The link angle, the length and the mass of the *i*-th link, are respectively q_i, l_i, and m_i, for i = 1,2. Introducing the assumption that the centers of masses are in the middle of levers and without considering the friction, the dynamic model of a rigid two-link robot can be written as follows [21]:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) = \tau, \tag{1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^{2x1}$ are the robotic link position, the velocity and the acceleration vector, respectively; $\boldsymbol{\tau} \in \mathbb{R}^{2x1}$ is the torque input vector; $M(\boldsymbol{q}) \in \mathbb{R}^{2x2}$ is the positive definite inertia matrix; $C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{2x2}$ is the centripetal Coriolis force matrix; and $G(\boldsymbol{q}) \in \mathbb{R}^{2x1}$ is the gravitational vector. The elements $M_{ij}(\boldsymbol{q})$, i = 1,2 of the inertia matrix $M(\boldsymbol{q})$ are as follows [21], [2]:

$$\begin{aligned} M_{11} &= \frac{1}{3}m_1 l_1^2 + \frac{1}{3}m_2 l_2^2 + m_2 l_1^2 + m_2 l_1 l_2 \cos q_2, \\ M_{12} &= M_{21} = \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2 \cos q_2, \\ M_{22} &= \frac{1}{3}m_2 l_2^2. \end{aligned} \tag{2}$$

In the case of the robot from Figure 2, $\boldsymbol{q}$, is the vector of angular displacements, $\boldsymbol{q} = [q_1 \quad q_2]^T$. The elements $C_{ij}(\boldsymbol{q},\dot{\boldsymbol{q}})$ (i,j = 1,2) are presented as:

$$C_{11} = -\frac{1}{2} m_2 l_1 l_2 \dot{q}_2 \sin q_2,$$
$$C_{12} = -\frac{1}{2} m_2 l_1 l_2 \sin q_2 \left(\dot{q}_1 + \dot{q}_2\right), \quad (3)$$
$$C_{21} = -\frac{1}{2} m_2 l_1 l_2 \dot{q}_1 \sin q_2, \; C_{22} = 0.$$

Finally, the elements of the gravitational torque vector $G(\boldsymbol{q})$ are given by:

$$G_1 = \left(\frac{1}{2} m_1 l_1 + m_2 l_1\right) g \cos q_1 +$$
$$+ \frac{1}{2} m_2 l_2 g \cos\left(q_1 + q_2\right), \quad (4)$$
$$G_2 = \frac{1}{2} m_2 l_2 g \cos\left(q_1 + q_2\right).$$

3. TRAJECTORY PLANNING

Smooth movement entails a constant change of position as well as constant changes in velocities and accelerations. This movement in living beings is perfectly natural, as it requires the least amount of energy. It is self-evident that abrupt or jumpy changes, whether in acceleration, speed, or position, require far more energy than gradual changes [22]. There are numerous approaches to the robot's trajectory planning from the initial q_i to the final q_f position. Linear interpolation, for example, gives continuous velocity movement, but it can also result in abrupt starting or stopping, which can lead to theoretically infinite accelerations. Furthermore, abrupt changes in velocity greatly increase the risk of high accelerations, which can be difficult to perform from the actuator's standpoint. The trapezoidal velocity profile is a realistically feasible implementation of motion at a constant speed [8] but, in this approach, the acceleration is a discontinuous function of time in the total duration of the movement T_f. Unwanted oscillations of the robot structure may result from this abrupt shift in acceleration. Therefore, it is advisable to choose a function that will allow continuous changes, not only in position and velocity but also in acceleration. As suitable candidates for interpolation functions, polynomials are imposed, because it is generally known that continuous polynomials are infinitely differentiable smooth functions with all of their derivatives. Following [22], position profile using the polynomial function can be expressed as:

$$q(t) = q_i + (q_f - q_i) p(\tau), \text{ for } p = \frac{q - q_i}{q_f - q_i}, \; \tau = \frac{t}{T_f}, \quad (5)$$

where p represents normalized position and τ is mapping the real time from the interval $t \in [0, T_f]$ to the normalized interval $t \in [0, 1]$. By differentiating expression (5) velocity and acceleration are determined as:

$$\dot{q}(t) = \frac{q_f - q_i}{T_f} p'(\tau) \text{ and } \ddot{q}(t) = \frac{q_f - q_i}{T_f^2} p''(\tau). \quad (6)$$

For the purposes of this paper, and in order to obtain a smooth motion without interruptions in acceleration, the polynomial of the fifth degree is proven to be enough, Figure 3. For the polynomial of the fifth degree, there are six boundary conditions:

$$p(0) = 0, \quad p'(0) = 0, \quad p''(0) = 0,$$
$$p(1) = 1, \quad p'(1) = 0, \quad p''(1) = 0. \quad (7)$$

Differentiating the expression for a fifth-order polynomial over normalized time yields the expression for normalized velocity and normalized acceleration. The system of equations is obtained by substituting the boundary conditions in these polynomials, which represents the solution that determines unknown coefficients. Final solution is:

$$p(\tau) = 6\tau^5 - 15\tau^4 + 10\tau^3,$$
$$p'(\tau) = 30\tau^4 - 60\tau^3 + 30\tau^2, \quad (8)$$
$$p''(\tau) = 120\tau^3 - 180\tau^2 + 60\tau.$$

