RESEARCH ARTICLE

Obstacle-Aware Navigation of Soft Growing Robots via Deep Reinforcement Learning

HAITHAM EL-HUSSIENY[1] **AND IBRAHIM A. HAMEED**[2], **(Senior Member, IEEE)**

[1]School of Innovative Design Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt
[2]Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 7034 Trondheim, Norway

Corresponding authors: Haitham El-Hussieny (haitham.elhussieny@ejust.edu.eg) and Ibrahim A. Hameed (ibib@ntnu.no)

ABSTRACT Soft growing robots, are a type of robots that are designed to move and adapt to their environment in a similar way to how plants grow and move with potential applications where they could be used to navigate through tight spaces, dangerous terrain, and hard-to-reach areas. This research explores the application of deep reinforcement Q-learning algorithm for facilitating the navigation of the soft growing robots in cluttered environments. The proposed algorithm utilizes the flexibility of the soft robot to adapt and incorporate the interaction between the robot and the environment into the decision-making process. Results from simulations show that the proposed algorithm improves the soft robot's ability to navigate effectively and efficiently in confined spaces. This study presents a promising approach to addressing the challenges faced by growing robots in particular and soft robots general in planning obstacle-aware paths in real-world scenarios.

INDEX TERMS Soft robotics, growing robots, reinforcement learning, deep Q learning.

I. INTRODUCTION

The exploration of confined spaces, such as those encountered in Minimally Invasive Surgeries (MIS) or the inspection of archaeological sites, presents significant challenges for traditional rigid robot designs [1], [2]. Consequently, there is a critical need for innovative materials and locomotion systems in robotics to navigate these challenging environments effectively. Drawing inspiration from biological systems like elephant trunks, octopus tentacles, and snakes, the development of soft continuum robots featuring continuous bending backbones has facilitated non-destructive navigation in congested spaces [3], [4], [5]. However, the limited lengths of traditional continuum robots restrict their capacity to explore more distant spaces [6].

Addressing this limitation, the concept of growth mobility, inspired by the way plants grow, has emerged as a groundbreaking approach in robotics. Growing robots, emulating the biological growth of plants, can extend their lengths, volumes, or knowledge, gradually adapting to their environment [7]. These robots, made of soft materials or

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

equipped with flexible joints, can extend to reach faraway spaces while maintaining compliance in confined settings. Prior efforts in designing long, flexible robots suitable for tight spaces include the ?Active Hose? by Tsukagoshi et al., a multi-degree-of-freedom robot designed for search and rescue applications [8]. Isaki et al. developed a flexible, extended cable with a camera for exploration in narrow areas [9], and the Slime Scope, a pneumatically driven soft arm, was created for use in rubble environments [10]. However, these designs often require moving the entire robot body, leading to significant friction with the environment.

Recent advancements in plant-inspired robotics have led to the development of two novel types of growing robots capable of extending their body lengths by adding materials at their tips. Sadeghi et al. proposed a plant root-like robot with a 3D printer-like head at its tip, enabling efficient steering by varying material deposition speeds [11], [12]. However, the growth speed is environment-dependent. In contrast, Hawkes et al. developed a vine-like growing robot using a "tip-eversion" mechanism [13], [14]. This pneumatically-driven robot, made from thin-walled polyethylene tubing, can extend for tens of meters and navigate slippery or sticky environments. Steering is achieved through environmental

interaction or by using Series Pneumatic Artificial Muscles (SPAM) for controlled movements [4], [15]. The unique combination of lengthening capacity, length-to-diameter ratio, and a compliant body allows vine-like robots to effectively navigate through distant, cluttered environments, as demonstrated in [16] and [17].

The very features that make vine-like robots so versatile also introduce complex challenges in motion planning. One of the critical aspects of vine robots is their irreversible growth process. Once a segment of the robot has extended or turned in a particular direction, retracting or reversing this action is not feasible. This irreversible nature of growth necessitates highly accurate and forward-thinking motion planning. In their recent work, [18] have pioneered the use of a Model Predictive Control (MPC) approach in the context of vine robot navigation, incorporating the kinematics of a vine robot as the predictive model, enabling advanced motion planning and obstacle avoidance. However, the development of an accurate kinematic model for vine-growing robots poses significant challenges.

This research introduces a new application of deep reinforcement learning, specifically Deep Q Networks (DQN), to enhance the navigational capabilities of soft growing robots, as depicted in Figure 1. Our algorithm capitalizes on the inherent flexibility of soft robots, seamlessly integrating environmental interactions into the decision-making process. This approach is particularly geared towards enabling navigation through densely cluttered spaces, a task that presents significant challenges for conventional robotic systems.
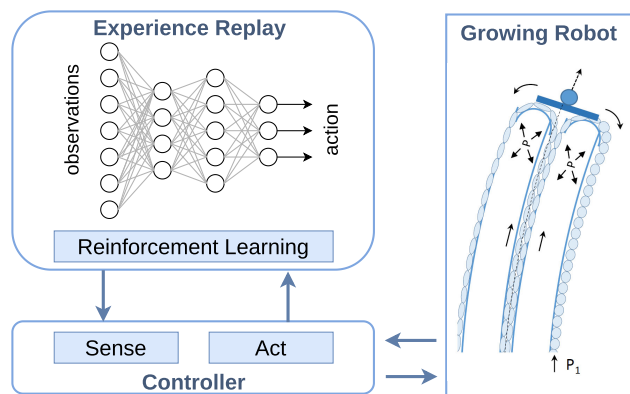
lays the groundwork for their enhanced utilization in diverse, real-world applications.

The paper is organized as follows: Section II introduces the kinematic model of the vine growing robot, including the interaction model with obstacles. In Section III details the proposed DQN reinforcement learning algorithm, including the observation, action and reward definitions. In Section IV, we present the simulation results to asses the performance of our DQN approach for planning safe paths for growing robots. Finally, Section V concludes the paper, summarizes our main contributions, and outlines potential directions for future work.

