IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 April 2024, accepted 23 May 2024, date of publication 29 May 2024, date of current version 10 June 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3406768*

RESEARCH ARTICLE

# Automatic Curriculum Determination for Deep Reinforcement Learning in Reconfigurable Robots

**ZOHAR KARNI[1], OR SIMHON[2], DAVID ZARROUK[2], (Member, IEEE), AND SIGAL BERMAN[1], (Senior Member, IEEE)**
[1]Department of Industrial Engineering and Management, Ben Gurion University of the Negev, Beer Sheva 8410501, Israel
[2]Department of Mechanical Engineering, Ben Gurion University of the Negev, Beer Sheva 8410501, Israel

Corresponding author: Sigal Berman (sigalbe@bgu.ac.il)

This work was supported in part by Helmsley Charitable Trust through the Agricultural, Biological, and Cognitive Robotics Center of Ben-Gurion University.

**ABSTRACT** Deep reinforcement learning (DRL) is a prevalent learning method in robotics. DRL is commonly applied in real-world scenarios, such as learning motion behavior in rough terrain. However, the lengthy learning epochs reduce DRL practicability in many such environments. Curriculum learning can significantly enhance the efficiency of DRL, but establishing a curriculum is challenging, partly because it can be difficult to assess the operation complexity for each task. Determining operation complexity can be especially difficult for reconfigurable search and rescue robots. We present a method for learning based on an automatically established curriculum tuned to the robot's perspective. The method is especially suitable for outdoor environments with multiple obstacle variants, e.g., environments encountered in search and rescue missions. After an initial learning stage, the behavior of a robot when overcoming each obstacle variant is characterized using Gaussian mixture models (GMMs). Hellinger's distance between the GMMs is computed and used to cluster the variants hierarchically. The curriculum is determined based on the formed clusters and the average success rate in each cluster. The method was implemented on RSTAR, a highly maneuverable and reconfigurable field robot that can overcome a variety of obstacles. Learning using the automatically determined curriculum was compared to learning without a curriculum in a simulation with three obstacle types: a narrow channel, a low entrance, and a step. The results show that learning using the automatically determined curriculum enables overcoming obstacles faster and with higher success rates than learning without a curriculum for all obstacles, especially for complex obstacle variants. The developed method offers a promising method for learning motion behavior in real-world scenarios.

**INDEX TERMS** Deep reinforcement learning, curriculum learning, reconfigurable robot, Hellinger's distance.

## I. INTRODUCTION

Deep reinforcement learning (DRL) is a prevalent learning method in robotics [1], [2], [3], [4], [5], [6]. The method is particularly suitable for robots operating in unstructured environments since, with reinforcement learning, the robot learns optimal behavior directly through interaction with the environment. This interaction obviates the need for explicitly detailing the solution, and the designer is required only to provide the means for assessing the robot's performance. The downside of the method, particularly for complex situations, is that DRL typically requires a lengthy interaction with the environment [6]. The complexity of real-world operations thus poses a significant challenge for such learning methods [7], and the need for a lengthy interaction limits the practicability of DRL in many real-world scenarios.

One way of addressing the complexity is via transfer learning. Transfer learning is used to expedite deep neural network

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

training [8], [9] and has been successfully used in DRL [10]. In transfer learning, previously learned network weights form a baseline from which the subsequent learning effort starts. Curriculum learning extends transfer learning by establishing a learning order (a curriculum) [11], [12], [13], [14], [15], [16]. Determining a curriculum, i.e., choosing the learning tasks and their order of presentation, is complex since an effective curriculum is tailored to the required operations, the environment, and the agent's capabilities. In addition, the order of presentation as training progresses must prevent the agent from forgetting behaviors previously learned [17].