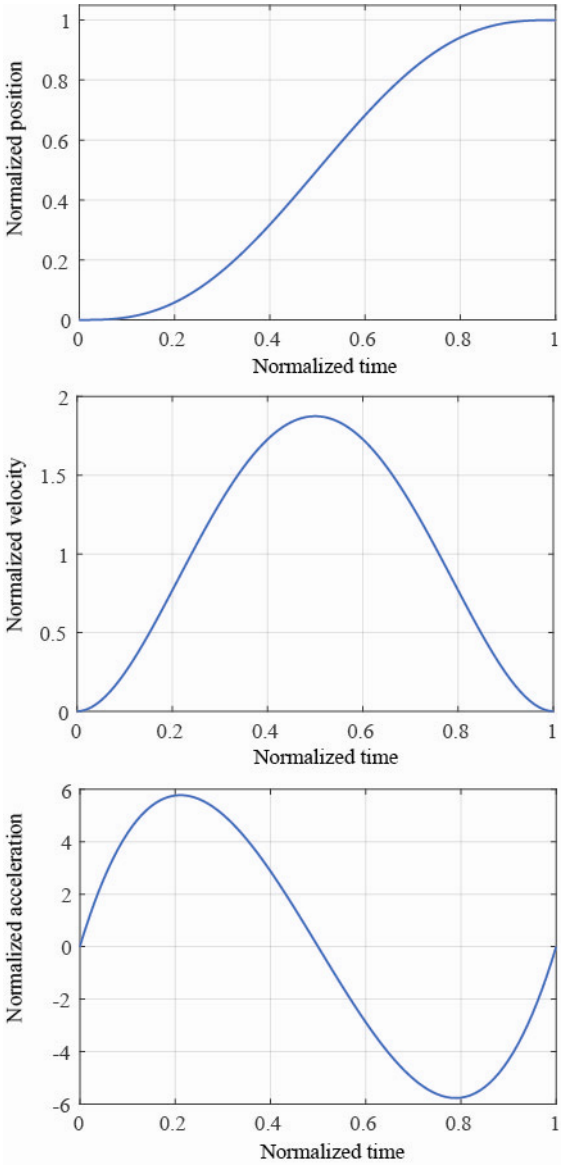

Figure 3. The polynomial velocity profile: the position, velocity and acceleration profiles

4. FUZZY LOGIC CONTROLLER

The fuzzy control approach will be used in the following paragraphs to develop a fuzzy controller that can move a two-link robot along a chosen trajectory. As

a result, two fuzzy proportional derivative (PD) type controllers, one for each individual link will be created. Defining the input and output variables, choosing the fuzzification and defuzzification procedure, and, most significantly, specifying the rule-base of the fuzzy controller are all crucial aspects of building a fuzzy controller. The error and error derivation of link position are the input variables of the FLC. The link control input, or torque, is the fuzzy controller's output variable. The common normalized interval [-1, 1] is used to define all membership functions for the controller inputs and outputs. Symmetric triangular functions with an equal base and 50% overlap with neighbouring membership functions for all of the membership functions (except the two at the ends, which are trapezoidal) are utilized, as illustrated in Figure 4 and Figure 5.

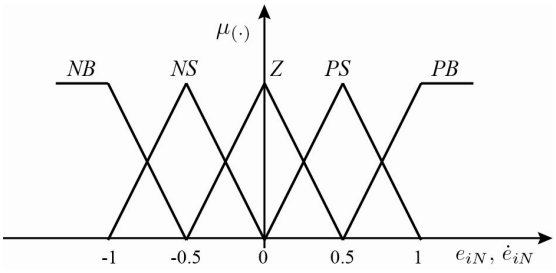

Figure 4. The input membership functions

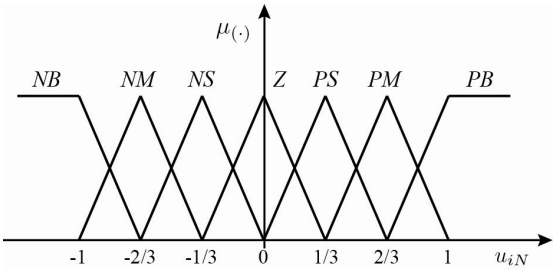

Figure 5. The output membership functions

Furthermore, both fuzzy controllers have the same membership function, where e_{iN} and $\dot{e}_{iN}$ for all i = 1,2… denote the normalized error and normalized derivative of the error, respectively. For link 1 and link 2, the normalized control signals are represented as u_{iN} for all i = 1,2, respectively. Each fuzzy set in a conventional fuzzy partition defines the linguistic variable's value. The fuzzy linguistic variables NB, NM, NS, Z, PS, PM and PB represent negative large, negative medium, negative small, zero, positive small, positive medium, and positive big values. As a result, Table 1 shows the fuzzy IF-THEN rules for robot trajectory control.

Table 1: Fuzzy IF-THEN rules for the robot trajectory control

e_N \ $\dot{e}$	**NB**	**NS**	**Z**	**PS**	**PB**
NB	NB	NB	NM	NS	Z
NS	NB	NM	NS	Z	PS
Z	NM	NS	Z	PS	PM
PS	NS	Z	PS	PM	PB
PB	Z	PS	PM	PB	PB

Using normalized domains necessitates a scale transformation, also known as input normalization, which converts the physical values of the input variables into a normalized domain. Figure 6 shows a Simulink model of a two-link robot system with fuzzy control in Matlab/Simulink. Furthermore, output denormalization converts the control output variable's normalized value into its physical domain. The following are the relationships between scaling factors and input and output variables, as stated above:

$$e_{iN} = S_{e_i} \cdot e_i,\ \dot{e}_{iN} = S_{de_i} \cdot \dot{e}_i,\ u_i = S_{u_i} \cdot u_{iN},\ \text{for } i = 1,2, \quad (9)$$

where e_i, $\dot{e}_i$ and u_i are error, the derivative error and control input, respectively. The suggested FLCs are implemented using the product inference engine and the center average defuzzification approach**.**

5. OPTIMIZATION OF FLC

Metaheuristic involves abstract algorithms for stochastic optimization that can be applied for solving multiple different constrained and unconstrained nonlinear sys–tems problems. In this research, AVOA is applied for optimizing fuzzy controller.

5.1 African vultures optimization algorithm

Vultures are not known as animals that attack other healthy animals, as they are feeding with corpses or diseased and wounded animals. Their specific behaviour served as an inspiration for creating new metaheuristic optimization algorithm in 2021 [23]. There are four phases in the algorithm that represent feeding process of African vultures.