II. MODELING OF VINE ROBOTS

In this research, we focus on the vine-growing robot recently developed by [16]. Using an eversion mechanism [19], the robot can elongate its tip up to tens of meters. The robot's body is made of a thin-walled polyethylene tube, initially inverted, as depicted in Figure 2. The robot's length can be increased by applying air pressure to its chamber, which permits the tip to move away from its base. The eversion mechanism allows the vine robot to navigate easily through adhesive environments without getting stuck. To control the robot's bending, three serial Pneumatic Actuator Muscles (sPAM) are installed around its circumference, and air pressure is applied to these actuators. The arrangement of these sPAM actuators enables 360° steering, achieved by manipulating the input pressure for each actuator, as described in [16].
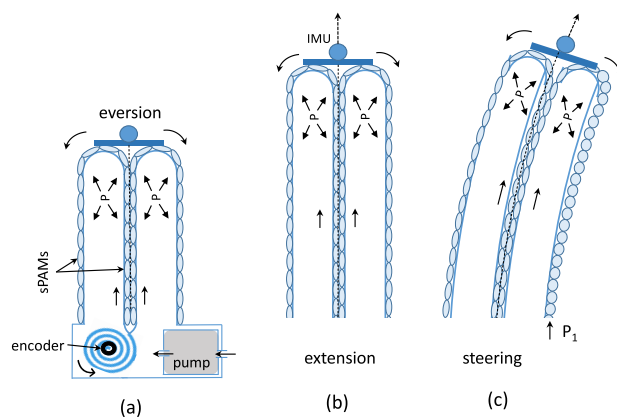

FIGURE 1. Enhancement of Movement Skills in the Vine-Growing Robot Through Deep Reinforcement Learning: This process involves the robot's adaptive learning from its interactions with the environment.

The findings from our simulations indicate a marked improvement in the ability of soft robots to exploit obstacles effectively to reach challenging goals, leveraging the DQN framework. This advancement addresses a critical obstacle in soft robotics, significantly enhancing the operational performance of soft growing robots in real-life scenarios, especially those characterized by complex, obstacle-rich environments. By equipping these robots with the capability to plan and execute obstacle-aware paths more effectively, this research

FIGURE 2. Operational Mechanics of the Growing Vine Robot. (a) The application of air pressure to the robot's central tube aids in extending the tip, as shown in (b). The steering mechanism, depicted in (c), is achieved by altering the air pressure in one or more of the soft pneumatic artificial muscles (sPAMs) surrounding the vine robot.

This section is dedicated to exploring the connection between actuation parameters and the position of the robot's tip, as delineated by the kinematic model. Additionally, it will illustrate the method of integrating the robot body's interaction with obstacles into the kinematics. This integration will be achieved by resolving a nonlinear constrained optimization problem that symbolizes this interaction.

A. KINEMATICS OF VINE ROBOTS

Given the inherent characteristics of soft growing robots, particularly their lightweight construction and the typically slow-paced nature of their operational movements, the primary focus in research and application tends to be on statics and kinematics, rather than dynamics. The forward kinematics of the vine robot are formulated predominantly under the guiding principle of the constant-curvature assumption, as detailed in the seminal work of Jones and Walker [20]. This robot is conceptualized as a single-section, extensible entity exhibiting continuum-like characteristics, endowed with dual degrees of freedom for curvature and bending, along with an additional degree for axial extension.

The spatial orientation and positioning of the robot, denoted as $\mathbf{T}_r^b$, are intrinsically linked to its configurational state, represented by $q \in \mathbb{R}^3$. This state vector q encompasses the pivotal elements defining the robot's structure: the length of the robot (s), the curvature (κ), and the angle of the curvature plane (ϕ), all of which are comprehensively illustrated in Figure 3. The pose $\mathbf{T}_r^b$ is derived from these parameters, offering a precise mathematical representation of the robot?s positioning and orientation in space.

$$
\mathbf{A} = \begin{bmatrix}
\cos^2\phi(\cos\kappa s - 1) + 1 & \sin\phi\cos\phi(\cos\kappa s - 1) \\
\sin\phi\cos\phi(\cos\kappa s - 1) & \cos^2\phi(1 - \cos\kappa s) + \cos\kappa s \\
\cos\phi\sin\kappa s & \sin\phi\sin\kappa s \\
0 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
-\cos\phi\sin\kappa s & \frac{\cos\phi(\cos\kappa s - 1)}{\kappa} \\
-\sin\phi\sin\kappa s & \frac{\sin\phi(\cos\kappa s - 1)}{\kappa} \\
\cos\kappa s & \frac{\sin\kappa s}{\kappa} \\
0 & 1
\end{bmatrix}.
$$
(1)

The robot's tip position $p = [x, y, z]^T \in \mathbb{R}^3$ in Cartesian space can be stated from Eq. (1) as,

$$
x = \frac{\cos\phi(\cos\kappa s - 1)}{\kappa},
$$
$$
y = \frac{\sin\phi(\cos\kappa s - 1)}{\kappa},
$$
$$
z = \frac{\sin\kappa s}{\kappa}
$$
(2)

In the context of this research, we have adopted the simplifying assumption of a planar environment, denoted mathematically as $\phi = 0$. This assumption is primarily for the sake of analytical and computational simplicity, allowing us to focus on the core aspects of the algorithm and control strategy without the additional complexity introduced by a three-dimensional environment. However, it is important to note that the extension of our model and methodologies to a spatial, three-dimensional environment is straightforward.

The kinematic formulation encapsulated in Eq. (2) is key for understanding the vine robot's intrinsic behavior and practical application in autonomous navigation. This model is integral to the training regimen of the Deep Q-Network (DQN) agent, a critical component in enabling
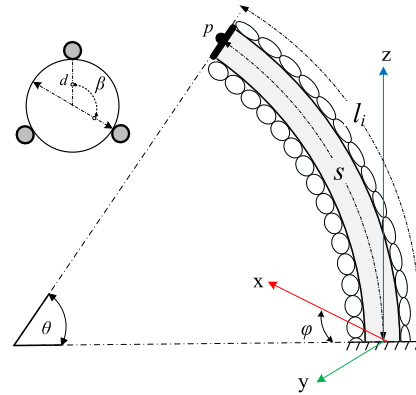

FIGURE 3. Schematic of vine-like growing robot and its configuration parameters. The robot is characterized by its length s, curvature $\kappa = \frac{\theta}{s}$, and angle of robot plane ϕ.

the vine robot to accurately navigate towards designated target positions, particularly in scenarios involving interaction with its surrounding environment.

