IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 June 2022, accepted 12 July 2022, date of publication 1 August 2022, date of current version 23 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3195928

RESEARCH ARTICLE

Energy-Efficient Gait Optimization of Snake-Like Modular Robots by Using Multiobjective Reinforcement Learning and a Fuzzy Inference System

AKASH SINGH[1], WEI-YU CHIU[1], (Senior Member, IEEE), SHRI HARISH MANOHARAN[1], AND ALEXEY M. ROMANOV[2], (Senior Member, IEEE)

[1]Multi-Objective Control and Reinforcement Learning (MOCaRL) Laboratory, Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300044, Taiwan

[2]Institute of Artificial Intelligence, MIREA—Russian Technological University, 119454 Moscow, Russia

Corresponding author: Wei-Yu Chiu (chiuweiyu@gmail.com)

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 110-2221-E-007-097-MY2.

ABSTRACT Snake-like modular robots (MRs) are highly flexible, but, to traverse a challenging terrain or explore a region of interest, MR needs to attain efficient locomotion depending on a tradeoff between objectives like forward velocity and power consumption of the robot. The objectives can vary with different weights depending upon the situation, reflecting relative objective importance. This study developed a multiobjective reinforcement learning algorithm based on a fuzzy inference system (FI-MORL) to select the most appropriate gait parameters of snake-like MRs according to the objective weights. The developed algorithm employs a fuzzy inference system to reduce the number of states in an environment, which results in faster learning. The proposed approach uses the previously learned experience to rapidly achieve the best objective values in response to a change in weights. While setting equal importance to the objectives, FI-MORL delivers superior performance than single-objective reinforcement learning algorithms by consuming 2% less power and gaining 2.5% higher velocity since it mitigates the effect of weight change, similar performance found comparing an actor-critic algorithm. Likewise, the proposed method outperforms by consuming 14% less power and achieving 11% higher velocity than traditional methods like proximal policy optimization, deep Q-network, and vanilla policy gradient. Even after weight change, FI-MORL achieved a 14% higher reward than the above methods. The proposed FI-MORL framework can effectively converge quicker and efficiently handle the changes in objective weights.

INDEX TERMS Energy efficiency, fuzzy inference system, gait optimization, modular robot, multiobjective reinforcement learning, snake-like modular robot.

I. INTRODUCTION

A modular robot (MR) consists of individual modules that facilitate it to perform various tasks in an environment by autonomously changing its shape and behavior. A module generally includes actuators to assist in shape-changing, and locomotion [1]. New modules can be autonomously attached to or detached from MRs [2] to form a distinctive morphology that makes MRs self-reconfigurable [3].

Among the various MRs, snake-like MRs have gathered attention due to their flexibility and applicability in situations where the human presence can be critical, such as military stealth operations, space exploration, and underwater inspection [2]. Often, these robots encounter situations

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

where an MR may need to change gaits for locomotion at different stages. For example, during the teleoperation, an MR equipped with a camera can move at a fast gait speed for locomotion to reach the destination on a known terrain while using a slower gait speed in an unknown environment [4], [5]. In the oil and gas industries, snake robots might inspect pipes underwater for a longer duration, requiring energy-efficient gait control; during an oil leak, the robot needs to move fast to support the repair [6]. These scenarios require MRs to move fast in certain situations and stay energy-efficient in others.

Although snake-like MRs are helpful, a tradeoff exists between the robot speed and power consumption, primarily due to the limited energy-carrying capabilities of the MR. Thus it is essential to reduce power consumption whenever applicable and increase the robot speed. However, achieving such locomotion for snake-like MRs is not accessible due to the complexity of controlling multiple degrees of freedom (DoFs). Several snake-like MR control mechanisms have been developed to achieve autonomous locomotion [7].

In 1994, Hirose [8] developed serpentine equations to produce the sinusoidal type of locomotion mimicking the gait of a snake. Serpentine has been the most straightforward and energy-efficient locomotion compared to others [9]. Ma [10] proposed a serpentine curve to model locomotion for snake-like MRs and achieved high locomotion efficiency. Dehghani and Mahjoob [11] developed a modified serpentine equation to reduce slipping while varying the parameters. Since then, several studies on MRs used the serpentine equation to attain a faster speed for locomotion [12]–[15].

Using the serpentine equation, researchers used multiple techniques to get locomotion with optimized power consumption. Bing *et al.*. [16] used Reinforcement Learning (RL) and proximal policy optimization algorithm to develop energy-efficient locomotion for different velocities. Kelasidi *et al.* [17] used a weighted-sum approach by combining objectives such as power consumption and velocity. They varied the weights heuristically and used particle swarm optimization to obtain Pareto optimal solutions. Some studies have also optimized the locomotion to get desired forward velocity. Christensen *et al.* [18] proposed distributed reinforcement learning and independent morphological methods to optimize the gait control table of different MRs to achieve high velocity. Spröwitz *et al.* [19] investigated the locomotion of a snake-like MR with a gradient-free optimization algorithm that learns central pattern generator parameters to control each module. Crespi and Ijspeert [20] employed a biological central pattern generator by using a heuristic optimization algorithm to adjust the travel speed of a snake-like MR. Chee *et al.* [21] proposed a multiobjective hybrid genetic algorithm and self-adaptive differential evolution approach to optimize the parameters of a gait control equation and co-evolve the morphology as well as controller of a snake-like MR. Wu and Ma [22] proposed a central pattern generator based approach to control the gait of a snake-like MR. Cao *et al.* [23] implemented multiple locomotions on different parts of the snake robot's body to analyze the robot performance concerning speed and energy on sloped terrain. Many works have focused on various locomotions.

Objectives like lower power consumption or increased velocity for locomotion have been optimized in the literature, but these objectives were generally weighted using fix weighting coefficients. When a new terrain or a new scenario is encountered, different weights are given to reflect the priority of one objective over the other. Using existing approaches, the optimization or learning process must restart in response to the different weighted objectives, incurring a high time cost.

In this study, we solved the two contradictory objectives: minimize the power consumption and maximize the average velocity. In contrast with conventional single-objective reinforcement learning, we proposed a solution based on multiobjective reinforcement learning and fuzzy inference (FI-MORL). By solving the problem, a snake-like MR can quickly switch between an energy-efficient mode and a fast velocity mode for locomotion without incurring an extra time cost.

In the proposed FI-MORL algorithm, observation states are constructed based on the power consumed by each module. In this manner, the proposed method learns energy-efficient and faster gaits for a snake-like MR. The fuzzy inference system helps discretize continuous observation states and reduce sensitivity to the environment, thus achieving faster learning. The proposed method separately updates two Q-tables: one relates to power consumption and forward velocity. An action is then taken on the basis of a weighted-sum of the the two Q-tables. This practice allows the MR to avoid relearning when the weighted coefficient changes. We compared the proposed algorithm with the fuzzy inference single-objective reinforcement learning (FI-SORL) algorithm and benchmark deep single-objective reinforcement learning algorithms. After a change of the weighting coefficient, the proposed method reached steady-state objective values faster than the SORL algorithms.

The main contributions of this paper are as follows.

- The proposed FI-MORL method can systematically balance two conflicting objectives: speed and power consumption.
- The fuzzy inference system is developed to reduce the number of possible states, thereby reducing the computation burden and achieving quicker convergence.
- The proposed method can expeditiously achieve the best locomotion after changing weights by addressing both objectives.

The rest of this paper is organized as follows. The system model of a snake-like MR is briefly described in Section II. Section III describes the control mechanism of a snake-like MR, the performance metric used in this study, and the problem formulation. The proposed FI-MORL algorithm, in which states are designed using a fuzzy inference system, actions, and rewards, is explained in Section IV. The

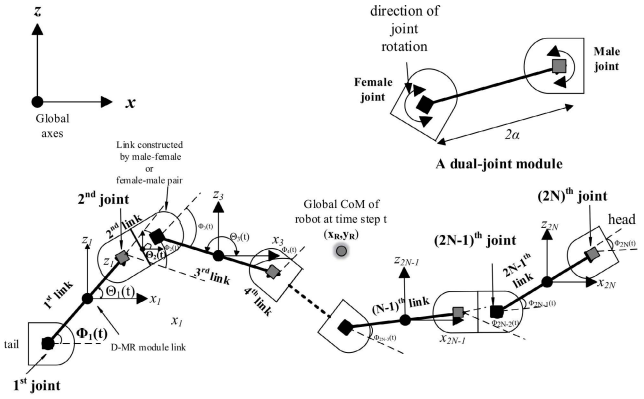

FIGURE 1. Parameters of 2D snake-like MR kinematics.