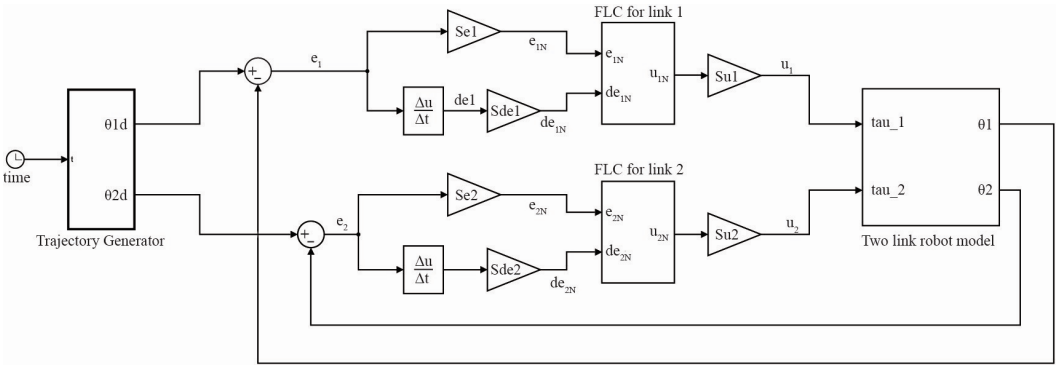

Figure 6. Simulink model of the 2-DOF gripping mechanism with fuzzy control

The first phase: Determining the best vulture

After the initialization of population, for every solution fitness is calculated by applying it to the function that is considered. The possible solutions are deployed into two different groups. The best solution of each group represents the best vulture into that group. This can be mathematically described by following equation:

$$R(i)=\begin{cases} BestVulture_1 \ if \ p_i = L_1 \\ BestVulture_2 \ if \ p_i = L_2 \end{cases}, \quad (10)$$

where *R(i)* represents the best vultures. Parameters L_1 and L_2 are defined before the optimization, have the values between 0 and 1, and

$$L_1 + L_2 = 1. \quad (11)$$

Index *i* represents *i*-th solution and Roulette wheel is used for choosing the best solution in each group:

$$p_i = \frac{F_i}{\sum_{j=1}^{n} F_i}, \quad (12)$$

where number of solutions is marked with *n*, and *F* is the satiety of the vultures. Calculations are repeated with each iteration for the whole population.

The second phase: Starvation rate of vultures

Satiated vultures have more energy which gives them ability to fly longer distances in finding food process. On the other hand, if vultures are starving, they do not have enough energy to fly, become aggressive and search for food next to more sinewy vulture. Starvation rate can be calculated as:

$$F = (2 \times rand_1 + 1) \times z \times \left(1 - \frac{iter_i}{maxiter}\right) + t, \quad (13)$$

where $rand_1$ is random number between 0 and 1, *z* has random value from -1 to 1, $iter_i$ represents current iteration, *maxiter* is total number of iterations and *t* can be calculated as:

$$t = h \times \left(\sin^{\omega}\left(\frac{\pi}{2} \times \frac{iter_i}{maxiter}\right) + \cos\left(\frac{\pi}{2} \times \frac{iter_i}{maxiter}\right) - 1\right) \quad (14)$$

In (14) *h* has random value between -2 and 2, and ω has constant value that is determined before optimization process. If absolute value of *F* is greater than 1, then AVOA starts exploration phase because vultures will look for food in different territory, if that is not the case AVOA starts exploitation phase as vultures look for food near the other vultures.

The third phase: Exploration phase

This phase encompasses two strategies, which can be described by (15) and (17).

$$P(i+1) = R(i) - D(i) \times F. \quad (15)$$

D(i) can be calculated via following expression:

$$D(i) = \left| X \times R(i) - P(i) \right|. \quad (16)$$

The second strategy can be mathematically described as:

$$P(i+1) = R(i) - F + rand_2 \times \left((ub - lb) \times rand_3 + lb \right). \quad (17)$$

Vulture position vector in the next iteration is marked with *P(i+1)*, *P(i)* is position of vulture in current position, *X* is position where vultures move randomly to save food from others and it is defined as:

$$X = 2 \times rand. \quad (18)$$

In (17) $rand_2$ has random values between 0 and 1, and $rand_3$ has value near 1 which increases the randomness coefficient. Symbols *ub* and *lb* are upper and lower limit of the variables, respectively. Which strategy will be implemented is determined with following expression:

$$P(i+1) = \begin{cases} (15) \ if \ P_1 \geq rand_{p1} \\ (17) \ if \ P_1 < rand_{p1} \end{cases}. \quad (19)$$

P_1 is parameter settled before optimization and has value between 0 and 1, $rand_{p1}$ is random value from the same range.

The fourth phase: Exploitation phase

This phase has two steps. Each of them has two different strategies. Parameters P_2 and P_3 determine which strategy will be chosen in first and second step, respectively. Values for those parameters, which can be between 0 and 1, are settled before search operation.

First step. The AVOA enters this step when absolute value of *F* is between 1 and 0.5. It means that stronger vultures have enough energy and do not want to share food, but weaker vultures gather around them and cause small conflict. Firstly, random value between 0 and 1, $rand_{p2}$, is generated. Strategy selection in this step is determined by following equation:

$$P(i+1) = \begin{cases} (21) \ if \ P_2 \geq rand_{p2} \\ (25) \ if \ P_2 < rand_{p2} \end{cases}. \quad (20)$$

First strategy can be described as:

$$P(i+1) = D(i) \times (F + rand_4) - d(t), \quad (21)$$

$$d(t) = R(i) - P(i). \quad (22)$$

Random value, $rand_4$, is between 0 and 1 and it is used for increasing the random coefficient, while *d(t)* is the distance of the vulture from one of two strongest vultures. The second strategy is based on rotational flight of vultures. To represent this flight mathematically, spiral equation is created between one of the two strongest vultures and all other vultures.