Central to the operational mechanics of the vine robot is the actuation lengths, denoted by $l = [s, l_1, l_2, l_3]$. These lengths represent the tangible actuation space, encompassing both the core robot length and the lengths of the soft Pneumatic Artificial Muscles (sPAMs). Notably, utilizing shape parameters within this framework does more than merely dictate the robot's movements; it effectively generalizes the control problem. This generalization ensures that the control strategies developed are not limited to this specific vine robot but apply to a broader spectrum of continuum-like robots adhering to the constant-curvature model previously discussed.

B. INTERACTION MODELING

When a vine robot makes contact with an obstacle, it adapts its form to accommodate this interaction. Predicting and understanding this shape adaptation is essential for refining control algorithms and ensuring smooth, efficient navigation.

In order to integrate the influence of obstacles into the change of the robot's shape, we implemented the strain energy minimization principle as emphasized in [21]. In general, the approach involves treating a single-section continuum robot of length s as a flexible beam formed of N sub-segments. Each segment is subjected independently to bending deformation caused by the external torque resulting the actuation and the interaction with the environment as depicted in Figure 4. Consequently, the robot assumes a configuration that optimizes its strain energy at a state of static equilibrium. This optimization leads to a curvature alignment denoted as $\kappa \in \mathbb{R}^N$ across the segmented length of the vine robot's body, divided into N sections. This alignment serves as the central objective, encapsulated by the following key objective function with the aim of finding the set of curvatures $\kappa \in \mathbb{R}^N$ for the sub-segments.

$$
\arg\min_\kappa \left(\gamma_1 \sum_{i=0}^{N-1} (\kappa_i - \kappa_t)^2 + \gamma_2 \sum_{i=0}^{N-2} (\kappa_{i+1} - \kappa_i)^2 \right)
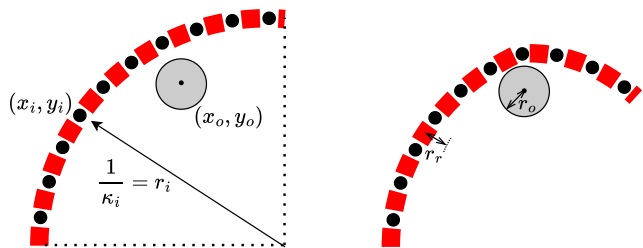$$
(3)

FIGURE 4. Effect of a point obstacle on the continuum robot shape. On the left the robot has no interaction with the obstacle and the robot curvature is the same to all segments. On the right, the robot has collided with a point obstacle making the robot curvature not consistent along it's backbone.

trim=5cm 4cm 9cm 1cm,clip In the specified equation, the formulation is strategically divided into two principal terms, each addressing a distinct aspect of the vine robot's curvature control. The first term is designed to minimize the deviation between the curvature of each sub-segment of the robot and a predefined target curvature, κ. This target curvature is established under the assumption of obstacle-free conditions, serving as a benchmark for the desired trajectory of the robot in an unobstructed environment.

Simultaneously, the second term in the equation is tasked with a different objective: it aims to reduce the disparity in curvature between adjacent segments of the robot. This is a critical consideration, as it adheres to the principle of minimizing strain energy, thereby preserving the integrity of the robot's continuous backbone structure. The necessity to maintain a smooth transition between segments is paramount in ensuring the robot's efficient and effective movement, particularly in scenarios involving complex navigation paths.

The incorporation of the factors γ_1 and γ_2 in this equation plays a vital role. These weighting factors are employed to establish a balance between the dual objectives of the equation.

To avoid the growing robot from colliding with the obstacle, the optimization problem in Eq. (3) is subjected to the constraint that the Cartesian coordinates (x_i, y_i) of the tip of each sub segment i is within a safe distance r_r from the center of the obstacle at the location (x_o, y_o) with radius r_o as follow, assuming the obstacle location is known

$$d_{io} \geq r_o + r_r, \quad \forall i = 0, 1, \dots, N-1 \tag{4}$$

where $d_{io} = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}$ is the Euclidean distance from the segment i of the robot and the obstacle location.

While the constant curvature model serves as the foundational assumption in our kinematic modeling of the vine robot, it's important to recognize that interactions with environmental obstacles can significantly alter the robot's shape, thereby challenging this assumption. In scenarios where the robot comes into contact with an obstacle, its behavior diverges from that of a single-segment, constant curvature entity. Instead, the robot begins to exhibit characteristics akin to a multi-segmented structure, displaying varying curvatures along its length.

III. DQN REINFORCEMENT LEARNING ALGORITHM

This section elaborates on the Reinforcement Learning (RL) algorithm employed to train the growing robot. The primary objective is to enable the robot to efficiently navigate towards a specified goal within its environment, while skillfully navigating around and interacting with obstacles. The RL agent operates by observing the current state $\boldsymbol{x} \in \mathbb{R}^9$ of the growing robot, which is defined within the observation space as:

$$\boldsymbol{x} = [s \ \kappa \ \dot{s} \ \dot{\kappa} \ x_g \ y_g \ x_o \ y_o \ d]^T \tag{5}$$

In this state vector $\boldsymbol{x}$, s and κ represent the robot's length and curvature, respectively, and $\dot{s}$ and $\dot{\kappa}$ are the corresponding time derivatives of these quantities. These variables are crucial for the RL agent to understand and predict the robot's physical configuration and movement dynamics. To maintain awareness of the goal and obstacle locations within the environment, the coordinates (x_g, y_g) for the goal and (x_o, y_o) for the obstacle are included in the observation space. Additionally, the Euclidean distance d is calculated and tracked. This distance, defined between the robot's tip (x, y) as derived in Eq. (2) and the goal position, is given by:

$$d = \sqrt{(x - x_g)^2 + (y - y_g)^2} \tag{6}$$

In order to ensure the generalizability of the RL model to different goal locations, the position of the goal is varied randomly during the training process, with a probability of 0.2 for changing in each training iteration. This randomization strategy is designed to expose the growing robot to a diverse set of scenarios, thereby enhancing its ability to adapt and perform effectively in a wide range of environmental conditions and goal configurations.