TABLE 1. Parameters of the snake-like MR.

Symbols	Description	Range
N	Number of modules in snake-like MR	
α	Half length of a snake-like MR module link	
m	Mass of one module	
COM	Center of Mass (CoM) of a snake-like MR	
(x, z)	Global coordinate axes	$\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}$
(x_i, z_i)	Global CoM coordinates of ith link of a snake-like MR	$\boldsymbol{x_i}, \boldsymbol{z_i} \in \mathbb{R}^{2N}$
$(x_{\mathrm{R}}, z_{\mathrm{R}})$	Global CoM coordinates of a snake-like MR	$\boldsymbol{x}_{\mathrm{R}}, \boldsymbol{z}_{\mathrm{R}} \in \mathbb{R}^2$
$\Theta_i(t)$	Angle between ith link and global $\boldsymbol{x}$ axis at time step t	$\boldsymbol{\Theta_i(t)} \in \mathbb{R}^{2N}$
$\Phi_i(t)$	ith joint angle at time step t	$\boldsymbol{\Phi_i(t)} \in \mathbb{R}^{2N}$
$\overline{\Theta}(t)$	Orientation of snake-like MR at time step t	
$R_i^{\mathrm{GF}}(t)$	Local rotation matrix of link i at time step t	

simulation results obtained for the FI-SORL algorithm, existing deep SORL algorithms, and the proposed algorithm are presented in Section V. Finally, the conclusions are reported in Section VI.

II. SYSTEM MODEL

This section presents the mathematical modeling of a snake-like MR with N dual-joint Dtto Explorer MR (D-MR) [24] modules, inspired by the M-TRAN module moving on the horizontal surface.

A snake-like MR consists of multiple connected modules, and each module has one or more actuators or joints. Servo motors act as actuators for modules and have sufficient torque to actuate the joints with reasonable force. With appropriate actuator commands, the entire mechanism can mimic snake-like movement. Fig. 1 represents the kinematic parameters and coordinates of a snake-like MR and the symbols defined in Table 1. In this study, N identical D-MR modules connected along the same axis to build a snake-like MR. Each module contains its Center of Mass (CoM) in its center position with a length of 2α and the mass m. Every module has two motorized joints (a "male" and a "female" joint), which are collectively responsible for providing motion to the snake-like MR at each time step $t \in \{1, 2, 3, \ldots, T\}$, where T represents the last time step of snake-like MR motion.

For the mathematical representation of snake-like MR's kinematic model, we adopt the Denavit Hartenberg convention [25], a systematic method for defining the kinematics model of any serially connected mechanism [26]. The male/female module along with the connection between male-female / female-male module represents a link; see Fig. 1. The angle of ith link, where $i \in \{1, 2, 3, \ldots, 2N\}$ is denoted by $\Theta_i(t)$, representing the angle between the link and the x axis of global coordinate in the counterclockwise direction. The links connected to the joints can actuate in both clockwise and counterclockwise directions. The angle of the ith joint is denoted by $\Phi_i(t)$ and obtains as follows:

$$\Phi_i(t) = \Theta_i(t) - \Theta_{i+1}(t) \qquad \forall i \in \{1, 2, 3, \ldots, 2N\}. \quad (1)$$

The orientation or heading of the snake-like MR is denoted by $\overline{\Theta}(t)$, which can be described using the average link angle as follows [27]:

$$\overline{\Theta}(t) = \frac{1}{N} \sum_{i=1}^{N} \Theta_i(t). \quad (2)$$

A snake-like MR moves in the global coordinate system. The local coordinate system of each link starts at its CoM. When the link angle $\Phi_i(t)$ is 0°, the local coordinate axes of the ith link, x_i and z_i aligns with the global x-axis and z-axis, respectively. The rotation matrix of the ith link with the global coordinate system is as follows:

$$R_i^{\mathrm{GF}}(t) = \begin{bmatrix} \cos\Theta_i(t) & -\sin\Theta_i(t) \\ \sin\Theta_i(t) & \cos\Theta_i(t) \end{bmatrix}. \quad (3)$$

The global CoM $(x_{\mathrm{R}}, y_{\mathrm{R}})$ of a snake-like MR is located around/along the snake-like MR body, depending on its shape. The link's coordinates assist in determining the position of a snake-like MR $p_{\mathrm{R}} \in \mathbb{R}^2$ in global coordinates. The p_{R} of a snake-like MR considering locomotion on the xy-plane evaluated as follows:

$$p_{\mathrm{R}} = \begin{bmatrix} p_{\mathrm{R}x} \\ p_{\mathrm{R}y} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} y_i \end{bmatrix} \quad (4)$$

where x_i and y_i are the CoM of module joint in local coordinates. With help of above equations the locomotion of snake-like MR can be created and later visualized.

III. LOCOMOTION OF A SNAKE-LIKE MR

In this section, we first introduce the concept of providing snake-like locomotion to an MR. Then, we present our performance metric for evaluating energy-efficient locomotion. Finally, the problem formulation presents the need to optimize the locomotion by considering energy efficiency.

A. LOCOMOTION CONTROL

To control the locomotion of a snake-like MR, we use the serpenoid curve derived by Hirose [8]. Serially connected multi-DoF robot mechanism can use serpenoid curve based control to generate snake-like motion by adopting different motion curves.

The serpenoid curve can be generated using the following serpenoid equation:

$$\Phi_i(t) = \begin{cases} A_{\mathrm{m}} \sin(F_{\mathrm{m}} t + \delta_{\mathrm{m}} i), & \text{if } i \text{ is odd} \\ A_{\mathrm{f}} \cos(F_{\mathrm{f}} t + \delta_{\mathrm{f}} i), & \text{if } i \text{ is even.} \end{cases} \quad (5)$$

This equation generates a joint angle $\Phi_i(t)$ for each time step t. Changes in the joint angle cause changes in the shape and movement of a snake-like MR. We consider the male joints of the jth D-MR module, where $j \in \{1, 2, 3, \ldots, N\}$, as odd joints and the female joints as even joints.

The six parameters in (5) controls the locomotion of entire snake-like MR. The parameters A_{m} and A_{f} denote the amplitudes of the curve for even and odd joints, respectively. The phases δ_{m} and δ_{f} of the curves adjust the timing between even and odd joints, respectively, to produce motion. Multiplying the phase parameters with joint index i represents waves' propagation along with the snake-like MR. The frequency parameters F_{m} and F_{f} for both joints provide sine-wave-like and cosine-wave-like motion, respectively. The frequency parameters are multiplied by the time step t to determine the speed of the gait cycle.

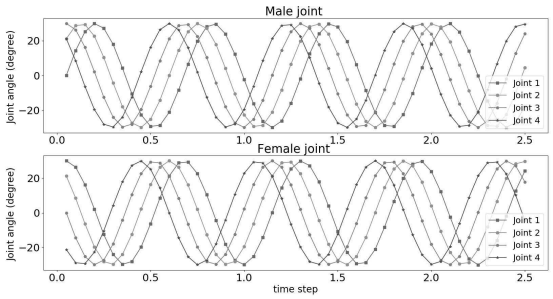

FIGURE 2. Angle variation of each joint based on (5).

A real snake has negligible separation between the joints in its body and can easily adjust its shape to resemble a perfect linear sinusoidal curve; in contrast, MR has a finite length 2α between two joints, restricting its motion. By using a locomotion strategy based on the serpenoid equation, a snake-like MR can achieve a nonlinear sinusoidal-curve-like shape. We can provide different shapes to a snake-like MR by varying the six parameters as mentioned above of the serpenoid equation. This study primarily focuses on linear progression motion obtained by maintaining constant values for the six parameters in (5) for a particular time [28], because linear progression motion conserves momentum and thus is more energy-efficient than other types of motion.