Curricula for robots are based on a separation of learning tasks according to different criteria, e.g., the execution timeline, robot operations, or environment targets or tasks. In curricula arranged by timeline, the mission that the robot is required to learn is divided into sub-stages or sub-goals, arranged according to their execution timeline, and the robot gradually learns the overall mission either from start to finish or vice versa [18], [19], [20]. This type of curriculum may be automatically generated based on a generative adversarial network (GAN), where the GAN is trained to produce new goals with increasingly distant states from the target [13]. In curricula arranged by robot operations, the robot's mission is divided into distinct operations [12] via manual segmentation, interactive feedback, or prior knowledge about the mission and the operations. However, the above methods typically require human intervention and may perform sub-optimally in the case of erroneous underlying assumptions regarding the robots' capabilities [7]. Various methods have been developed for automatically deconstructing a complex mission into sub-operations to enhance reinforcement learning, e.g., by applying abstract high-level behavior sketches to learn composable deep sub-policies [21]. In curricula arranged by tasks in the environment, the presented tasks are divided into subsets. For example, for automatic curriculum generation for learning generative models from data points, the data points were divided into clusters. The clustered data was presented to a learning algorithm based on data point centrality within each cluster, i.e., from central, denser regions to the cluster boundary [22]. Since points on cluster boundaries are commonly related to outliers and noise, learning with the devised curriculum was more robust to the potentially harmful effects of outliers and noise. In an approach suitable for a more elaborated definition of tasks, task distributions are interpolated based on distribution similarity. The distribution is shifted in accordance with agent performance [23], [24]. Teacher-student and self-paced learning prescribe various methods for the selection of the following learning episode based on current previous learning scores [15], [16], [25]. In procedural content generation (PCG), each environment instance is a level. As in teacher-student and self-paced learning, prioritized level replay was suggested for selecting the next level based on estimated learning potential [26]. The current work suggests a new separation criterion, obstacle characteristics, vital for operations typically encountered in search and rescue scenarios.

Losses of human life caused by disasters like earthquakes and hurricanes have driven the development of robotic systems to assist rescuers. Search-and-rescue robots have attracted significant attention in recent years due to their participation in search operations following such disasters [27]. There is an obvious need for search-and-rescue robots to be fast, but there is an equally important need for them to be flexible due to the complicated and rough terrain in disaster areas [28]. Compared to fixed-design robots, reconfigurable mobile robots provide superior mobility and safety in irregular terrain. Mobile robots with active articulated elements are particularly suitable for moving on rough terrain. These elements facilitate configuration control according to environmental conditions and adapting the center of mass to the terrain [29]. Wheeled robots with internal articulated elements can adapt their configuration, reposition their center of mass, and influence the contact forces against the terrain [30]. Actively articulated wheel-on-leg robots can adapt to the terrain, control the forces on the wheels, move through narrow passages, and climb over complex obstacles [31]. A quadruped robot fitted with such mechanisms can even climb over obstacles that are much larger than itself [32]. DRL has been used for learning optimal motion behaviors for off-road robots, such as a 4 × 4 wheeled robot [33], a quadruped robot [32], a crawler robot [34], and the RSTAR (Rising Sprawl-Tuned Autonomous Robot), which is a four-wheel drive, reconfigurable, off-road quadruped search-and-rescue robot [35], [36] (Figure 1).

The RSTAR can be fitted with wheels, spoked legs, or their combination to improve stability and reduce energy consumption. The unique advantages of RSTAR lie in its ability to rotate its wheel axles and to change its shape and center of mass [37]. The robot's configuration can be changed via a sprawl mechanism and a four-bar extension mechanism (FBEM). The sprawl mechanism can change the angle between wheel pairs (one on each side - port and starboard) and the main body. The FBEM can move the robot's main body forward and backward with respect to the wheels. Independent control of the FBEM and the sprawl mechanism enables shifting the center of mass forward-backwards and upwards-downwards in parallel (as demonstrated in the accompanying video). This high maneuverability of RSTAR allows it to operate successfully in the face of the particularly challenging obstacles commonly found in search and rescue scenarios. RSTAR can crawl on surfaces, climb vertically in a pipe or between two walls, move upside down, and climb obstacles whose height exceeds the diameter of its wheels. However, these unique abilities complicate the conceptualization and the realization of plausible motion behaviors. Therefore, learning with automatic curriculum determination, as proposed in the current study, was tested with the RSTAR.

A significant challenge facing outdoor mobile robots, especially in search and rescue missions, is dealing with uncertainty in the environment. The size and characteristics of the obstacles they must tackle may vary significantly. Therefore, the robot must learn to overcome multiple obstacle variants, but directly learning maneuvers for all variants is prohibitively time-consuming. A potential solution lies in applying curriculum learning, which may be expected to enhance learning efficiency significantly. Nonetheless, when establishing a curriculum for reconfigurable robots such as the RSTAR, it is critical to examine the operations and environments based on the robot's capabilities, which is non-intuitive for humans.