The reinforcement learning agent in this study is designed to manage the movement of a growing robot by sampling an action vector $\boldsymbol{a}$, defined as follows:

$$\boldsymbol{a} = [\dot{s} \ \dot{\kappa}] \tag{7}$$

Here, $\dot{s}$ represents the robot's growth speed, and $\dot{\kappa}$ indicates the rate of change in the robot's curvature. By incorporating these specific actuation variables, the proposed algorithm demonstrates versatility and adaptability across a broad range of continuum-like robotic systems. This flexibility is crucial as it ensures the applicability of the goal-reaching algorithms irrespective of the underlying actuation mechanism, whether it be cable-driven or pneumatically driven systems.

In the domain of Q-learning, a crucial aspect is the representation of actuation in discrete values. This discretization ensures that each actuation component of the robot operates within a predefined set of actions, effectively covering the entire operational range. Specifically, in our model:

1) The rate of change of curvature, $\dot{\kappa}$, is restricted to three discrete values, $[\kappa_l \ 0 \ \kappa_r]$ rad/s. These values correspond to the robot bending to the left, remaining stationary (no bending), and bending to the right, respectively.

2) The growth speed, $\dot{s}$, is either set to elongate the robot at a constant rate of s_m m/s or to halt its elongation, denoted as $\dot{s} = 0$.

Consequently, the action space for the robot is defined as a set of all possible combinations of these discrete values:

$$a = \left\{ \begin{bmatrix} \kappa_l \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_l \\ s_m \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ s_m \end{bmatrix}, \begin{bmatrix} \kappa_r \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_r \\ s_m \end{bmatrix} \right\}$$

This discrete action space encapsulates all the possible movements of the robot within its operational constraints, allowing for a structured approach in the Q-learning algorithm to optimize the robot's behavior for achieving specific goals. At the onset of each training episode, the RL algorithm initializes with predetermined parameters κ_0 and s_0. The algorithm initially adopts a strategy of exploration, where actions a are chosen randomly based on an exploration probability ϵ, as delineated in Algorithm 1. This exploratory phase is crucial for the agent to acquire diverse experiences and insights into the environment's dynamics.

As training progresses, the algorithm gradually shifts its focus towards exploitation, increasingly selecting actions that are anticipated to yield the highest cumulative reward. The formulation for calculating the cumulative reward R_k at each step k is expressed as:

$$R_k = r_k + \gamma r_{k+1} + \gamma^2 r_{k+2} + \ldots + \gamma^{n-k} r_n = r_k + R_{k+1} \tag{8}$$

In this equation, r_k represents the immediate reward assigned at each step k, and γ is the discount factor, which balances the importance of immediate and future rewards.

The computation of the immediate reward r_k in our proposed method is designed to incentivize goal-oriented behavior. Specifically, a high reward of $+1000$ is assigned if the robot's tip is within a maximum distance of $d_{\max} = 0.1$ from the goal, leading to the termination of the training episode. Conversely, if the robot's tip is farther from the goal, the reward is inversely proportional to this distance d, as defined in Eq. (6). The reward function is thus formulated as:

$$r_k = 1 - \left(\frac{\ln(1 + d_k)}{\ln(1 + d_{\max})} \right) \tag{9}$$

This reward structure ensures that the agent is motivated to minimize the distance to the goal, effectively guiding its learning process towards efficient and goal-directed actions.

During both the training and testing phases of our approach, the position of the obstacle is assumed to be fixed. This static positioning of the obstacle simplifies the learning and evaluation process, allowing the reinforcement learning agent to focus on understanding and adapting to a consistent environmental factor. However, it's noteworthy that our approach possesses the inherent capability to handle dynamic obstacles as well. This adaptability stems from the inclusion of the goal position in the agent's observation space. Since the agent is already equipped to process and respond
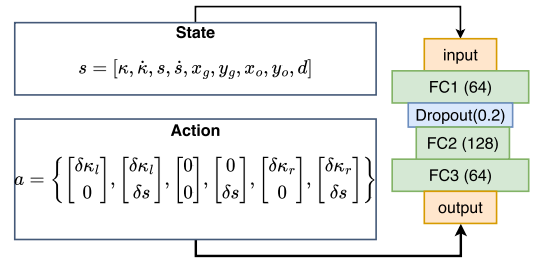

FIGURE 5. Illustration of the DQN Agent Architecture: This diagram depicts the agent as it observes state s and executes action a. The process of transforming the observed state into an actionable decision is facilitated through a feed-forward neural network, which serves as the core mechanism for mapping observations to corresponding actions.

Algorithm 1 Action Sampling

$P \leftarrow P_0$
for episode = 1 to N **do**
    for $k = 1$ to M **do**
        with probability ϵ select random action a_k, otherwise $a_k = \max_a Q(s, a)$
        $(s_{k+1}, r_k) = \mathbf{step}(s_k, a_k)$ ▷ Simulate the system
        $P_k = (s_k, a_k, s_{k+1}, r_k)$

Algorithm 2 Learning Q-Value

$\theta \leftarrow \theta_0$ ▷ Initialize network weights
while No termination signal recieved **do**
    Sample random minibatch of (s, a, r, s') from P.
    Set target $= r + \gamma \max_{a'} Q_\theta(s', a')$
    Set prediction $= Q_\theta(s', a')$
    $L = \frac{1}{2} \left[\text{target} - \text{prediction} \right]^2$
    $\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial \theta}$ ▷ Update network weights

---

to changes in the goal's location, extending this adaptability to accommodate moving obstacles is a feasible and logical progression.