A snake-like MR uses linear progression motion to propel its body forward when each joint actuates at every time step t according to (5). As displayed in Fig. 2, the joint angle variations of each male and female joint of a module can be visualized as a sine wave and cosine wave, respectively, by maintaining the following values: $A_{\mathrm{m}} = A_{\mathrm{f}} = 30°$, $F_{\mathrm{m}} = F_{\mathrm{f}} = 10°$ and $\delta_{\mathrm{m}} = \delta_{\mathrm{f}} = 45°$. The aforementioned setting is suitable for moving a snake-like MR in the forward direction with constant velocity. To change the speed of a snake-like MR under linear progression motion, we must change the frequency parameters F_{m} and F_{f} by maintaining fixed values of $A_{\mathrm{m}} = A_{\mathrm{f}}$ and $\delta_{\mathrm{m}} = \delta_{\mathrm{f}}$. It is worth mentioning that other values of $A_{\mathrm{m}}, A_{\mathrm{f}}$, δ_{m} and δ_{f} are also possible for different gaits, as shown in Table 2. This study primarily focuses on linear progression gait because of its straightforward motion, higher velocity and lower power consumption compared to other gaits.

TABLE 2. Serpentine equation parameters and gaits.

Parameters	Gaits											
	Linear Progression			Side-winding			Lateral Rolling			Helix Rolling		
A_m	30	30	45	40	20	30	10	20	5	60	80	40
A_f	0	10	0	0	0	0	10	20	5	60	80	40
$\delta_{\mathrm{m}}, \delta_{\mathrm{f}}$	180°											

B. PERFORMANCE METRICS

Snake-like MR has limited energy; therefore, the need arises to control the gait to achieve economical locomotion. We consider two performance metrics to evaluate a gait's energy efficiency: the power consumed by a snake-like MR and its average forward velocity. The forward velocity and power consumption as a performance metric ensure snake-like MR moves faster with minimum power consumption [16].

1) POWER CONSUMPTION

For a snake-like MR with N modules (or $2N$ joints), the power consumption of the ith joint during time slot k, denoted as $P_{i,k}^{\mathrm{joint}}$, is the summation of the absolute product of the torque $\tau_{i,k,t}$ and angular velocity $\phi_{i,k,t}$ at each step t within the kth time slot [16]. The power consumption can be expressed as follows:

$$P_{i,k}^{\mathrm{joint}} = \sum_{t=1}^{T} |\tau_{i,k,t}\phi_{i,k,t}| \qquad \forall i \in \{1, \ldots, 2N\}. \tag{6}$$

The power consumed by the jth module during time slot k is denoted as $P_{j,k}^{\mathrm{module}}$, calculated by summing the total power consumption of both joint actuators in the j^{th} module as follows:

$$P_{j,k}^{\mathrm{module}} = P_{2j-1,k}^{\mathrm{joint}} + P_{2j,k}^{\mathrm{joint}} \qquad \forall j \in \{1, \ldots, N\}. \tag{7}$$

The power consumption of a snake-like MR with N modules during time slot k can be expressed as follows:

$$P_k^{\mathrm{R}} = \sum_{j=1}^{N} P_{j,k}^{\mathrm{module}}. \tag{8}$$

The total power consumed by a snake-like MR during a run can be expressed as follows:

$$P^{\mathrm{R}} = \sum_{k=1}^{K} P_k^{\mathrm{R}}. \tag{9}$$

2) AVERAGE FORWARD VELOCITY

The average velocity of a snake-like MR is the ratio between the distance covered in one-time slot k and the time taken to cover this distance [17]. Assume that the, head module of a snake-like MR is initially positioned at (x_{k-1}, y_{k-1}) and then it reaches (x_k, y_k) in the global frame after time Δk. In this

case, the average velocity of a snake-like MR in time slot k is as follows:

$$v_k^{\mathrm{R}} = \frac{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{\Delta k}. \tag{10}$$

The average forward velocity of a snake-like MR during a run can be found by averaging velocity of its head module as follows:

$$v^{\mathrm{R}} = \frac{\sum_{k=1}^{K} v_k^{\mathrm{R}}}{K}. \tag{11}$$

C. PROBLEM FORMULATION

By adjusting the control parameters F_{m} and F_{f}, the first objective P_k^{R} is minimized under the following constraints:

$$P^{\min} \leq P_k^{\mathrm{R}} \leq P^{\max}. \tag{12}$$

The second objective v_k^{R} is maximized under the following constraints:

$$v^{\min} \leq v_k^{\mathrm{R}} \leq v^{\max} \tag{13}$$

where $P^{\min}$ and $P^{\max}$ are the minimum and maximum power consumed by a snake-like MR during a time slot k, respectively, and $v^{\min}$ and $v^{\max}$ are the minimum and maximum average velocities of a snake-like MR during time slot k, respectively. One can easily find the minimum and maximum values by optimizing each objective individually [29].

In this study, the optimization process uses normalized objectives. Thus, we can obtain the following equations:

$$f_{\mathrm{p},k}^{\mathrm{norm}} = \frac{P^{\max} - P_k^{\mathrm{R}}}{P^{\max} - P^{\min}} \text{ and } f_{\mathrm{v},k}^{\mathrm{norm}} = \frac{v_k^{\mathrm{R}} - v^{\min}}{v^{\max} - v^{\min}}. \tag{14}$$

The optimization approach should maximize both objectives in (14). Thus, the energy-efficient gait optimization problem of a snake-like MR with two objectives is formulated as follows:

$$\begin{aligned} \max \quad & w \sum_{k=1}^{K} f_{\mathrm{p},k}^{\mathrm{norm}} + (1-w) \sum_{k=1}^{K} f_{\mathrm{v},k}^{\mathrm{norm}} \\ \text{subject to } & (8),\ (10),\ (12),\ \text{and } (13) \end{aligned} \tag{15}$$

where $w \in (0, 1)$ is the weighting coefficient reflecting relative objective importance, this weighting coefficient may change slowly with time. A natural way of solving (15) is to apply single-objective reinforcement learning (SORL). In this case, one Q-table is maintained to produce proper actions. However, when the weighting coefficient changes, the algorithm has to relearn and update the Q-table accordingly, causing an extra time cost. A better solution in response to possible weight changing must be conceived.

IV. MULTIOBJECTIVE REINFORCEMENT LEARNING ALGORITHM WITH FUZZY INFERENCE SYSTEM

This section develops a fuzzy inference system for reducing the number of possible states for a snake-like MR. The developed method enables the proposed FI-MORL algorithm to select an appropriate observation state by simultaneously considering energy consumption and forward velocity, which expedites the learning process. Moreover, we develop an FI-MORL algorithm that can rapidly learn the locomotion strategy of a snake-like MR when weight change occurs.

Suppose that a snake-like MR is performing a task in an uncertain environment. Snake-like MR can use machine learning techniques for gait parameter optimization, and the RL algorithm is one of the most popular machine learning algorithms. The RL algorithm can solve a complex problem with or without prior knowledge of the uncertain environment. Let $\mathcal{S}$ be the state space and s_k be the state during time slot k, where $s_k \in \mathcal{S}$. Given s_k, the agent selects an action a_k from its action set $\mathcal{A}$. It then proceeds to the next state s_{k+1} and receives a reward R_{k+1} as feedback from the environment. The agent continues to repeat this process with new experiences until it reaches optimality.

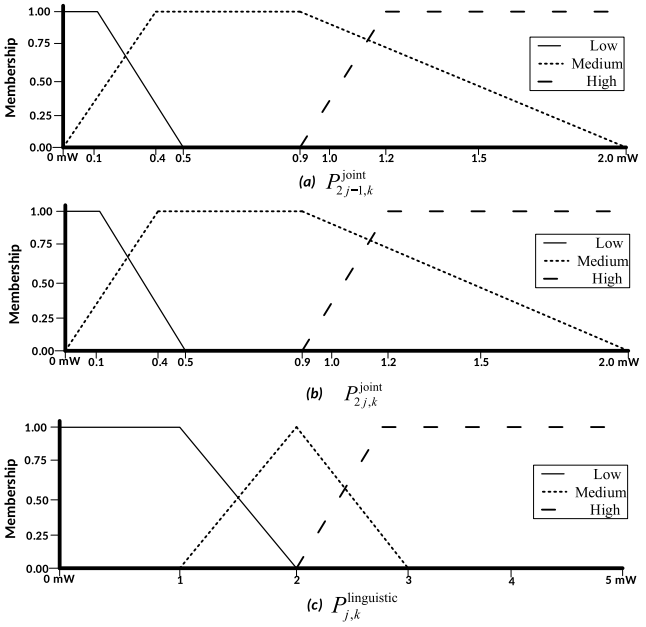

FIGURE 3. Membership functions of the fuzzy inference system: (a and b) membership functions of both joints, and (c) membership functions of a module.