Motion behavior analysis is a preliminary step for automatically establishing a curriculum from the RSTAR's perspective. When controlled by a deep neural network, the motion behavior of RSTAR can be complex and appear stochastic. Complex motion of this type can be analyzed using stochastic models, e.g., Gaussian mixture models (GMMs). Multidimensional spatio-temporal GMMs facilitate parsimonious motion behavior representation [38], [39], [40], [41], [42]. Classical maximum-likelihood-based goodness-of-fit measures (e.g., the $\chi^2$ test) cannot be used for multivariate distributions. However, the distance between the GMMs modeling the motion behaviors can be quantified using stochastic distance measures, e.g., the Hellinger distance (HD) [43], [44]. HD measures the similarity between two probability distributions P, Q and quantifies data separability,

$$D^2_{HD}\left(P||Q\right) = \frac{1}{2}\int_{-\infty}^{\infty}\left(\sqrt{p\left(x\right)} - \sqrt{q\left(x\right)}\right)^2 dx$$
$$= 1 - \int_{-\infty}^{\infty}\sqrt{p\left(x\right)q(x)}dx \qquad (1)$$

HD is a non-negative measure with values between 0 and 1. Higher values of the HD measure are associated with less similarity between the distributions, where 0 implies the distributions are identical. While HD between GMMs cannot be computed analytically, the unscented HD [43] provides a highly accurate estimate.

Using HD, the motion behavior for different obstacle variants can be clustered by hierarchical clustering [45]. The current paper proposes using hierarchical clustering based on HD to automatically establish a curriculum for learning to overcome obstacle variants. The learning is examined for the RSTAR with three obstacle types: a narrow channel, a low entrance, and a step (see associated video).

## II. METHOD

The framework for learning based on an automatically designed learning curriculum adapted to the capabilities of the learning robot has three stages (Figure 2): the initiation stage, which includes data acquisition; the analysis stage, during which a curriculum is formed based on the acquired data; and the curriculum-based learning stage, wherein the robot learns based on the devised curriculum. The performance is tested after the learning has ended.

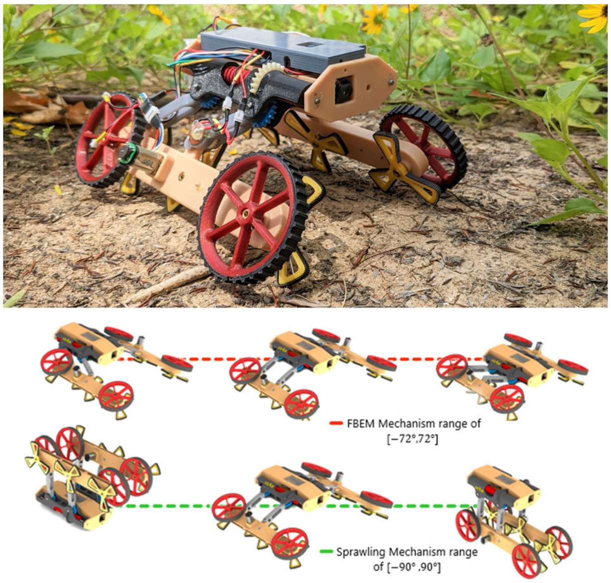
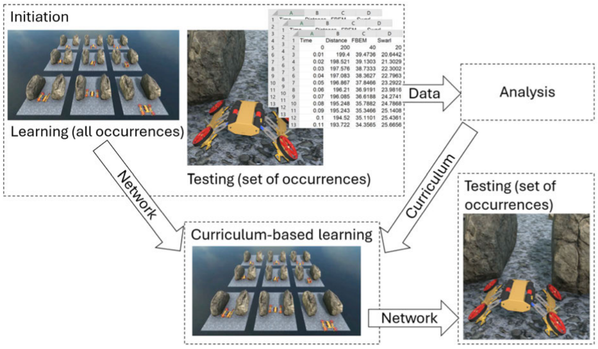


FIGURE 1. Top: RSTAR robot. Bottom: the FBEM angle changing the width and the length of the robot and the sprawl angle, changing the relative angle between the legs and the main body.


Initiation
Learning (all occurrences)
Testing (set of occurrences)
Data
Analysis
Network
Curriculum
Curriculum-based learning
Network
Testing (set of
occurrences)

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Time | Distance | FBEM | Swarl |
| 2 | 0 | 200 | 40 | 20 |
| 3 | 0.01 | 199.4 | 39.4736 | 20.6442 |
| 4 | 0.02 | 198.521 | 39.1303 | 21.3029 |
| 5 | 0.03 | 197.576 | 38.7333 | 22.3002 |
| 6 | 0.04 | 197.083 | 38.3627 | 22.7963 |
| 7 | 0.05 | 196.867 | 37.8466 | 23.2922 |
| 8 | 0.06 | 196.21 | 36.9191 | 23.9816 |
| 9 | 0.07 | 196.085 | 36.6188 | 24.2741 |
| 10 | 0.08 | 195.248 | 35.7882 | 24.7868 |
| 11 | 0.09 | 195.243 | 35.3466 | 25.1408 |
| 12 | 0.1 | 194.52 | 35.1101 | 25.4361 |
| 13 | 0.11 | 193.722 | 34.3565 | 25.6656 |

FIGURE 2. Curriculum determination and learning framework.