$$S_1 = R(i) \times \left(\frac{rand_5 \times P(i)}{2\pi} \right) \times \cos\left(P(i)\right), \quad (23)$$

$$S_2 = R(i) \times \left(\frac{rand_6 \times P(i)}{2\pi} \right) \times \cos\left(P(i)\right). \quad (24)$$

Random values $rand_5$ and $rand_6$ are between 0 and 1. The second strategy can be determined as:

$$P(i+1) = R(i) - (S_1 + S_2). \quad (25)$$

Second step. When absolute value of F is less than 0.5 then algorithm enters this step. At the beginning random value between 0 and 1, $rand_{p3}$, is generated. Formula for choosing strategy in this step is given below.

$$P(i+1) = \begin{cases} (28)\ if\ P_3 \geq rand_{p3} \\ (29)\ if\ P_3 < rand_{p3} \end{cases}. \quad (26)$$

The first strategy is inspired by grouping of several species of vultures around same source of food.

$$A_1 = BestV_1(i) - \frac{BestV_1(i) \times P(i)}{BestV_1(i) - P(i)^2} \times F, \quad (26)$$

$$A_2 = BestV_2(i) - \frac{BestV_2(i) \times P(i)}{BestV_2(i) - P(i)^2} \times F. \quad (27)$$

In (26) and (27) $BestV_1$ and $BestV_2$ are the best vultures of the first and the second group in current iteration, respectively. Formula for the first strategy is:

$$P(i+1) = \frac{A_1 + A_2}{2}. \quad (28)$$

The second strategy represents the case when the best vultures are also weak to deal with the rest of vultures that are aggressive in order to gain food, so they move straight in direction of the best vulture. In this case, strategy equation is

$$P(i+1) = R(i) - |d(t)| \times F \times Levy(x). \quad (29)$$

Levy(d) is basically levy function (*LF*) that is used for increasing effectiveness of this optimization algorithm. It is calculated as:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}. \quad (30)$$

First strategy can be described as:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta 2}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta}. \quad (31)$$

In (30) u and v have random values from 0 to 1, and β is fixed value, by default it is equal to 1.5. All used parameters in AVOA are settled to default values.

5.2 Metaheuristic optimization algorithms for comparison

Results obtained by using previously explained algo–rithm are compared to results obtained by implementing three different optimization algorithms such as the whale optimization algorithm (WOA), the ant lion optimizer (ALO) and a sine cosine algorithm (SCA). All of the parameters have default values. The WOA was introduced in [18] and this metaheuristic algorithm is based on the hunting method of humpback whales that implies swimming up to the surface from depth, while creating bubbles in a spiral shape around the prey. The ALO is based on hunting behaviour of ant lions. They use pits as traps to capture ants.

Once the ants walk in trap, ant lions shoot sand from the centre of the pit in order to slide down captured ants. Algorithm was pre–sented in [24]. Another population-based method for optimization is SCA, proposed in [25].

This algorithm in exploration phase creates many random solutions from the set of solutions. Then in exploitation phase, position is updated by using sine and cosine trigono–metric functions in order to find optimal solution.

5.3 Optimization of FLC using the African vultures optimization algorithm

Fuzzy controllers, in general, contain a vast number of parameters that may be tweaked in order to achieve the best dynamical response. The form of the membership functions, the number of linguistic variables for input and output values of the set of rules, scaling factors, and other parameters are among them.

Furthermore, it is clear from using the predetermined membership functions defined in Section 4, as well as the set of rules (Table 1), that the performance of the fuzzy PD controller is dependent on the input and output scaling factors, and that the design of the fuzzy controller can be attributed to the choice of those input/output scaling factors. The adjustment only of the scaling factors, considering their correspondence to the gains of the controller, has been the exclusive emphasis of this research. In addition, the AVOA optimization technique was employed to create the best fuzzy PD controller.

Furthermore, the aforementioned parameters are all coded into a single agent, who is given a vector containing, in our example, six parameters. We used the algebraic sum of the ITAE (integral of time-weighted absolute error) performance criteria of both links as the objective function:

$$J = \int_0^\infty t \cdot \left[|e_1(t)| + |e_2(t)| \right] dt. \quad (32)$$

6. EXPERIMENTAL RESULTS

Finally, following simulation is performed on 2-DOF robot, shown in Figure 2, in order to implement opti–mized fuzzy controller with the proposed metaheuristic algorithm. The physical parameters for the gripping mechanism are m_1 = 0.00799 kg, m_2 = 0.00521kg, l_1 = 0.05831m and l_2 = 0.0422 m.

The initial position of the gripping mechanism is determined by the mechanism itself. In our case, the initial link configuration is defined as q_0 = [1.3963 - 0.5236]^T rad and lastly, the initial end-effector position is x_0 = 0.0373 m, y_0 = 0.0898 m.

The control task is to move that point from its initials to the finals coordinates defined by angles q_f = [0.7854 -0.7854]^T rad, and the end-effector position x_f = 0.0834 m, y_f = 0.0412 m.

The time required to reach this position is set to be T_f = 6 s. For determined initial and final movement points and specified arrival time at the final point, desired trajectory equations for each of the segment are:

$$q_1(t) = q_{01} + (q_{f1} - q_{01}) \cdot p\,, \tag{33}$$

$$q_1(t) = 1.3965 + (0.7854 - 1.3965) \cdot p\,, \tag{34}$$

$$q_2(t) = q_{02} + (q_{f2} - q_{02}) \cdot p\,, \tag{35}$$

$$q_2(t) = -0.5236 + (-0.7854 + 0.5236) \cdot p\,. \tag{36}$$

The desired end-effector trajectory of the 2-DOF manipulator is specified according to polynomial velocity profile defined in Section 3.

Efficiency of proposed optimized fuzzy controller is tested for the following desired trajectories of the mechanism:

1) First of all, desired trajectory is defined via (33) and (34), which corresponds the case that mechanism is placed in defined initial point.