Q-learning is a type of model-free learning and a simple method for enabling the agent to learn how to act optimally in a Markov decision process [30]. In Q-learning, an agent takes action a_k, in-state s_k by following a ϵ-greedy policy. It then receives a reward R_{k+1} with next state, s_{k+1} and updates the Q-table by using the following equation during every time slot k:

$$\begin{aligned} Q(s_k, a_k) \leftarrow\ & (1-\alpha) Q(s_k, a_k) \\ & + \alpha (R_{k+1} + \gamma \max_a Q(s_{k+1}, a)) \end{aligned} \tag{16}$$

where α and γ are the learning rate and discount factor, respectively.

To apply Q-learning and update the Q-table for a snake-like MR, it needed to convert the sensor's information from continuous states into discrete states.

The state received from the environment can be sensor information or calculated value. Including the power consumed by each module as an observation state is advantageous for achieving energy efficiency. This state selection information helps the agent understand which modules are consuming a high amount of energy so that the proposed learning algorithm can adjust the movement of modules to achieve a tradeoff between both objectives. The power consumed by a module has a continuous value; however, the algorithm requires discrete states to update the Q-table. Fuzzy inference system can construct a discrete state because fuzzification is a method of generalizing the Boolean concept to partial truth or false [31]. Fuzzy inference system with RL algorithms implemented in [32], [33], and can be supportive to design a system that can reduce states in an environment through human reasoning, known as "Rules."

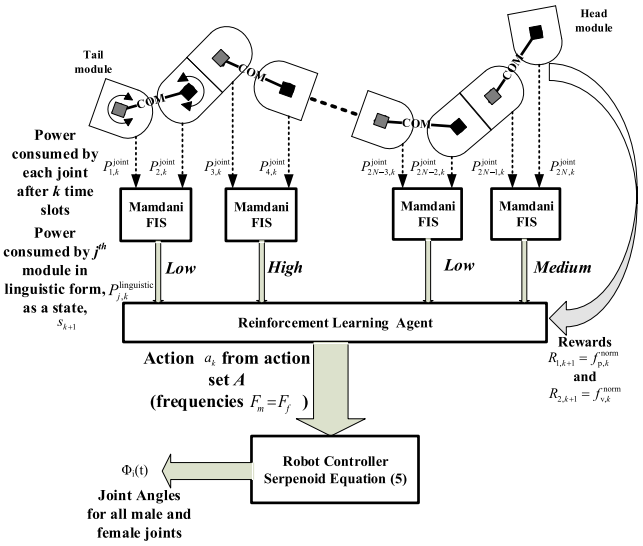

FIGURE 4. Relationship between robot models, components of RL, and Mamdani fuzzy inference system.

Using a fuzzy inference system for each module of a snake-like MR, fuzzy inference system can obtain the discrete state for each time slot k. This discrete state with rule-based expert knowledge is suitable for applying in RL algorithm and results in fast learning. Fig. 4 presents the relationship between the robot models, components of RL, and Mamdani fuzzy inference system. Let s_k be the current state, given s_k, a snake-like MR selects an action a_k from $\mathcal{A}$. It then proceeds to the next state s_{k+1} and receives the rewards $R_{1,k} = f_{\mathrm{p},k}^{\mathrm{norm}}$ and $R_{2,k} = f_{\mathrm{v},k}^{\mathrm{norm}}$. The action set $\mathcal{A}$ represents the different frequencies ($F_{\mathrm{m}} = F_{\mathrm{f}}$) in the discrete form, which lead to different speeds during each time slot k. The next state s_{k+1} is obtained by inferring each fuzzy inference system.

During time slot k, both joints of each module of a snake-like MR consume some power to move a certain distance. The power consumption of the two joints of the

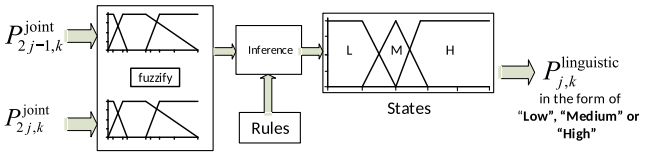

FIGURE 5. Mamdani fuzzy inference system for each module.

jth module, which are denoted as $P_{2j-1,k}^{\mathrm{joint}}$ and $P_{2j,k}^{\mathrm{joint}}$, can then be used as input for the fuzzy inference system. In this study, the Mamdani fuzzy inference system [34] is used for each module to obtain a linguistic form of the power consumed by the jth module during time slot k (i.e., $P_{j,k}^{\mathrm{linguistic}}$) as an output by using min-max operations. The Mamdani fuzzy inference system of a module is in Fig. 5.

Triangular and trapezoidal membership functions are used in this study (Fig. 3). These membership functions can overlap each other and indicates the degree to which a given value belongs to different fuzzy sets [35]. The membership functions for fuzzy set A and B are denoted as $\mu_A(P_{2j-1,k}^{\mathrm{joint}})$ and $\mu_B(P_{2j,k}^{\mathrm{joint}})$, respectively, where $A = B = \{$"Low", "Medium", "High"$\}$ represents the set of linguistic terms (fuzzy labels) for each input variable. The parameter $\mu_C(P_{j,k}^{\mathrm{linguistic}})$ represents the membership functions for fuzzy set $C = \{$"Low", "Medium", "High"$\}$, where C represents the linguistic terms for the output variable $P_{j,k}^{\mathrm{linguistic}}$.

Given the inputs to these membership functions, the resultant values are between 0 and 1 for each joint of the jth module. After the linguistic terms and membership function values are defined, expert knowledge can formulate the fuzzy inference system rules.

The rule u to obtain $P_{j,k}^{\mathrm{linguistic}}$ can be expressed as follows

$$\begin{aligned}&\text{Rule } u:\\ &\text{If} \quad P_{2j-1,k}^{\mathrm{joint}} \text{ is } A_u \text{ and } P_{2j,k}^{\mathrm{joint}} \text{ is } B_u\\ &\text{Then} \quad P_{j,k}^{\mathrm{linguistic}} \text{ is } C_u\end{aligned}$$

for $u = 1, 2, \ldots, U$, where U is the maximum number of constructed rules; $A_u \in A$ and $B_u \in B$ are the linguistic terms from fuzzy sets A and B, respectively; and $C_u \in C$ is the output variable's linguistic term from fuzzy set C.

On the basis of the values of $P_{2j-1,k}^{\mathrm{joint}}$ and $P_{2j,k}^{\mathrm{joint}}$ during time slot k and the obtained membership function values, n rules can be fired. Let $I = \{I_1, I_2, I_3, \ldots, I_n\}$ be the set containing the indices of all the fired rules out of all U rules. With more than one input variable combined with the "and" logical connection under all the n fired rules, the truth value of the combined proposition of each fired rule $\mathcal{X}_{I_i}$ can be obtained as follows:

$$\begin{aligned}\mathcal{X}_{I_i} = \min(&\mu_{A_{I_i}}(P_{2j-1,k}^{\mathrm{joint}}), \mu_{B_{I_i}}(P_{2j,k}^{\mathrm{joint}}))\\ &\forall\, I_i \in I \text{ and } i = 1, 2, 3, \ldots, n.\end{aligned} \tag{17}$$

The index of a fired rule having maximum truth value n^* and its corresponding rule index I_{rule} can be obtained through the argmax operation within the truth values of the fired n rules

TABLE 3. Fuzzy rules of the fuzzy membership functions in Fig. 3.