### A. INITIATION

The method aims to adapt the curriculum learning to the capabilities of the robot within the given environment. To this end, data regarding the robot's capabilities within the environment are required. To facilitate the acquisition of such data, an initial learning stage is conducted with a pre-determined number of iterations of all the required occurrences of the task, for example, for overcoming a step obstacle for all the size variants of the obstacle. The learning at this stage does not need to converge to successful operation but rather to start forming motion behaviors.

After the initial learning, the robot is tested with a set of occurrences of the required task, e.g., for the task of overcoming a step obstacle, the occurrences can be steps of different heights. The tests are repeated several times for each task occurrence. During the execution, data regarding the robot's behavior is collected and stored. For example, when overcoming a step obstacle, the paths performed by the robot are stored.

### B. ANALYSIS

The data generated during the tests in the initiation stage is used in the analysis stage to generate the appropriate curriculum (Figure 3). The behavior of the robot for each obstacle occurrence is characterized by a GMM distribution function:

$$p(x) = \sum\nolimits_{i=1}^{K} w_i g_i \left(x|\mu_i, \sum\nolimits_i\right), \quad i = 1, \ldots, K, \quad \sum\nolimits_{i=1}^{K} w_i = 1$$

$$g_i\left(x|\mu_i, \sum\nolimits_i\right) = \frac{1}{(2\pi)^{\frac{m}{2}}\sqrt{|\sum_i|}} \times \exp\left\{-\frac{1}{2}(x-\mu_i)' \sum\nolimits_i^{-1} (x-\mu_i)\right\} \tag{2}$$

where $x$ is an m-dimensional vector of motion features, $w_i$ are the mixture weights, $K$ is the number of mixture model components, and $g_i$ are the Gaussian densities with a mean vector $\mu_i$ and a covariance matrix $\Sigma_i$. The parameters of the GMM ($\mu_i$, and $\Sigma_i$) are determined using the expectation-maximization method [46], [47].

The number of model components, $K$, is determined using the Bayesian information criterion (BIC),

$$\text{BIC} = -2L + T \cdot \ln(N) \tag{3}$$

where $L$ is the log-likelihood of the model, $T$ is the number of independent parameters in the GMM, and $N$ is the number of observations used in fitting the model [47].

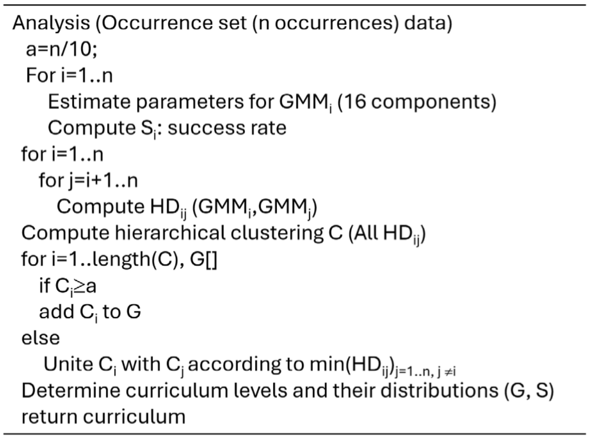
The m-dimensional motion feature vector is spatio-temporal, representing the robot's configuration over time. The distances between the GMMs are characterized using HD. HD values were computed using the unscented transform [43]. Task success rate is also calculated for each obstacle variant.

The HD distances between the GMMs are used for clustering the obstacle variants into groups using agglomerative hierarchical clustering [48]. This clustering determines that obstacle variants for which the robot behaves similarly are clustered into the same group. The initial separation is based on the link inconsistency coefficient (which identifies divisions for which similarities change abruptly) [48]. However, clusters with a small number of obstacle occurrences (less than an order of magnitude with respect to the number of variants) are merged with adjacent clusters based on their distance (according to a hierarchical clustering dendrogram).