2) In the second case, it is assumed that gripping mechanism is not placed at initial point, but that there is certain deviation that equals 5 degrees for each segment (link). In practise, for example, this corresponds the case that mobile robot did not take completely correct position in its movement and positioning in front of the production machine. Task is still that mechanism continues the movement along to the given trajectory (33) and (34), therefore in the presence of the initial deviation.

3) Finally, the robustness of the proposed control algorithm is tested for the case when the masses of the segments increase three times.

For all of the previously mentioned algorithms the population is set to 30, while the total number of iterations is set to 50. Number of agents represents the number of potential optimal fuzzy controllers. As it was already mentioned, all of the parameter values that are used in the implementation of the optimization algo–rithms are taken from the original papers [18, 21, 22, 23].

Table 2 summarizes the results obtained by apply–ing the different optimization approaches to the pro–posed fuzzy control. The results are averaged over ten independent runs, and the best results are indicated in the bold type.

Table 2. The average and the best objective function values

Optimization algorithm	Objective function – average value	Objective function – the best value
ALO	0.1539	0.0769
SCA	0.0712	0.0597
WOA	0.1494	0.0623
AVOA	**0.0636**	**0.0558**

The convergence curve of the objective function value is depicted in Figure 7**.** From Table 2 and Figure 7, it is possible to conclude that, according to minimal value of the objective function, SCA and AVOA gave better results compared to WOA and ALO, but the best result is obtained by AVOA. Likewise, it can be noticed that the convergence rates of SCA and AVOA are very similar. In addition, after the optimization with AVOA the obtained parameters for the scaling factors are:

$$S_{e_1} = 1.999\,,\quad S_{de_1} = 0.0106\,,\quad S_{u_1} = 1.5\,,$$
$$S_{e_2} = 1.998\,,\quad S_{de_2} = 0.0104\,,\quad S_{u_2} = 0.6834\,.$$

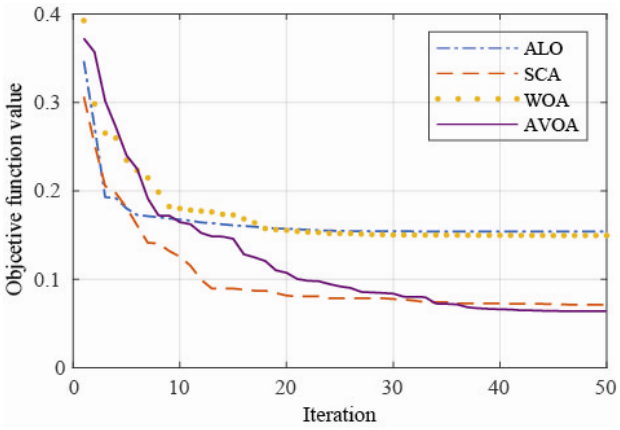

Figure 7. The convergence curve of the objective function value

In the following two pictures, the comparison between the real trajectory and the desired trajectory of the link 1 (Figure 8) and link 2 (Figure9) are shown.

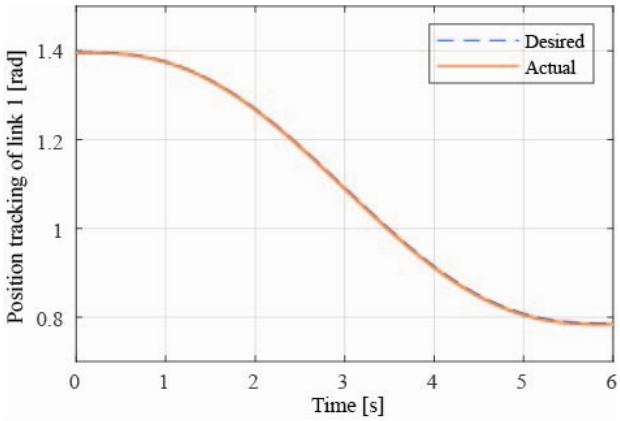

Figure 8. A comparison between the desired and real trajectory of link 1

There we can also observe that the real and desired trajectory curves both almost match, with very slight deviations, nearly neglectable. Moreover, the errors of position tracking for link 1 and link 2, are given in Figure 10 and Figure 11, respectively.

The error for the position tracking of the first link is less than 0.0025 rad, while for the second link it is less than 0.0015 rad.

Finally, in Figure 12 and Figure 13 we have depicted the control torque of both link 1 and link 2. Results for the second case that implies deviation from initial point of gripping device of the mechanism are given in Figure 14 and Figure 15. It can be seen that it takes 0.03 seconds for trajectories to harmonize.