Rule u		A_u		B_u		C_u
1	If $P^{\text{joint}}_{2j-1,k}$ is	**Low**	and $P^{\text{joint}}_{2j,k}$ is	**Low**	Then $P^{\text{linguistic}}_{j,k}$ is	**Low**
2	If $P^{\text{joint}}_{2j-1,k}$ is	**Low**	and $P^{\text{joint}}_{2j,k}$ is	**Medium**	Then $P^{\text{linguistic}}_{j,k}$ is	**Medium**
3	If $P^{\text{joint}}_{2j-1,k}$ is	**Low**	and $P^{\text{joint}}_{2j,k}$ is	**High**	Then $P^{\text{linguistic}}_{j,k}$ is	**Medium**
4	If $P^{\text{joint}}_{2j-1,k}$ is	**Medium**	and $P^{\text{joint}}_{2j,k}$ is	**Low**	Then $P^{\text{linguistic}}_{j,k}$ is	**Medium**
5	If $P^{\text{joint}}_{2j-1,k}$ is	**Medium**	and $P^{\text{joint}}_{2j,k}$ is	**Medium**	Then $P^{\text{linguistic}}_{j,k}$ is	**Medium**
6	If $P^{\text{joint}}_{2j-1,k}$ is	**Medium**	and $P^{\text{joint}}_{2j,k}$ is	**High**	Then $P^{\text{linguistic}}_{j,k}$ is	**High**
7	If $P^{\text{joint}}_{2j-1,k}$ is	**High**	and $P^{\text{joint}}_{2j,k}$ is	**Low**	Then $P^{\text{linguistic}}_{j,k}$ is	**Medium**
8	If $P^{\text{joint}}_{2j-1,k}$ is	**High**	and $P^{\text{joint}}_{2j,k}$ is	**Medium**	Then $P^{\text{linguistic}}_{j,k}$ is	**High**
9	If $P^{\text{joint}}_{2j-1,k}$ is	**High**	and $P^{\text{joint}}_{2j,k}$ is	**High**	Then $P^{\text{linguistic}}_{j,k}$ is	**High**

as follows:

$$n^* = \arg\max_n(\mathcal{X}_{I_1}, \mathcal{X}_{I_2}, \ldots, \mathcal{X}_{I_n})$$
$$I_{\text{rule}} = I_{n^*} \tag{18}$$

where $I_n \in I$. The output of the fuzzy inference system, namely $P^{\text{linguistic}}_{j,k}$ (Fig. 5) can be expressed as follows:

$$P^{\text{linguistic}}_{j,k} = C_{I_{\text{rule}}}. \tag{19}$$

This inferred linguistic term $P^{\text{linguistic}}_{j,k}$ is a state in the RL algorithm. Nine rules are used for each fuzzy inference system (details are presented in Table 3).

When optimizing the gait parameters of a snake-like MR, it is necessary to determine the importance of objectives during learning. In a specific situation, the importance of objectives must change with changing weight. The learning method should not take long to reach a steady-state in this situation. The maximization problem in (15) can be solved using the SORL optimization method; however, the proposed FI-MORL optimization method is more effective than SORL optimization during the change in weights of objectives to provide one objective with higher importance than other objectives.

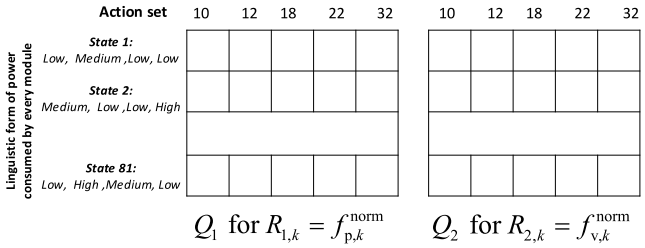

FIGURE 6. Representation of the Q-table for the proposed FI-MORL algorithm.

In the proposed FI-MORL algorithm two Q-tables for power consumption and forward velocity are initialized randomly. The Q-tables are displayed in Fig. 6, where Q_1 is the Q-table for $R_{1,k} = f_{\text{p},k}$ and Q_2 is the Q-table for $R_{2,k} = f_{\text{v},k}$. The observation state $s_k \in \mathcal{S}$ in both Q-tables represents the linguistic output from each fuzzy inference system after each time slot k. This state is designed as follows for N modules:

$$s_k = [P^{\text{linguistic}}_{1,k}, P^{\text{linguistic}}_{2,k}, P^{\text{linguistic}}_{3,k}, \ldots, P^{\text{linguistic}}_{N,k}]. \tag{20}$$

The action set $\mathcal{A}$ comprises different sets of frequencies ($F_{\text{m}} = F_{\text{f}}$) in the two Q-tables. During learning, the rewards $R_{1,k}$ and $R_{2,k}$ are designed to encourage the agent to maximize the objectives in (15).

The Q-table representing these two objectives can be updated as follows:

$$Q_i(s_k, a_k) \leftarrow (1-\alpha)Q_i(s_k, a_k) + \alpha(R_{i,k+1} + \gamma \max_a Q_i(s_{k+1}, a)), \tag{21}$$

for $i = 1, 2$.

Algorithm 1 presents the pseudocode of the proposed FI-MORL algorithm. The details of this algorithm describe the following texts. The Q-tables for both objectives are initialized according to Fig. 6. At the start of each episode in line 2, the snake-like MR is initialized at the origin. One episode contains K steps from time slot 1 to time slot K. The agent selects an action a_k in line 5 according to the weighted-sum Q-table ($wQ_1(s_k, \cdot) + (1-w)Q_2(s_k, \cdot)$) to consider the weight effect. Then, the agent takes the action a_k and obtains the rewards $R_{1,k}$ and $R_{2,k}$ as well as the next state s_{k+1} in line 6. In line 7, both Q-tables, namely Q_1 of $f^{\text{norm}}_{\text{p},k}$ and Q_2 of $f^{\text{norm}}_{\text{v},k}$, are updated. The algorithm continues to repeat the process of lines 3-8 until termination occurs. The termination of the episode occurs when the snake-like MR reaches a particular destination. After Q-tables, Q_1 and Q_2 reach steady values, the best objective value $wQ_1 + (1-w)Q2$ can be achieved in a few learning episodes when a weight change occurs.

V. SIMULATION RESULTS

We simulated our snake-like MR in a virtual robot experimentation platform (V-REP) [36], which is a physics engine for simulating robot models according to real experiences. The V-REP software receives information regarding the joint angles of a snake-like MR in every time step t from a Python script to perform locomotion. A snake-like MR consisting of four D-MR modules simulated in V-REP and Python script generates the joints angle based on (5). Each D-MR module

Algorithm 1 Proposed FI-MORL for Energy-Efficient Gait Optimization

Require: learning rate $\alpha \in (0, 1]$, exploration rate $\epsilon > 0$, discount factor γ, weight w.
Initialize $Q_i(s_k, a_k)$ for all $s_k \in \mathcal{S}$, $a_k \in \mathcal{A}$, arbitrarily.
Ensure: control parameters ($F_\mathrm{m} = F_\mathrm{f}$) to ensure minimum P^R and maximum v^R.
1: **Loop** for each episode:
2: Initialize the snake-like MR at origin.
3: **Loop** for every time slot k in an episode:
4: Obtain current state s_k from (20).
5: Choose a_k using ϵ-greedy derived from $wQ_1(s_k, \cdot) + (1 - w)Q_2(s_k, \cdot)$.
6: Take action a_k by setting control parameters in (5), observe rewards $R_{1,k}$, $R_{2,k}$ and s_{k+1}.
7: Update $Q_1(s_k, a_k)$ and $Q_2(s_k, a_k)$ using (21).
8: **Until** s_k is terminal

had two joints that rotate along the z-axis in a range of $-90°$ to $90°$ and has a maximum actuator torque of 0.52 Nm. During the experiment, the following amplitude and phase values were used for each male and female joint: $A_m = A_f = 30°$ and $\delta_\mathrm{m} = \delta_\mathrm{f} = 180°$. The frequency values F_m and F_f were set to be equal and considered as different actions in $\mathcal{A}$. These frequencies, representing the five actions $F_\mathrm{m} = F_\mathrm{f} \in \{10°, 12°, 18°, 22°, 31°\}$, vary the speed of the snake-like MR and result in different values of power consumption and average velocity during each time slot k. The control parameters mentioned above are applied in (5) to generate a gait for the snake-like MR during each time slot k. Each episode contains K learning steps, and one learning step or time slot k of the learning algorithms contains $t = 10$ time steps for V-REP.

The sensors inside V-REP facilitate calculating the power consumption and average velocity of the snake-like MR. The sensor values were transmitted to a Python script asynchronously as a feedback signal after every time step t. The time steps t in (5) for the snake-like MR controller varied from 0 to 950 ms in an interval of 50 ms (as per the V-REP default value) because ten sets of gait values were required for a one-time slot k. As the snake-like MR moved linearly along the x-direction, an episode terminates when the moving distance is 0.5 m. During this run, the total power consumption P^R and average velocity v^R were obtained using (9) and (11), respectively.