The complexity of a group is defined based on the average success rate: the lower the success rate, the higher the group complexity. Groups with a low success rate require additional learning iterations. A level in the current work is defined by a discrete probability function of groups (each group has a probability weight, and the sum of weights of a level is equal to 1). A curriculum is defined by levels, where the number of levels equals the number of groups for which the success rate is low. The levels are arranged according to increasing complexity, and a probability function is defined for each level (2). The probability function is a polynomial distribution function of the groups, and the tasks in each group are uniformly distributed.

$$Z_i \sim U(LB_i, UB_i), \quad i = 1 \ldots N$$
$$P_j \sim \sum\nolimits_{i=1}^{N} w_i Z_i, \quad \sum\nolimits_{i=1}^{N} w_i = 1, j = 1 \ldots M \tag{4}$$

where $N$ is the number of groups; $\text{LB}_\text{i}$ and $\text{UB}_\text{i}$ are the lower and upper bounds, respectively, of the obstacle dimensions in group $i$. The designer selects these bounds, e.g., based on robot capabilities; $w_i$ are the distribution weights, where the weights of the groups at each level vary according to the curriculum; and $M \leq N$ is the number of curriculum levels. At each level $j$, the highest weight is given to one group. Values of lower complexity groups have non-zero weights ($w_i < 0$) to prevent forgetting [17], values of higher complexity groups may also be non-zero to develop some familiarization.



```
Analysis (Occurrence set (n occurrences) data)
  a=n/10;
  For i=1..n
     Estimate parameters for GMM$_i$ (16 components)
     Compute S$_i$: success rate
 for i=1..n
    for j=i+1..n
       Compute HD$_{ij}$ (GMM$_i$,GMM$_j$)
 Compute hierarchical clustering C (All HD$_{ij}$)
 for i=1..length(C), G[]
    if C$_i$≥a
    add C$_i$ to G
 else
     Unite C$_i$ with C$_j$ according to min(HD$_{ij}$)$_{j=1..n, j \neq i}$
 Determine curriculum levels and their distributions (G, S)
 return curriculum
```

**FIGURE 3. Pseudocode for analysis stage.**

### C. CURRICULUM-BASED LEARNING

After the curriculum is determined, the robot continues to learn based on the curriculum. Any deep reinforcement learning algorithm that facilitates transfer learning can be used at this stage. The learning starts using the probability function of the first level for task presentation. At each level, the learning continues for a preset initial number of iterations, $k$. After that, the quality of the learning is determined, and a decision is made as to whether to continue learning at the current level or to progress to the next level. If the decision is to remain at the current level, learning continues for an additional predetermined number of iterations $p$, where $p < k$ and the learning quality is re-tested until a decision to continue to the next level is reached. After the robot completes the final level, the learning is completed.

In reinforcement learning, the agent maximizes the expected cumulative reward. Therefore, the learning quality is defined based on examining whether the current change in the cumulative reward is significantly smaller than the change in the cumulative reward at the beginning of the level,

$$\frac{r_{current_L} - r_{current_F}}{r_{beginning_L} - r_{beginning_F}} < 0.1 \tag{5}$$

where $F$ indicates the sum of the first $W$ learning steps of the iteration, and $L$ indicates the sum of the last $W$ learning steps of the iteration. This test determines whether the robot is still learning and should remain on the current level or has already learned enough and can continue to the next level.

## III. EXPERIMENT

Learning based on the developed method was compared to learning without a curriculum. The RSTAR robot learned to overcome obstacles with each method in a simulated environment (see accompanying video). Three obstacle types were defined, where each obstacle posed different challenges for the RSTAR robot. We hypothesized that for the same number of steps, learning with the automatically determined curriculum would lead to better performance than learning without a curriculum.

### A. ENVIRONMENT

The simulated RSTAR was modeled based on a physical RSTAR, length 150 mm, width 115–290 mm, and height 42–125 mm [35], [36]. The robot learned to overcome three different obstacle types: a narrow channel, a low entrance, and a step (Figure 4). Overcoming each obstacle type was learned in a separate experiment. The target location was placed such that the robot had to overcome the obstacle to reach the target. The narrow channel variants were 180–320 mm wide, so the robot was required to reduce its width to pass through the channels (the narrower the channel, the more complex the task). The low entrance variants were 55–140 mm high, so the robot was required to lower its body to crawl underneath the obstacle (the lower the entrance, the more complex the task). The step obstacle variants were 21–50 mm high, so the robot was required to climb over the step while maintaining its balance (the higher the step, the more complex the task). Obstacle sizes were determined according to the capabilities of the robot. Based on prior work with the robot, the step obstacle is known to be the most difficult for the robot to overcome, and the narrow channel is the simplest.