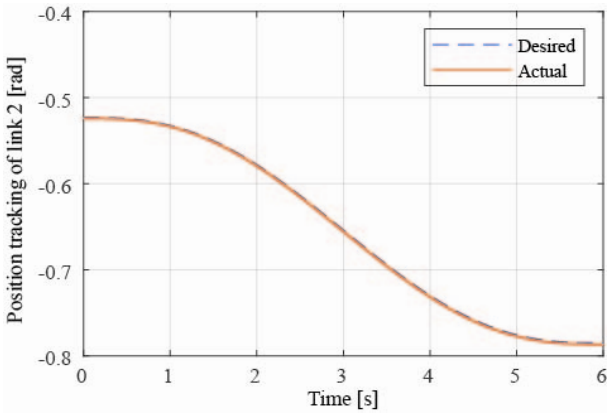

Figure 9. A comparison between desired and real trajectory of link 2

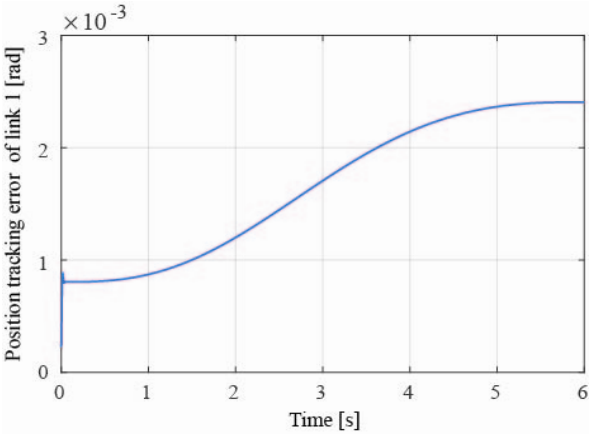

Figure 10. Position tracking error of link 1

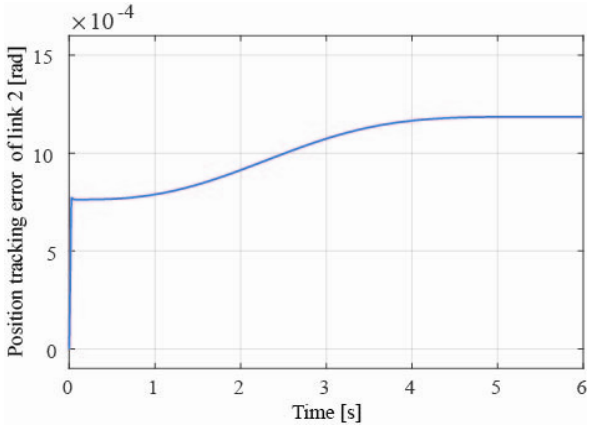

Figure 11. Position tracking error of link 2

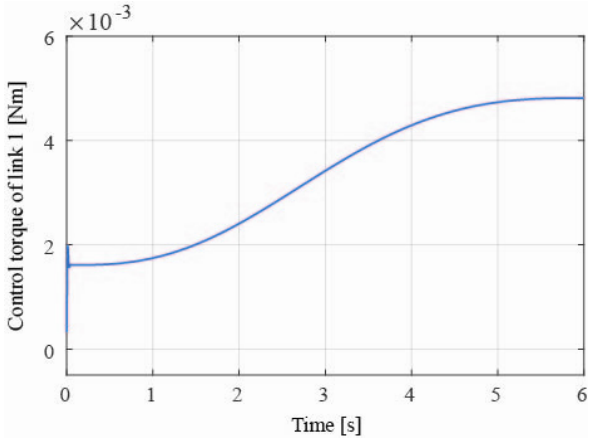

Figure 12. Control torque of link 1

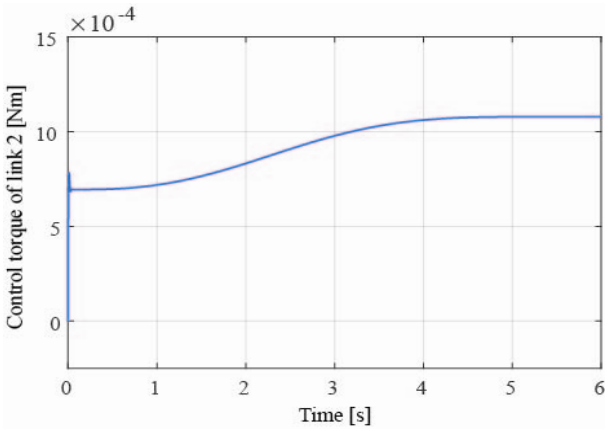

Figure 13. Control torque of link 2

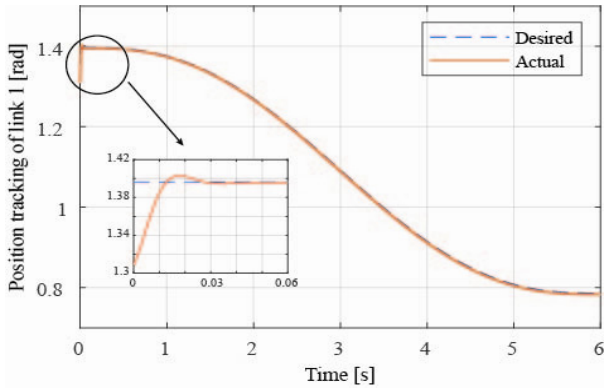

Figure 14. A comparison between the desired and real trajectory of link 1 (nonzero initial conditions)

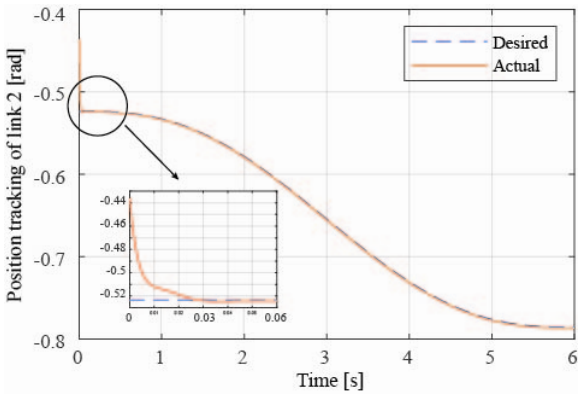

Figure 15. A comparison between the desired and real trajectory of link 2 (nonzero initial conditions)

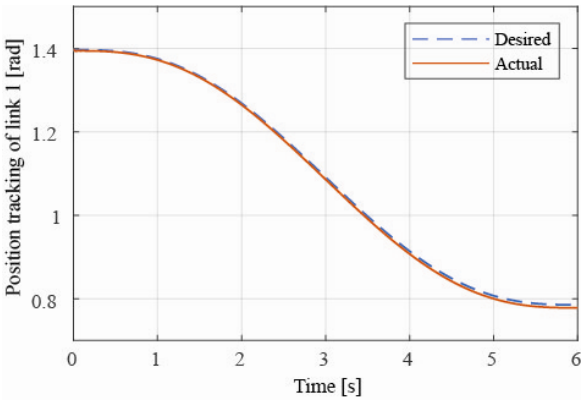

Figure 16. A comparison between the desired and real trajectory of link 1 (increased mass of links three times)