We compared the proposed method with FI-SORL and benchmark deep SORL algorithms, namely deep Q-networks (DQN) [37], PPO [38], actor-critic (AC) [39] and vanilla policy gradient (VPG) [40]. These benchmark algorithms used Tensorforce, which is an open-source deep RL framework [41], built on top of Tensorflow framework [42] and compatible with Python. The FI-SORL method used the same observation states (obtained with the fuzzy inference system) as the proposed method. The major difference between the FI-MORL and FI-SORL is that the FI-MORL maintains two

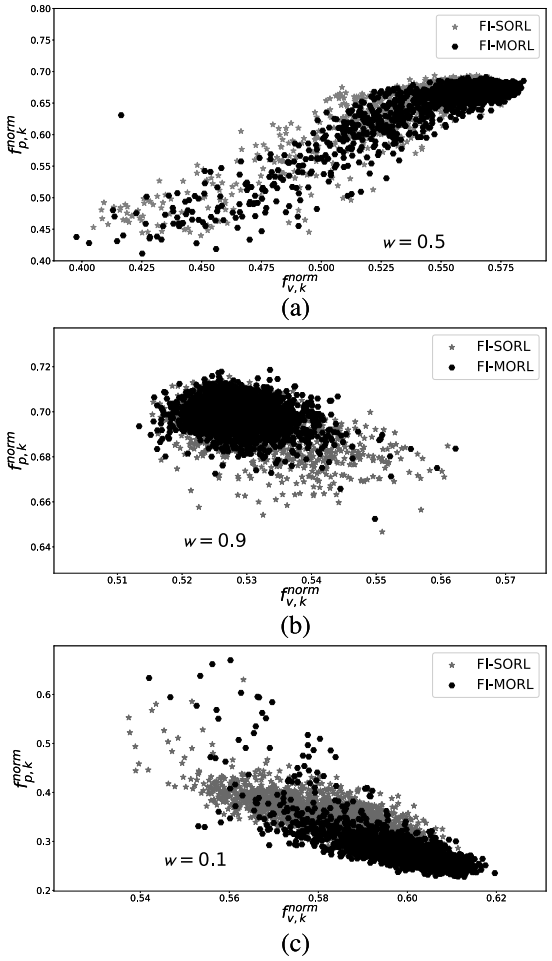

FIGURE 7. Scatter plots of the objective values of the proposed FI-MORL method and the FI-SORL method: (a) equal objective importance with $w = 0.5$, (b) higher importance of the power objective with $w = 0.9$, and (c) higher importance of the velocity objective when $w = 0.1$.

Q-tables for the value update while the FI-SORL has only one Q-table during the learning and execution process. As such, FI-MORL uses separate reward signals associated with power consumption and forward velocity; FI-SORL uses combined reward signals (a weighted sum) as follows:

$$U_\mathrm{R} = w f_{\mathrm{p},k}^{\mathrm{norm}} + (1 - w) f_{\mathrm{v},k}^{\mathrm{norm}}. \tag{22}$$

The other methods used continuous states derived from the torque angular velocity of each joint (also used for calculating the power consumed by each joint in the fuzzy inference system input) and angular position of each joint. The action settings were the same for all the methods. The parameter U_R in (22) was used as reward for all methods, and the Q-table in FI-SORL was updated using (16). The weighting coefficient w was used to balance the tradeoff between power consumption and average velocity in the reward U_R of all SORL algorithms.

The simulation was executed ten times with the results averaged to neglect the randomness of all algorithms. In our simulations, w was initially maintained at 0.5 (both power

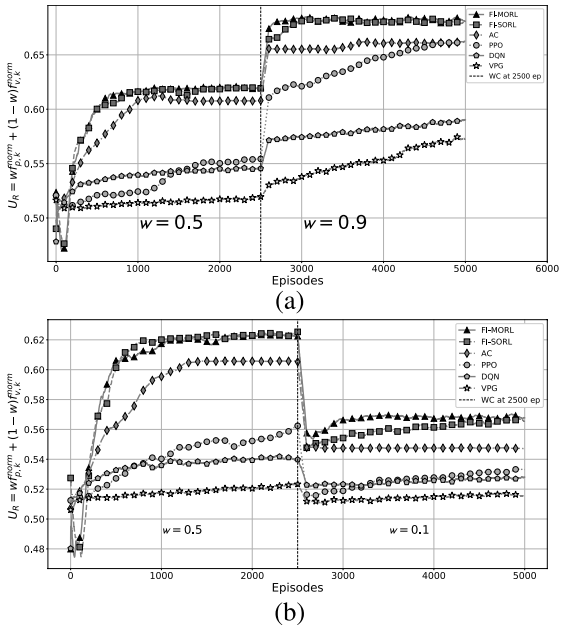

FIGURE 8. Comparison of the learning curves before and after a weight change during episode 2501: (a) from equal importance for both objectives to higher importance for the power objective and (b) from equal importance for both objectives to higher importance for the velocity objective.

TABLE 4. Steady-state values obtained with various learning methods.

Average of last 100 episodes values after reaching steady-state for $w = 0.5$			
Method	Power Consumption P_R	Average Velocity v_R	Average Return U_R
FI-SORL	124.33 mW	0.0264 m/s	0.6181
FI-MORL	122.83 mW	0.0268 m/s	0.6189
DQN [37]	141.39 mW	0.0245 m/s	0.5458
PPO [38]	144.49 mW	0.0236 m/s	0.5538
AC [39]	126.76 mW	0.0259 m/s	0.6074
VPG [40]	147.71 mW	0.0241 m/s	0.5193

and velocity with equal importance). After all the matching algorithms reached a steady state, the weights were changed to 0.9 and 0.1 to reflect the relative importance of the power and velocity objectives, respectively. The simulations were performed using a desktop with an Intel(R) i5-9400F, 2.90 GHz CPU with 16 GB of RAM.

Fig. 8 displays the plot of the average reward U_{R} during each episode. The plot averages over every 100 episodes to evaluate which algorithm is the quickest to learn and reach a steady-state under equal objective importance ($w = 0.5$) until 2500 episodes. As displayed in Fig. 8(a), the proposed algorithm and FI-SORL algorithm reached a steady-state after approximately 800 and were faster than the other algorithms. The proposed FI-MORL achieved an average of 14% higher rewards than DQN, PPO, and VPG. Also, it was able to outperform AC and FI-SORL by a sheer 1%. This result is due to the proposed approach involving a fuzzy inference system to approximate the state space, thus reducing the number of states to explore. The proposed and FI-SORL methods have finite states (only 81 linguistic states), whereas the other methods have infinite possible states because they consider continuous values. Another reason explains that the proposed method and FI-SORL approach are simple and involve learning only through Q-tables, whereas the other methods involve using deep learning networks, which require considerable computational resources.

The proposed algorithm reached a steady-state after approximately 800 episodes, ensuring that the proposed approach achieved a minimum power consumption and maximum average velocity faster than the benchmark deep SORL methods with $w = 0.5$. Fig. 9(a) to 9(d) depicts the variations in the power consumption and average velocity over each episode, respectively. In the beginning, the proposed FI-MORL approach and FI-SORL underperformed other algorithms and thus their boxplots associated with the velocity and power in Figs. 9(b) and 9(d) contained some outliers. After 500 episodes, the FI-MORL, FI-SORL, and AC performed competently in reducing power consumption and increasing velocity. At the steady-state, the proposed FI-MORL outperformed DQN, PPO, and VPG by consuming an average of 14% less power and attaining 11% higher velocity. Thus in the power consumption boxplot Fig. 9(b), the interquartile range for FI-MORL remains the lowest than other algorithms. Similarly, in average velocity boxplot Fig. 9(d), the interquartile range for the proposed method remains the highest. Meanwhile, AC and FI-SORL performed equally well, yet the proposed method overtook them by consuming an average of 2% less power and gaining 2.5% higher velocity. Table 4 summarizes the steady-state values of comparable methods (the average of the final 100 episodes). The simulation results indicate that the proposed method can find a solution faster than comparable methods when both objectives are equally weighted.

We then changed the weights during the learning process to determine the speed with which an algorithm reached a steady-state balance between the power consumption and forward velocity of a snake-like MR.