The deep Q-learning algorithm [22] is utilized in this research to train the agent to reach the desired goal while maximizing the accumulative reward of Eq. (8). The RL agent learns to select the best possible action a at every state s as described in Algorithm 2 and Figure 5. This is represented by a policy $\pi(s)$

$$\pi(s) = \max_a Q(s, a) \tag{10}$$

where $Q(s, a)$ is approximated in Q-learning algorithm as

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right) \tag{11}$$

where s' is the next state while a' is the next action. Meanwhile, α is the learning rate.

In conventional Q-learning [23], where the state space is small, the Q value is represented as a table of state and the corresponding Q of each action in that state. However, in this

navigation problem, the state space is enormous. Thus, a deep neural network is used to approximate the Q-table as $Q_\theta(s, a)$, where θ is the deep neural network that is trained as follows:

$$\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial \theta} \qquad (12)$$

where L is the loss:

$$
\begin{aligned}
L &= \frac{1}{2} \left[\text{target} - \text{prediction} \right]^2 \\
&= \frac{1}{2} \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]^2 \qquad (13)
\end{aligned}
$$

IV. RESULTS AND DISCUSSION

A comprehensive series of numerical experiments designed to evaluate the effectiveness of the DQN-based method for obstacle-aware navigation in growing robots. Initially, the focus was on analyzing the interaction model. This model crucially predicts how the morphology of the vine robot alters in response to encounters with obstacles. In Figure 6, we illustrate the initial phase of the robot's navigation sequence, wherein it begins in a configuration that is free from any collision with the obstacle, depicted in transparent red. Subsequently, as the robot navigates through its environment, there is an incremental increase in its bending motion, applying constant $\dot{\kappa}$. This progression continues until the point of collision with the obstacle. Upon this interaction, the robot's shape undergoes a transformation, conforming to the parameters set out by the interaction model discussed in Section II. Such an analysis is vital for understanding the adaptive behavior of the robot under varying environmental constraints.
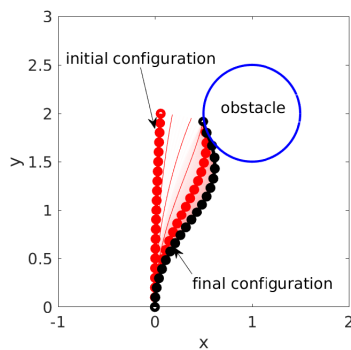

FIGURE 6. The vine growing robot interacting with an obstacle after applying constant actuation of $\dot{\kappa}$.

In this study, we configured the replay memory to accommodate a maximum of 20,000 steps, as delineated in Algorithm 1. This memory utilizes a First-In-First-Out (FIFO) strategy for data replacement, ensuring a continual update of experiences for the learning agent. The training protocol, detailed in Algorithm 2, comprised an extensive series of 200,000 episodes (N), with each episode constrained to a maximum of 100 steps ($k_{\max}$). This comprehensive approach is designed to provide the agent with a wide array of scenarios for robust training.

The learning thread, initiating after a prelude of 20 episodes, facilitates the accumulation of initial experiences in the replay memory, essential for effective learning. A batch size of 64 was chosen for the random selection of experiences from the memory pool (denoted as P), balancing the computational efficiency and diversity in experience sampling.

The exploration rate of the network was initially set to 1.0, promoting a purely exploratory approach at the onset of the learning phase. Subsequently, this rate was decremented by 0.05 following each episode, culminating at zero. Such a gradual decrement facilitates a strategic shift from exploration to exploitation, enabling the agent to progressively rely on learned behaviors over random actions.

For optimization, the ADAMS optimizer [24] was employed, renowned for its efficacy in handling large datasets and complex variable interactions. Finally, the discount factor and learning rate were set at 0.5 and 0.1, respectively.

A. FIXED GOAL, OBSTACLE-FREE

In the initial phase, the training process of the proposed DQN-based navigation system was evaluated in a goal-reaching scenario. This scenario was characterized by a fixed goal position set in an environment free of obstacles. Figure 7 presents the learning progression when the goal is set at coordinates $(1, 3)$, with the vine robot starting from initial parameters $s_0 = 1$ meter and $\kappa_0 = 0.01$ m^{-1}.
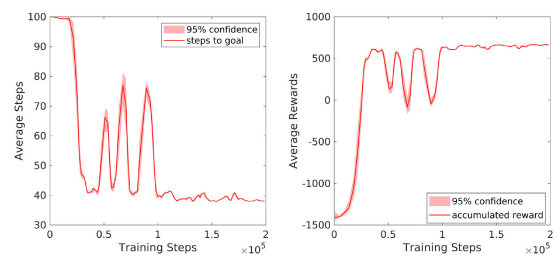

FIGURE 7. Evaluation of the Proposed DQN Goal-Seeking Algorithm: This figure illustrates the learning performance of the algorithm, showcasing two key metrics. On the left, the graph displays the number of steps required by the agent to reach the goal. On the right, the graph depicts the accumulated reward achieved by the agent.

The left side of Figure 7 illustrates a notable decrease in the number of steps required for the robot to successfully complete the reaching phase. This reduction indicates an improvement in the efficiency of the robot's path-finding as the training progresses. On the right side, the figure depicts an increase in the average accumulated reward over the course of the training steps.

It is important to note the observable fluctuations in both the number of steps and the average reward, particularly in the early stages of training. These fluctuations can be attributed to the high exploration rate ϵ initially set in the training process. At this stage, the robot is more inclined to make random decisions, exploring a broader range of policies. This exploratory approach is crucial for the robot to discover and

learn from diverse situations, even though it may temporarily deviate from the policy that maximizes expected rewards. As training progresses and the exploration rate decreases, the robot increasingly adopts strategies that focus on reward maximization, leading to more consistent and goal-oriented behavior.