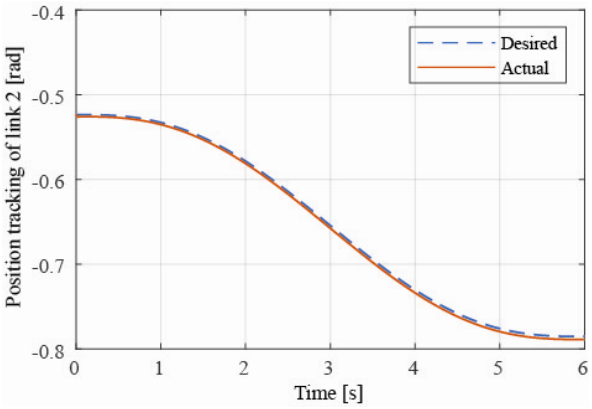

Figure 17. A comparison between the desired and real trajectory of link 2 (increased mass of links three times)

Furthermore, the robustness of designed fuzzy controllers is tested, for three times increased mass of each link. The parameters of the fuzzy controller that is optimized using the AVOA are unchanged, and comparisons of real and desired trajectories are shown in Figure 16 and Figure 17.

In the pictures above we can clearly notice that even though we enlarged the mass of link 1 and link 2, the optimized fuzzy algorithm works exceptionally well.

In addition, the errors of position tracking for link 1 and link 2, where the given links have a mass that has been increased three times in order to test the robus–tness, are given in Figure 18 and Figure 19, respec–tively.

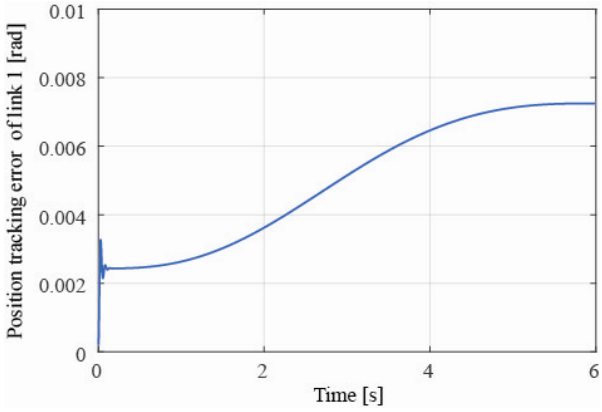

Figure 18. Position tracking error of link 1 (increase mass of links three times)

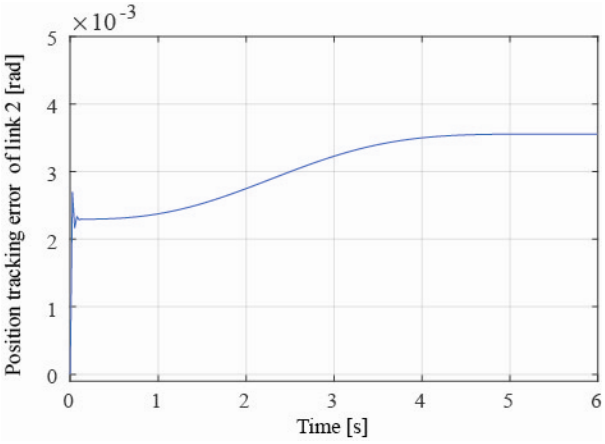

Figure 19. Position tracking error of link 2 (increase mass of links three times)

Here the error for the position tracking of the first link is about 0.008 rad, while for the second link it is about 0.004 rad.

7. CONCLUSION

This paper deals with the problem of the trajectory tracking control of a two-link gripping mechanism as a part of mobile robot working cycle. As potential solution, fuzzy PD controller with optimized parameters is proposed. AVOA is metaheuristic algorithm that is used for optimizing the scaling factors of fuzzy PD controller. In order to examine its performance, three other algorithms – WOA, ALO and SCA are used for comparison. It was shown that AVOA achieved the lowest objective function value, which makes it superior to the others. For analysing the trajectory tracking performance of the designed controller numerical simulations are performed. In order to test robustness of the proposed controller, a simulation is done in case of increasing the mass of robot segments three times. The simulation results showed that proposed controller has the ability to deal with the nonlinearities of the robot and the changing of its parameters. In all of the considered cases desired and actual trajectory are very close, so the position tracking errors have low values. Based on the presented results it can be concluded that proposed optimized fuzzy PD controller represents a good possible solution for solving stated problem.

ACKNOWLEDGMENT

The research of the authors was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI- MISSION4.0, 2020–2022.

Their work was as well financially supported by the Ministry of Education, Science and Technological Development of the Serbian Government, MPNTR RS under contract 451-03-68/2022-14/200105 from date 04. 02. 2022, TR 35004.

REFERENCES

[1] Bugarić U., Popović D. and Tošić S.: Methodology for analysis of working cycle, in *XVII International Conference on Material Flow, Machines and Devices in Industry*, 2002, Belgrade, pp. 4–5.

[2] Jovanović, R, Bugarić, U., Laban, L. and Vesović M.: Trajectory Tracking of a Two – Link Gripping Mechanism, in *X International Scientific Conference Heavy Machinery,* 2021, Kraljevo, Session C 1-9.

[3] Tai, K., El-Sayed, A. R., Shahriari, M., Biglarbegian M. and Mahmud S.: State of the art robotic grippers and applications. Robotics, Vol. 5 No. 2, pp. 11, 2016.

[4] Ishak, A. J., Soh A. C. and Ashaari M. A.: Position control of arm mechanism using PID controller. Journal of Theoretical & Applied Information Technology, Vol. 47, No. 2, pp. 798–806, 2013.

[5] Wu, Z. and Li, Y.: Design and control of a novel micro-gripper using adaptive backstepping slide

mode control method. Microsystem Technologies, Vol. 27 No. 12, pp. 4227-4239, 2021.

[6] Widhiada, W., Nindhia, T.G.T. and Budiarsa, N.: Robust control for the motion five fingered robot gripper. International Journal of Mechanical Engineering and Robotics Research, Vol. 4 No. 3, pp. 226, 2015.