The simulations were conducted using a Unity® software environment (real-time engine development platform), and the learning was programmed using the Unity Machine Learning Agents Toolkit (ML-agents, https://unity.com/products/machine-learning-agents). The training process was conducted in nine environments simultaneously, where the agents share a common learned behavior (network weights) to reduce the learning time. The simulations were performed with an Intel® Core™ i7-7700 processor 3.6GHz, 16GB RAM running a Windows 10 operating system x64 bits. The GMMs were computed using Matlab® (Version R20201a, Mathworks, USA). The statistical analysis was conducted with R using the RStudio interface (Version 1.2.5001, Open Source).

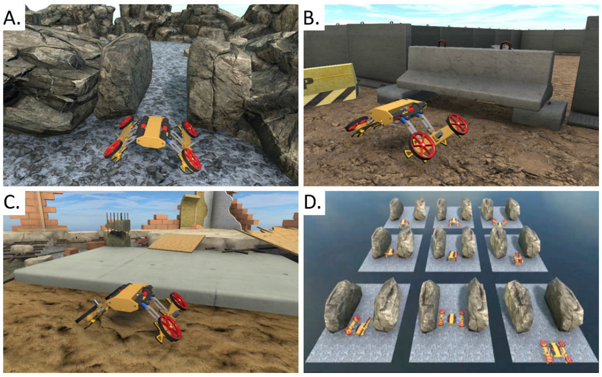


**FIGURE 4. The simulated environments: (A) Narrow channel, (B) Low entrance, and (C) Step. (D) Example of nine concurrent training environments.**

### B. REINFORCEMENT LEARNING

RSTAR learned to overcome the variants of the three obstacle types by using the Proximal Policy Optimization (PPO) DRL algorithm [49]. PPO is an on-policy gradient-based optimization algorithm implemented in Unity Machine Learning Agents Toolkit. The network was defined as a four-layer neural network with 256 neurons in each layer. The observations were of the robot configuration, namely, the yaw angle, the pitch angle (for the step obstacle), the FBEM angle, the sprawl angle, and the tangential speed of the robot. In addition, the observations included the distance of the robot from the obstacle and the dimensions of the obstacles, namely, width for the narrow channel or height for the low entrance or the step. The observations of the robot's actions were based on the robot's velocity in the forward (Z axis) direction (maximum change in each step ±1 mm/s) and changes in the FBEM and sprawl angles (maximal change in each step ±0.08°, which were chosen empirically to ensure smooth motion). The velocity range was ±250 mm/s, the FBEM angle range was ±72°, and the sprawl angle range was 0–180°. The reward was defined according to Eq. 6.

$$\begin{aligned}
K_b &= \begin{cases} 1 & (|Yaw| > 100) \,|\, (|Pitch| > 100) \\ 0 & else \end{cases} \qquad\qquad , \\
K_{yaw} &= \begin{cases} 0.0001 & |\boldsymbol{\theta}_{yaw}| \le 5 \\ 0.001 & 5 < |\boldsymbol{\theta}_{yaw}| \le 10, \\ 0.01 & 10 < |\boldsymbol{\theta}_{yaw}| \end{cases} \\
R_f &= \begin{cases} 5 & D_z = 0 \\ -5 & (D_z \neq 0)\&(R_Y < -1), \\ 0 & else \end{cases} \\
R_i &= -\frac{1 + 2K_b(i)}{S_m} - K_{yaw}(i)\, D_z(i) - 10^{-8}\overline{S(i)} \\
&\quad - 0.005\,|\Delta_S(i)| - 0.005\,|\Delta_{FBEM}(i)|, \\
V &= \sum\nolimits_{i=1}^{n} R_i + R_f
\end{aligned} \tag{6}$$

where $R_f$ is the final cost, $R_i$ is the cost at step $i$, $V$ is a value function, $D_z$ is the distance of the robot from the target along the $Z$ (forward) axis, $R_Y$ is the height of the robot, $S_m$ is the maximal number of learning steps in a simulation episode, $K_b$ is the backward fall identifier, $K_{yaw}$ is the yaw angle cost (attenuated by i), $\overline{S\ (i)}$ is the tangential speed at step $i$, $\Delta_s$ is the change in the sprawl angle, and $\Delta_{FBEM}$ is the change in the FBEM angle.