Figure 8 evaluates the DQN algorithm post-training, focusing on its ability to direct the robot towards a specified goal using the learned DQN model. The upper section of the figure graphically represents the decrease in error distance, which is the spatial discrepancy between the robot's tip and the designated goal. The lower section of the figure depicts the discrete control actions executed by the robot during its journey to the goal. These actions are a direct result of the decisions made by the DQN agent based on the trained model. A key observation from the results is that after 38 steps, equivalent to 3.8 seconds, the robot successfully reaches the goal, where the DQN agent applying zero control actions.
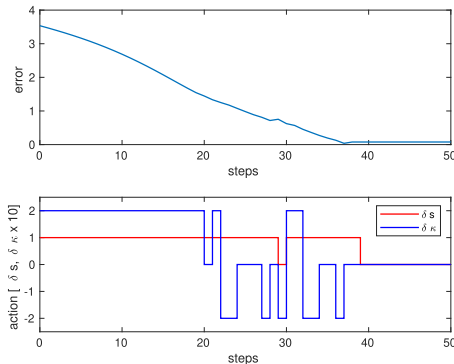

FIGURE 8. (Top) The decay of the error between the robot's tip and the target location of the trained DQN. **(Bottom)** The set of discrete action commands executed by the vine robot to reach the goal.

B. VARYING GOAL, OBSTACLE-FREE

To assess the generalization capabilities of the proposed DQN model, particularly in navigating towards a goal with varying location within the environment, we incorporated a goal-changing strategy during the training phase. This strategy involved altering the goal's location with a predefined probability of 10%, thus presenting the agent with diverse navigation challenges. The environment was kept free of obstacles for this phase of training to simplify the evaluation process and focus on the agent's ability to adapt to changing goals.

Figure 9 illustrates the training progress of the DQN agent under this variable goal location condition. Notably, the figure demonstrates a decreasing trend in the number of steps required for the agent to reach the goals as the episodes progress. Concurrently, there is an observable increase in the accumulated rewards over the episodes, indicating an improvement in the agent's efficiency and goal-reaching capability.
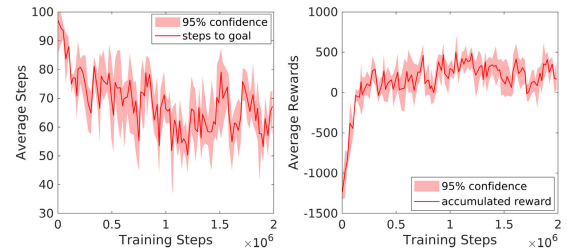

FIGURE 9. The learning performance of the DQN algorithm in an obstacle-free goal changing environment.
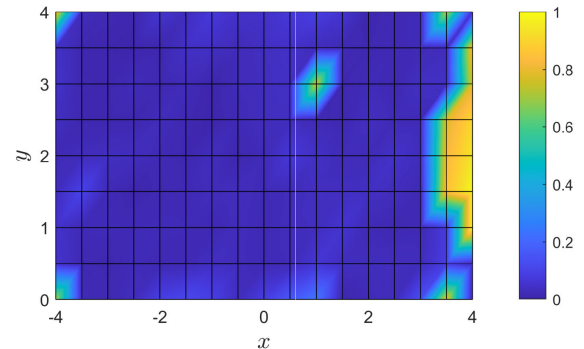

FIGURE 10. Normalized error distribution among the working environment of the growing robot.
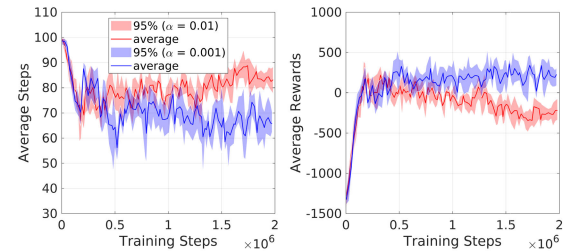

FIGURE 11. The training progress in terms of the number of steps and the accumulated rewards of the DQN agent for commanding the vine robot to reach a varying goal while considering the interaction with obstacles.
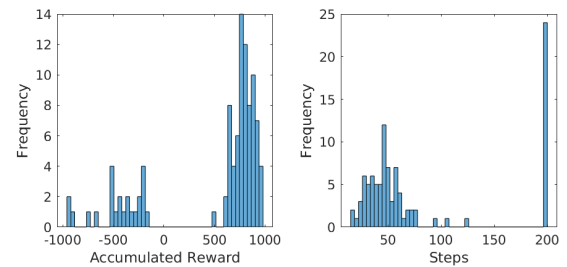

FIGURE 12. The histogram of the DQN agent performance in terms of the number of steps and the accumulated rewards in the varying goal obstacle-aware scenario.

The observed fluctuations in both the number of steps and the accumulated rewards can be attributed to the dynamic nature of the goal locations. Despite these variations in goal locations, the DQN model exhibits a notable robustness over the episodes. During the testing phase, the DQN agent was

tasked with directing the vine robot to reach various goals, with the goal locations (x_o, y_o) sequentially altered within the environment, adhering to the ranges $x_o \in [0, 4]$ and $y_o \in [-4, 4]$. Figure 10 presents a heat map illustrating the normalized error distribution across the working environment. The visualization indicates minimal errors in goal-reaching accuracy across diverse goal locations, with the exception of areas near the edges. This observation of increased error at the periphery could potentially be attributed to the constrained size of the simulated vine robot, which was limited to five meters.

C. VARYING GOAL, OBSTACLE-AWARE

In the final series of experiments, the DQN agent underwent extensive training, consisting of 2 million steps, within an environment that included a fixed obstacle with a radius of $r_o = 0.5$ meters, located at coordinates $(1, 2)$. Concurrently, the goal's position was dynamically altered throughout the training, with a 10% probability of being randomly relocated within the environmental boundaries. Figure 11 illustrates the agent's progressive learning curve, showcasing an efficient reduction in the number of steps required to reach the goal while simultaneously maximizing the accumulated rewards.