[7] Borisov, I.I., Borisov, O.I., Gromov, V.S., Vlasov, S.M., Dobriborsci, D. and Kolyubin, S.A.: Design of versatile gripper with robust control. IFAC-PapersOnLine, Vol. 51 No. 22, pp.56-61, 2018.

[8] Vesović M., Jovanović, R, Laban, L. and Bugarić, U.: Feedback Linearization Control of a Two – Link Gripping Mechanism, in *X International Scientific Conference Heavy Machinery,* 2021, Kraljevo, Session C 9-17.

[9] Jiang B., Karimi H. R., Yang S., Gao C. and Kao Y.: Observer-Based Adaptive Sliding Mode Control for Nonlinear Stochastic Markov Jump Systems via T-S Fuzzy Modeling: Applications to Robot Arm Model. IEEE Transactions on Industrial Electronics, Vol. 68 No. 1, pp. 466–477, 2021.

[10] Xu Q., Kan J., Chen S. and Yan S.: Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink. International Journal of Control and Automation, Vol. 7 No. 8, pp. 233–244, 2014.

[11] Rehman S., Khan S. and Alhemis L.: The effect of acceleration coefficients in particle swarm optimization algorithm with application to wind farm layout design. FME Transactions,Vol. 48 No. 4, pp. 922–930, 2020.

[12] Milenković B., Jovanović Đ. and Krstić, M.: An Application Of Dingo Optimization Algorithm (DOA) For Solving Continuous Engineering Problems. FME Transactions, Vol. 50 No. 2, pp. 331–338, 2022.

[13] Ahmed, M., Khamies, M., Kamel, S. and Magdy, G.: Designing Optimal $PD^{\mu}N - I^{\lambda}T$ Controller for LFC of Two-area Power Systems Using African Vulture's Optimization Algorithm, in *22nd International Middle East Power Systems Conference (MEPCON)*, 2021, Egypt, pp. 430-437.

[14] Guo, J., Yan, D., Cao, H. and Jiang, Z.: The Point to Point Trajectory Planning Based on the Ant Lion Optimizer, International Journal of Automation and Control, Vol. 10, No. 2, pp. 155-166, 2016.

[15] Bingul Z. and Karahan O.: A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control. Expert Systems with Applications Vol. 38, pp. 1017–1031, 2011.

[16] Zhao J., Han L., Wang L. and Yu Z.: The fuzzy PID control optimized by genetic algorithm for trajectory tracking of robot arm, in *12th World Congress on Intelligent Control and Automation*, 12 June-15 June 2016, Guilin (China), pp. 556–559.

[17] Kumar, S., Parhi, D.R., Muni, M.K. and Pandey, K.K: Optimal path search and control of mobile robot using hybridized sine-cosine algorithm and ant colony optimization technique. Industrial Robot, Vol. 47 No. 4, pp. 535-545, 2020.

[18] Mirjalili, S. and Lewis, A.: The Whale Optimization Algorithm, Advances in Engineering Software, Vol. 83, pp. 51-67, 2016.

[19] Cong, V.D., Hanh, L.D., Phuong, L.H. and Duy, D.A.: Design and development of robot arm system for classification and sorting using machine vision. FME Transactions, Vol. 50 No. 1, pp.181-181, 2022.

[20] Cong, V.D.: Industrial robot arm controller based on programmable System-on-Chip device. FME Transactions, Vol. 49 No. 4, pp.1025-1034, 2021.

[21] L. Sciavicco and B. Siciliano, "Modeling and control of robot manipulators," Springer-Verlag, London (England), 2000.

[22] https://www.ucg.ac.me/skladiste/blog_13269/objava_8796/fajlovi/35_industtrijska781.pdf, University of Montenegro, pp. 87-102, (last accessed 12/05/22)

[23] Abdollahzadeh, B., Gharehchopogh F.S. and Mirjalili, S.: African Vultures Optimization Algorithm: A New Nature-inspired Metaheuristic Algorithm for Global Optimization Problems, Computers & Industrial Engineering, Vol. 158, 2021. 107408

[24] Mirjalili, S.: The Ant Lion Optimizer, Advances in Engineering Software, Vol. 95, pp. 80-98, 2015.

[25] Mirjalili, S.: A Sine Cosine Algorithm for Solving Optimization Problems, Knowledge-Based Systems, Vol. 96, pp. 120-133, 2016.

ФАЗИ УПРАВЉАЊЕ ОПТИМИЗОВАНО АЛГОРИТМОМ АФРИЧКИХ СУПОВА ЗА ПРАЋЕЊЕ ПУТАЊЕ ЗАХВАТНОГ МЕХАНИЗМА СА ДВА СЕГМЕНТА

Р. Јовановић, У. Бугарић, М. Весовић, Н. Перишић

У овом раду је приказан пропорционално – диференцијални фази управљачки систем за праћење путање захватног механизма са два степена слободе. У циљу постизања кретања захватног механизма без наглог заустављања, коришћен је полиномијални профил брзине. Алгоритам оптимизације афричких супова, као један од најновијих метахеуристичких алгоритама, употребљен је за одређивање оптималних улазно/излазних фактора скалирања предложеног фази управљачког система, а према изабраној функцији циља. Резултати добијени овим алгоритмом су упоређени са три нова и популарна оптимизациона алгоритма, која су инспирисана: кретањем китова, кретањем мрављих лавова и математичким функцијама синус и косинус. Такође, приказани су резултати симулација рада система које одговарају случају да се захватни механизам налази у дефинисаном почетном положају, као и када се претпостави да постоји извесно одступање

положаја сваког сегмента, јер захватни механизам није у свом дефинисаном почетном положају. Коначно, робусност предложеног алгоритма управљања тестирана је за случај када се масе сегмената захватног механизма повећају три пута. Резултати су показали да је предложени управљачки систем способан да се носи са нелинеарностима захватног механизма, променама почетних позиција и параметара. Кретање захватног механизма је глатко и прати унапред задату путању.