As displayed in Fig. 8, the weight change occurred at 2500 episodes. The learning plots in Fig. 8(a) and 8(b) indicate that the proposed method can reach a steady state faster than the other comparable methods after a weight change. Two weight settings were considered: from $w = 0.5$ to $w = 0.9$ and from $w = 0.5$ to $w = 0.1$. The objective with a higher weight was considered more important than the other. The weights were changed to 0.9 (more emphasis on power consumption) and 0.1 (more emphasis on forwarding velocity) from 0.5 after 2500 episodes. To reach a steady-state, the FI-SORL method required approximately 1000 episodes after the weight was changed to 0.9 [Fig. 8(a)] and approximately 2000 episodes after the weight was changed to 0.1 [Fig. 8(b)]. The FI-MORL obtained the best weighted-sum value faster than all the algorithms, including the benchmark deep SORL algorithms.

Fig. 7 presents a scatter plot of $f_{\mathrm{p},k}^{\mathrm{norm}}$ and $f_{\mathrm{v},k}^{\mathrm{norm}}$ objective values during each episode for the FI-MORL and FI-SORL algorithms. Fig. 7(a) indicates the scatter points for the

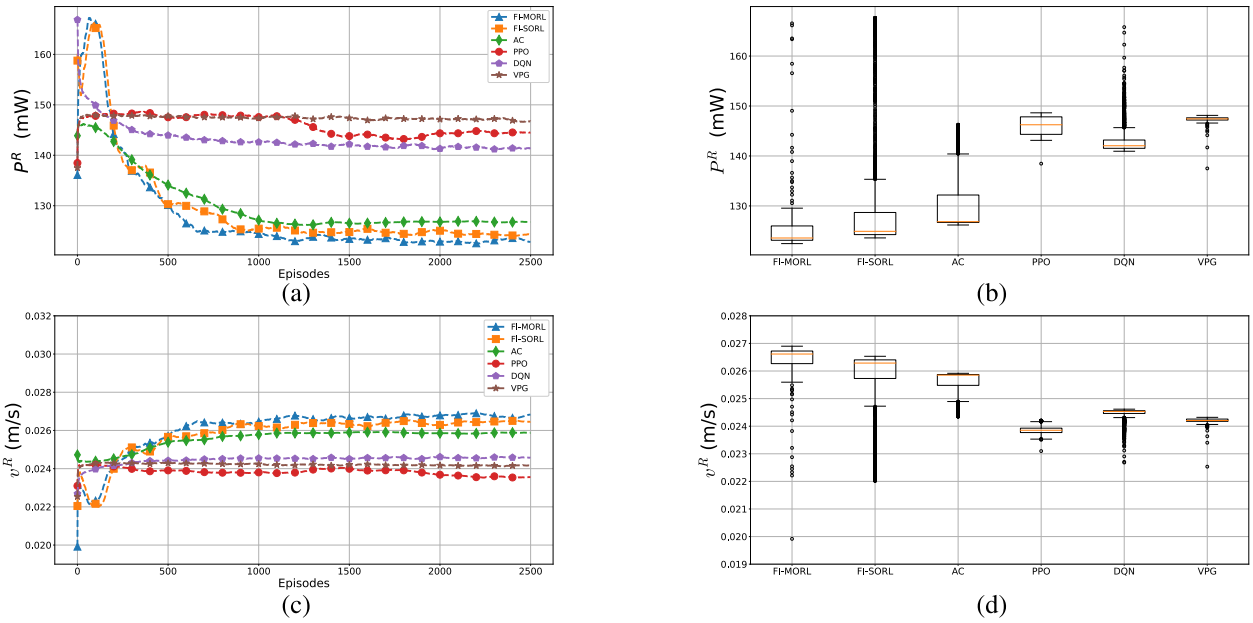

FIGURE 9. (a), (b) Power consumption and (c), (d) average velocity of a snake-like MR during each run when $w = 0.5$.

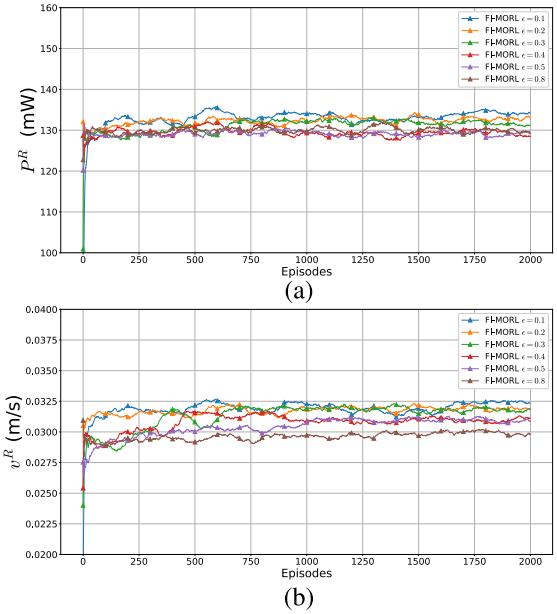

FIGURE 10. Different values of exploration rate ϵ and its effect on (a) power consumption and (b) average velocity.

objective values are confined toward the region where both objectives have equal importance ($w = 0.5$). When the weight changes to 0.9, the scatter points of FI-MORL in Fig. 7(b) are confined toward the expected region as per the weight setting (the y-axis representing the power consumption objective, which has higher importance than the velocity objective); however, for FI-SORL, the scatter points are distributed in different areas. When the weight changes to 0.1, the scatter points of FI-MORL in Fig. 7(c) follow the weight setting and are confined toward x-axis (which represents the velocity objective, which has higher importance than the power consumption objective); however, the scatter points of FI-MORL are distributed in a different area.

The results shown in Figs. 8 and 7 indicate that after a weight change, the FI-MORL method achieves the best objective values faster than the FI-SORL methods do and exhibits the desired weight change effect.

Although our model provided better results than existing ones, several threats that can affect its performance exist. For example, our model adopted a discrete action set. A continuous action set can be chosen to provide higher resolution so that a larger action space can be explored, but this can increase model complexity and time cost.

For Q-learning, the exploration rate ϵ was maintained at 0.1 throughout the training process. Fig. 10 shows the learning curves of various values of ϵ ranging from 0.1 to 0.8. The changes in power consumption during $\epsilon \in [0.1, 0.8]$ varied less than 3% compared with $\epsilon = 0.1$. The average velocity fluctuation remained less than 5% during $\epsilon \in [0.1, 0.5]$ and increased by 7% if $\epsilon = 0.8$. The small changes in average velocity and power consumption illustrated that the proposed method was insensitive to the variation of the exploration rate if $\epsilon \in [0.1, 0.5]$.

VI. CONCLUSION

This study investigates the energy-efficient gait optimization of snake-like MRs using multiobjective reinforcement learning with a fuzzy inference system, presenting a weighted-sum problem formulation for the power consumption and average velocity of a snake-like MR. In the literature, objective weights are assumed to be fixed without considering possible changes over time. After a weight change, the proposed

FI-MORL algorithm rapidly achieved the best steady-state objective values. Deep SORL based optimization methods required a longer learning time than the proposed method due to deep learning networks. By contrast, the proposed method involved using a fuzzy inference system to reduce the number of possible states and achieve rapid Q-table based learning. After a weight change, all the SORL based approaches required additional learning time to reach steady-state values, whereas the proposed method determined a 14% faster steady-state weighted-sum value compared to the traditional method and 1% faster than the AC and FI-SORL.

REFERENCES

[1] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, "Current trends in reconfigurable modular robots design," *Int. J. Adv. Robotic Syst.*, vol. 14, no. 3, May 2017, Art. no. 1729881417710457.

[2] B. Liu, M. Liu, X. Liu, X. Tuo, X. Wang, S. Zhao, and T. Xiao, "Design and realize a snake-like robot in complex environment," *J. Robot.*, vol. 2019, pp. 1–9, Feb. 2019.

[3] S. Feng, "Design, analysis, planning, and control of a novel modular self-reconfigurable robotic system," Ph.D. dissertation, Dept. Mech. Eng., Virginia Tech, Blacksburg, VA, USA, 2022.

[4] M. Tesch, J. Schneider, and H. Choset, "Expensive multiobjective optimization for robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 973–980.

[5] V. K. Pediredla, S. Annamraju, A. R. N., and A. Thondiyath, "Enhancement of high-fidelity haptic feedback through multimodal adaptive robust control for teleoperated systems," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5526–5536, Dec. 2021.

[6] S. Ali, A. Ashraf, S. B. Qaisar, M. Kamran Afridi, H. Saeed, S. Rashid, E. A. Felemban, and A. A. Sheikh, "SimpliMote: A wireless sensor network monitoring platform for oil and gas pipelines," *IEEE Syst. J.*, vol. 12, no. 1, pp. 778–789, Mar. 2018.