This training efficacy was further analyzed under two distinct learning rate scenarios, $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$, as demonstrated in Figure 11. The comparative analysis suggests that a lower learning rate tends to enhance the training performance, potentially leading to a more precise and refined learning outcome. However, it's important to note that this improved performance might come at the cost of increased computational resources. This trade-off between learning rate and computational demand is a crucial consideration in the optimization of the DQN training process, particularly in complex environments where obstacle navigation and dynamic goal-seeking are key factors.

During the testing phase, the proficiency of the DQN agent was rigorously evaluated through a series of 100 trials, wherein it was tasked to navigate to randomly set goals within a maximum limit of 200 time steps. The performance outcomes of these trials are comprehensively visualized in the histograms presented in Figure 12, focusing on two key metrics: the number of steps taken and the accumulated rewards.

The data illustrates that in approximately 75% of the trials, the robot successfully accrued positive rewards ranging between 500 and 1000. This result indicates a high level of efficiency in reaching the goals. Additionally, it was observed that the majority of these successful trials were completed within 50 steps, further underscoring the agent's effectiveness.

However, it is noteworthy that in about 25% of the trials, the robot failed to reach the goal within the stipulated 200 steps. This failure could be attributed to a couple of factors: Firstly, the random goal locations might occasionally be generated within the obstacle's space, rendering them physically inaccessible. Secondly, the goals may be positioned at distant locations in the environment, posing a challenge for the robot to reach them within the 200-step limit.

To mitigate this issue and decrease the failure rate, a couple of strategies could be implemented: One approach would be to modify the goal generation algorithm to ensure that goals are not placed within or too close to the obstacle's location. Alternatively, extending the time limit beyond 200 steps could provide the robot with additional leeway to reach more distantly placed goals. Either of these adjustments could enhance the overall success rate of the DQN agent in effectively reaching the goals across a variety of environmental conditions.

Figure 13 depicts a specific testing scenario for the DQN agent, illustrating a strategic navigation task involving the vine robot, marked with red indicators, and an obstacle, represented by a blue circle. This scenario demonstrates the robot's ability to exploit the presence of the obstacle in
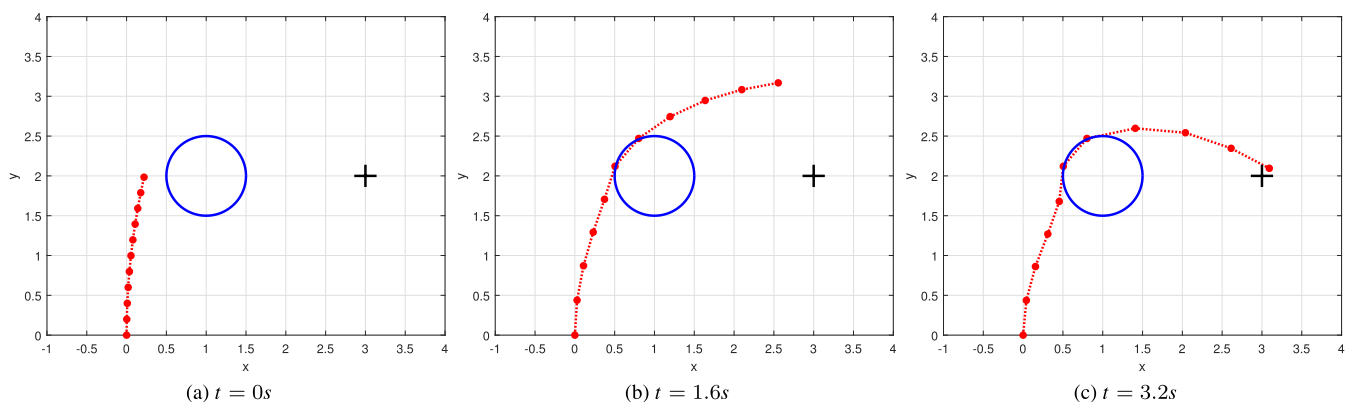

(a) $t = 0s$　　　　　　　　　　(b) $t = 1.6s$　　　　　　　　　　(c) $t = 3.2s$

FIGURE 13. The trained DQN agent successfully commands the vine growing robot to reach the goal (represented in a black cross) while considering the interaction with the obstacle located at $(1, 2)$.

order to reach a goal that is otherwise challenging to access. The figure sequentially portrays the robot's movement across different time steps, highlighting its adaptive navigation strategy.

A key observation from this scenario is the alteration in the robot's shape upon encountering the obstacle, deviating from the constant curvature model initially assumed. This deviation is not just a consequence of the robot's interaction with the obstacle but is also strategically advantageous. The vine robot's inherent compliance, a fundamental characteristic of soft robotics, is effectively utilized here to navigate around the obstacle. This flexibility allows the robot to mold its path and shape in response to environmental constraints, showcasing an advanced level of adaptability.

The scenario illustrated in Figure 13 exemplifies the practical benefits of the robot's compliance in real-world applications. It demonstrates the robot's capacity to dynamically adjust its form and trajectory in complex environments, leveraging obstacles not as hindrances, but as aids in navigating towards difficult-to-reach goals. This ability to intelligently interact with and utilize environmental features underscores the sophistication and potential of the DQN-guided navigation strategy in soft robotics.

V. CONCLUSION

In this research a Deep Q Network (DQN)-based navigation approach is implemented for a soft vine-growing robot resulting in significant insights into the capabilities and adaptability of this technique in environments with obstacles. Through a series of meticulously designed experiments, the study has demonstrated the proficiency of the trained DQN agent in guiding the robot to effectively reach designated goals, even in scenarios with challenging obstacle placements.

Particularly notable is the robot's ability to navigate around and utilize obstacles as part of its pathfinding strategy. This scenario illustrated the robot's sophisticated interaction with the environment, where it skillfully adapted its shape in response to obstacles, deviating from the constant curvature model. This adaptability, facilitated by the robot's inherent compliance, proved to be advantageous in navigating through and leveraging the environment to reach difficult goals.

The methodology implemented in this study adopts a discrete set of actions to accomplish the navigation task of the vine growing robot. This approach is characterized by its simplicity and ease of implementation, as it allows the robot to choose from a predefined set of actions at each step. However, this discrete action space may limit the robot's ability to perform fine-grained movements, which can be crucial in tasks requiring high precision.