The final cost $R_f$ rewards the robot for reaching the goal or strongly penalizes it for falling off the surface. The step cost $R_i$ has five components. The first component is divided by the maximal number of learning steps and is related to simulation time and the robot falling on its back during the run. The second component is associated with the yaw deviation and promotes a straight approach toward the obstacle. The remaining three components are designed to encourage a reduction in energy consumption by imposing penalties for high speed and excessive motion of the sprawl and FBEM mechanisms.

### C. PROCEDURE

In the initiation stage, the number of learning steps for each learning episode was set to 500,000 for the narrow channel and the low entrance and 1,000,000 for the step obstacle. The difference in the number of learning steps was the greater difficulty of overcoming the step obstacle. The step obstacle is more complex since the robot must move its center of mass both forward-backward and up-down. In each learning episode, an obstacle occurrence was randomly selected from the appropriate range of obstacle sizes, and the configuration (sprawl and FBEM angles) and position of the robot in the workspace were randomly determined. The initiation stage was conducted with a different seed in each run and twice for each obstacle, so a total of 6 independent networks (two for each of the three obstacle types) were learned.

A set of obstacle occurrences evenly distributed within the size range of the obstacle was defined for each obstacle type. For the narrow channel, the set included 15 widths; for the low entrance, the set included 18 heights; and for the step, the set included 30 heights. After the initial learning stage, data was collected for each obstacle occurrence with the learned networks. Thirty repetitions of the task execution were conducted for each obstacle occurrence, with each of the two networks learned for the obstacle type. Each repetition started from the same initial position and configuration. Repetitions lasted 20 s at most, or less if the robot overcame the obstacle faster. There were 450 repetitions for the narrow channel, 540 for the low entrance, and 900 for the step obstacle. The data collected in each simulation time step included the time elapsed from the start of the trial, the sprawl angle, the FBEM angle, and the Euclidean distance from the target. The collected data was arranged as a four-dimensional feature vector.

Four-dimensional GMMs were computed based on the feature vectors for each occurrence of each obstacle, i.e., 15 models for the narrow channel, 18 models for the low entrance, and 30 models for the step. The number of GMM components was set at 16 after testing the models with 2–20 components for the highest step height based on the minimum estimated BIC. This step is the most challenging obstacle for the robot to overcome and, therefore, most probably requires the most model components. HD was computed between all models of each obstacle.

For each obstacle, GMMs were clustered into groups using hierarchical clustering with HD as the distance measure. If a group contained two GMMs or less, it was merged with the adjacent group to which it was most similar according to the dendrogram. The task complexity of the group was determined by the average success rates in overcoming the obstacle in each group, as ‘easy,’ ‘medium,’ or ‘difficult.’ A curriculum was determined for each obstacle type based on the formed groups and the average success rate.

The robot continued to learn using transfer learning with or without the curriculum. With the curriculum, the robot progressed based on an automatic performance examination after pre-determined training durations. Each curriculum stage started with $k = 500{,}000$ learning steps. The number of learning steps averaged at the start and end, $W$, was set to 5,000. If the robot did not pass the examination, it learned for another $p = 100{,}000$ learning steps, after which it was examined again. After ten consecutive iterations, the robot progressed to the next curriculum stage without an additional test.

The learning duration for both with and without a curriculum was determined according to the progress attained when learning with a curriculum. Learning with the curriculum was terminated when the robot successfully completed the final curriculum level. For each obstacle type, the same duration and the same initial network were used when learning without a curriculum. This way, pairs of learning processes (with and without a curriculum) that lasted the same time were created. Four networks were learned for each obstacle, two with and two without a curriculum.

### D. ANALYSIS

To test the performance of the learned behaviors, the robot was required to overcome three obstacles, each in a different simulation environment: A narrow channel with variable width between 180 mm to 320 mm (in steps of 10 mm – 15 values), a low entrance with variable height between 55 mm and 140 mm (in steps of 5 mm – 18 values), and a step with variable height between 21 mm and 50 mm (in steps of 1 mm – 30 values). For each obstacle occurrence, the robot was required to overcome the obstacle 30 times with each of the four learned networks (120 trials). For each obstacle, the orientation and configuration of the robot at the beginning of each trial were the same: the robot was placed at the same distance from the obstacle, the sprawl angle was initialized to 48°, and the FBEM angle was initialized to 0°. The trial ended when the robot overcame the obstacle or when the pre-allocated task execution time had expired.