[7] J. Liu, Y. Tong, and J. Liu, "Review of snake robots in constrained environments," *Robot. Auto. Syst.*, vol. 141, Jul. 2021, Art. no. 103785.

[8] T. Owen, "Biologically inspired robots: Snake-like locomotors and manipulators by shigeo hirose," *Robotica*, vol. 12, no. 3, pp. 282–284, 1994.

[9] R. Ariizumi and F. Matsuno, "Dynamic analysis of three snake robot gaits," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1075–1087, Oct. 2017.

[10] S. Ma, "Analysis of snake movement forms for realization of snake-like robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Dec. 1999, pp. 3007–3013.

[11] M. Dehghani and M. J. Mahjoob, "A modified serpenoid equation for snake robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Feb. 2009, pp. 1647–1652.

[12] K.-H. Chang and Y.-Y. Chen, "Efficiency on snake robot locomotion with constant and variable bending angles," in *Proc. IEEE Workshop Adv. Robot. Social Impacts*, Aug. 2008, pp. 1–5.

[13] M. H. A. Majid, M. R. Khan, and S. N. Sidek, "Development of wheel-less snake robot with two distinct gaits and gait transition capability," *Int. J. Autom. Comput.*, vol. 10, no. 6, pp. 534–544, Dec. 2013.

[14] Z. Zhou, H. Wang, D. Li, and H. Deng, "Motion control curve of snake-like robot based on centroid stability," in *Proc. IEEE Int. Conf. Unmanned Syst.*, Oct. 2019, pp. 826–830.

[15] D. Li, C. Wang, H. Deng, and Y. Wei, "Motion planning algorithm of a multi-joint snake-like robot based on improved serpenoid curve," *IEEE Access*, vol. 8, pp. 8346–8360, 2020.

[16] Z. Bing, C. Lemke, Z. Jiang, K. Huang, and A. Knoll, "Energy-efficient slithering gait exploration for a snake-like robot based on reinforcement learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5663–5669.

[17] E. Kelasidi, M. Jesmani, K. Y. Pettersen, and J. T. Gravdahl, "Multi-objective optimization for efficient motion of underwater snake robots," *Artif. Life Robot.*, vol. 21, no. 4, pp. 411–422, Nov. 2016.

[18] D. J. Christensen, U. P. Schultz, and K. Stoy, "A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots," *Robot. Auto. Syst.*, vol. 61, no. 9, pp. 1021–1035, Sep. 2013.

[19] A. Sproewitz, R. Moeckel, J. Maye, and A. J. Ijspeert, "Learning to move in modular robots using central pattern generators and online optimization," *Int. J. Robot. Res.*, vol. 27, nos. 3–4, pp. 423–443, Mar. 2008.

[20] A. Crespi and A. J. Ijspeert, "Online optimization of swimming and crawling in an amphibious snake robot," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 75–87, Feb. 2008.

[21] W. S. Chee, J. Teo, and K. Kinabalu, "Empirically comparing three multi-objective optimization approaches for the automated evolution of snake-like modular robots," in *Proc. Int. Conf. Artif. Intell. Pattern Recognit.*, Kuala Lumpur, Malaysia, Nov. 2014, pp. 175–183.

[22] X. Wu and S. Ma, "CPG-based control of serpentine locomotion of a snake-like robot," *Mechatronics*, vol. 20, no. 2, pp. 326–334, Mar. 2010.

[23] Z. Cao, D. Zhang, and M. Zhou, "Modeling and control of hybrid 3-D gaits of snake-like robots," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4603–4612, Oct. 2021.

[24] Alberto. *Dtto Explorer Modular Robot*. Accessed: Nov. 8, 2019. [Online]. Available: https://github.com/otrebla333/Dtto-Modular-Robot

[25] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[26] P. Liljebäck, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl, "A 3D motion planning framework for snake robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL USA, Sep. 2014, pp. 1100–1107.

[27] P. Liljeback, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl, *Snake Robots: Modelling, Mechatronics, and Control*. Cham, Switzerland: Springer, 2013.

[28] K. Melo, M. Hernandez, and D. Gonzalez, "Parameterized space conditions for the definition of locomotion modes in modular snake robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Guangzhou, China, Dec. 2012, pp. 2032–2038.

[29] W. Jakob and C. Blume, "Pareto optimization or cascaded weighted sum: A comparison of concepts," *Algorithms*, vol. 7, no. 1, pp. 166–185, Mar. 2014.

[30] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 4868–4878.

[31] B. Bede, *Mathematics of Fuzzy Sets and Fuzzy Logic*. London, U.K.: Springer, Jan. 2013.

[32] N. Kumar, S. S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4919–4928, Aug. 2021.

[33] A. Bonarini, A. Lazaric, F. Montrone, and M. Restelli, "Reinforcement distribution in fuzzy Q-learning," *Fuzzy Sets Syst.*, vol. 160, no. 10, pp. 1420–1443, May 2009.

[34] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Hum.-Comput. Stud.*, vol. 51, no. 2, pp. 135–147, 1999.

[35] B. Bede and I. J. Rudas, "Takagi–Sugeno approximation of a Mamdani fuzzy system," in *Advance Trends in Soft Computing*, vol. 312. Cham, Switzerland: Springer, Dec. 2014, pp. 293–300.

[36] R. Byrtus and J. Vechetová, "Trident snake robot motion simulation in V-REP," in *Proc. Int. Conf. Modeling Simulation Auto. Syst.* Cham, Switzerland: Springer, 2018, pp. 27–42.

[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[39] V. R. Konda and V. S. Borkar, "Actor-critic-type learning algorithms for Markov decision processes," *SIAM J. Control Optim.*, vol. 38, no. 1, pp. 94–123, Jan. 1999.

[40] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2219–2225.

[41] A. Kuhnle, M. Schaarschmidt, and K. Fricke. *Tensorforce: A Tensorflow Library for Applied Reinforcement Learning*. Accessed: Nov. 20, 2019. [Online]. Available: https://github.com/tensorforce/tensorforce

[42] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Conf. Oper. Syst. Design Implement.* Savannah, GA, USA: USENIX Association, Oct. 2016, pp. 265–283.

AKASH SINGH received the M.S. degree in electrical engineering from the National Tsing Hua University (NTHU), Taiwan, in 2020. He has published seven papers in IEEE, Elsevier, and other conferences in robotics and communication. His research interests include robotics, reinforcement learning, data science, and deep learning.

WEI-YU CHIU (Senior Member, IEEE) received the Ph.D. degree in communications engineering from the National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2010. He is currently an Associate Professor of electrical engineering with NTHU. His research interests include multiobjective optimization and reinforcement learning, and their applications to control systems, robotics, and smart energy systems. He was a recipient of the Youth Automatic Control Engineering Award bestowed by Chinese Automatic Control Society, in 2016, the Outstanding Young Scholar Academic Award bestowed by Taiwan Association of Systems Science and Engineering, in 2017, the Erasmus+Programme Fellowship funded by European Union (staff mobility for teaching), in 2018, and the Outstanding Youth Electrical Engineer Award bestowed by Chinese Institute of Electrical Engineering, in 2020. From 2015 to 2018, he had been serving as an Organizer and the Chair for the International Workshop on Integrating Communications, Control, and Computing Technologies for Smart Grid (ICT4SG). He is a Subject Editor of *IET Smart Grid*.

SHRI HARISH MANOHARAN received the B.E. degree in electrical engineering from Anna University, India, in 2010, and the M.Tech. degree in robotics from SRM University, India, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering with the National Tsing Hua University, Hsinchu, Taiwan. His research interests include multiobjective evolutionary algorithms, machine learning, and robotics.

ALEXEY M. ROMANOV (Senior Member, IEEE) received the degree (Hons.) in mechatronic engineering and the Ph.D. degree in electrical and electronics engineering from MIREA—Russian Technological University, Moscow, Russia, in 2010 and 2014, respectively, and the Habilitation degree, in 2021. In 2022, he became a Full Professor. He is currently a Full Professor with MIREA—Russian Technological University. During his career, he took part in a wide range of scientific and industrial projects. He has coauthored more than 80 articles and patents. His current research interests include motion control, robotics, energy, real-time communication, and FPGA design.