As a natural progression of this research, it is essential to explore and analyze how this discrete action space approach compares with continuous space learning techniques. Continuous action spaces, as employed in algorithms like Asynchronous Advantage Actor-Critic (A3C) [25] and Deep Deterministic Policy Gradient (DDPG) [26], offer a broader range of potential actions at each step. This can lead to more nuanced and precise control strategies, potentially enhancing the robot's performance in complex tasks.

REFERENCES

[1] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robot. Autom. Mag.*, vol. 24, no. 2, pp. 42–53, Jun. 2017.

[2] G. Brantner and O. Khatib, "Controlling ocean one: Human–robot collaboration for deep-sea manipulation," *J. Field Robot.*, vol. 38, no. 1, pp. 28–51, Jan. 2021.

[3] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[4] J. D. Greer, L. H. Blumenschein, A. M. Okamura, and E. W. Hawkes, "Obstacle-aided navigation of a soft growing robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[5] I. A. Seleem, S. F. M. Assal, H. Ishii, and H. El-Hussieny, "Demonstration-guided pose planning and tracking for multi-section continuum robots considering robot dynamics," *IEEE Access*, vol. 7, pp. 166690–166703, 2019.

[6] P. Liljeback, K. Y. Pettersen, Ö. Stavdahl, and J. T. Gravdahl, "A review on modeling, implementation, and control of snake robots," *Robot. Auto. Syst.*, vol. 60, no. 1, pp. 29–40, 2012.

[7] E. Del Dottore, A. Sadeghi, A. Mondini, V. Mattoli, and B. Mazzolai, "Toward growing robots: A historical evolution from cellular to plant-inspired robotics," *Frontiers Robot. AI*, vol. 5, p. 16, Mar. 2018.

[8] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2001, pp. 2454–2459.

[9] K. Isaki, A. Niitsuma, M. Konyo, F. Takemura, and S. Tadokoro, "Development of an active flexible cable by ciliary vibration drive for scope camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3946–3951.

[10] D. Mishima, T. Aoki, and S. Hirose, "Development of pneumatically controlled expandable arm for search in the environment with tight access," in *Field and Service Robotics*. Berlin, Germany: Springer, 2003, pp. 509–518.

[11] A. Sadeghi, A. Mondini, E. Del Dottore, V. Mattoli, L. Beccai, S. Taccola, C. Lucarotti, M. Totaro, and B. Mazzolai, "A plant-inspired robot with soft differential bending capabilities," *Bioinspiration Biomimetics*, vol. 12, no. 1, Dec. 2016, Art. no. 015001.

[12] A. Sadeghi, E. Del Dottore, A. Mondini, and B. Mazzolai, "Passive morphological adaptation for obstacle avoidance in a self-growing robot produced by additive manufacturing," *Soft Robot.*, vol. 7, no. 1, pp. 85–94, Feb. 2020.

[13] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Sci. Robot.*, vol. 2, no. 8, Jul. 2017, Art. no. eaan3028.

[14] L. H. Blumenschein, A. M. Okamura, and E. W. Hawkes, "Modeling of bioinspired apical extension in a soft robot," in *Proc. Conf. Biomimetic Biohybrid Syst.*, in Lecture Notes in Computer Science. Springer, 2017, pp. 522–531.

[15] H. El-Hussieny, U. Mehmood, Z. Mehdi, S.-G. Jeong, M. Usman, E. W. Hawkes, A. M. Okarnura, and J.-H. Ryu, "Development and evaluation of an intuitive flexible interface for teleoperating soft growing robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4995–5002.

[16] M. M. Coad, L. H. Blumenschein, S. Cutler, J. A. R. Zepeda, N. D. Naclerio, H. El-Hussieny, U. Mehmood, J.-H. Ryu, E. W. Hawkes, and A. M. Okamura, "Vine robots," *IEEE Robot. Autom. Mag.*, vol. 27, no. 3, pp. 120–132, Sep. 2020.

[17] E.-H. Haitham, S.-G. Jeong, and J.-H. Ryu, "Dynamic modeling of a class of soft growing robots using Euler–Lagrange formalism," in *Proc. Soc. Instrum. Control Eng. (SICE)*. IEEE, Society of Instrument and Control Engineers, 2019.

[18] H. El-Hussieny, I. A. Hameed, and A. B. Zaky, "Plant-inspired soft growing robots: A control approach using nonlinear model predictive techniques," *Appl. Sci.*, vol. 13, no. 4, p. 2601, Feb. 2023.

[19] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "A soft, steerable continuum robot that grows via tip extension," *Soft Robot.*, vol. 6, no. 1, pp. 95–108, Feb. 2019.

[20] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–55, Feb. 2006.

[21] K. P. Ashwin, S. K. Mahapatra, and A. Ghosal, "Profile and contact force estimation of cable-driven continuum robots in presence of obstacles," *Mechanism Mach. Theory*, vol. 164, Oct. 2021, Art. no. 104404.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[23] 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*.

[24] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[25] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Intelligent Autonomous Systems 13*. Berlin, Germany: Springer, 2016, pp. 335–348.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

IBRAHIM A. HAMEED (Senior Member, IEEE) received the first Ph.D. degree in industrial systems and information engineering from Korea University, Seoul, South Korea, and the second Ph.D. degree in mechanical engineering from Aarhus University, Aarhus, Denmark. He is currently a Professor and the Deputy Head of Research and Innovation with the Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science, and Technology (NTNU), Norway. His current research interests include artificial intelligence, machine learning, optimization, and robotics. He is the Elected Chair of the IEEE Computational Intelligence Society (CIS) Norway Section.

• • •

HAITHAM EL-HUSSIENY received the M.Sc. and Ph.D. degrees in mechatronics and robotics engineering from Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt, 2013 and 2016, respectively. He is currently an Associate Professor of artificial intelligence and robotics engineering with E-JUST. His research interests include soft robots, soft haptics, teleoperation, model predictive control, and applied intelligence.