In each trial, data regarding the robot’s trajectory were collected. The following measures were computed: The success rate was graded as yes/no based on whether the robot reached the goal within the repetition duration. The task completion time was calculated in seconds, and if the goal was not reached, the time was set as the repetition duration. The path length of the sprawl angle and the path length of the FBEM angle (in degrees) were computed for successful paths, i.e., summation of the absolute values of the respective angles (sprawl or FBEM). The distance traveled in the environment was not used as a measure since it was highly correlated to the distance between the initial position and the obstacle and to the time duration of the task.

### E. STATISTICAL ANALYSIS

All statistical analyses were performed using R Studio IDE for R (version 1.2.5001, Open source), with a significance level of 5%. All analyses were conducted separately for each obstacle.

The success rates with and without a curriculum were compared with a 2-sample test for equality of proportions with continuity correction [50]. Since trial duration caps task completion time, task completion time was analyzed using survival analysis [51]. Survival analysis considers censoring, e.g., repetitions in which success was not reached during the trial duration, preventing bias in the estimates of the distribution. The survival curves (Kaplan-Meier estimates of survival) with and without a curriculum were compared using the log-rank test (Mantel-Haenszel) [52].

Sprawl and FBEM angles were analyzed with a linear mixed model (LMM) with multiple comparisons computed with the restricted maximum likelihood (REML) criterion for convergence. The fixed factors included complexity group (i.e., 'easy,' 'medium,' 'difficult'), learning method (with or without a curriculum), and their interaction. The random effect was the initial network used (two networks).

## IV. RESULTS AND DISCUSSION

### A. INITIAL LEARNING AND ANALYSIS

The feature vectors for each occurrence were successfully recorded for all three obstacle types. Figure 5 depicts a scatter plot of recorded data from all the repetitions of the 50 mm step. The shift in the FBEM angle when reaching the obstacle (for balancing the robot when climbing the high step) is apparent. The scatter plot of the points sampled from the estimated model is similar to the scatter plot of the recorded data. The BIC values for GMMs with 2-20 components show that 16 components are sufficient for the model (Figure 5C).

For all three obstacle types, the hierarchical clustering algorithm found three groups (Figure 6). The success rates (Table 1) of overcoming obstacles in each group were consistent with our understanding of the task, i.e., the success rate is lower for narrower channels, lower entrances, and higher steps. The groups were named in accordance with the success rates, i.e., easy, medium (in case of three groups), difficult.

For the narrow channel, the 'difficult' group had only two obstacle widths. Therefore, it was merged with the adjacent 'medium' group. For the low entrance, the 'medium' group contained only two heights, and it was more similar (based on the dendrogram) to the adjacent 'easy' group than to the

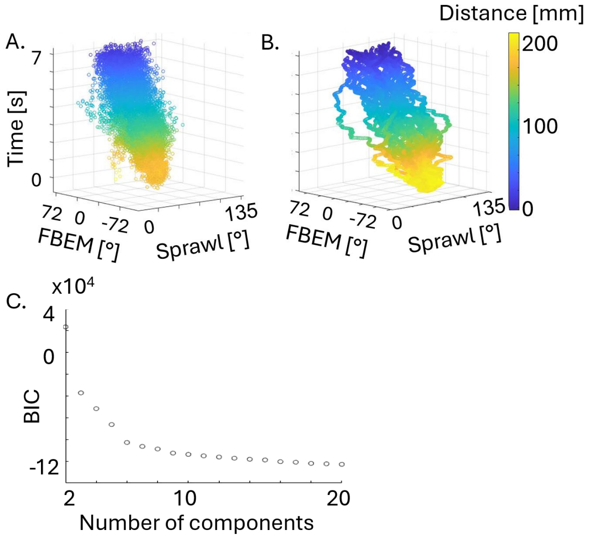


FIGURE 5. Constructing the GMM model for a 50 mm Step obstacle. Color codes distance from target. A. Scatter plot of recorded values from all initial learning repetitions. B. Scatter plot of values randomly sampled from the estimated GMM with 16 components. C. BIC values computed for estimated GMMs with 2-20 components.

TABLE 1. Success rate in overcoming the obstacle at the end of the initial learning.

| Obstacle | Cluster | Success rate (%) |
|---|---|---|
| Narrow channel | Easy | 100 |
| | Difficult | 40 |
| Low entrance | Easy | 99 |
| | Difficult | 76 |
| Step | Easy | 94 |
| | Medium | 19 |
| | Difficult | 0 |

TABLE 2. Curricula for the obstacle types by level (E: 'easy', M: 'medium', D: 'difficult').

| | Level 1 [%] | Level 2 [%] | Level 3 [%] |
|---|---|---|---|
| Narrow channel | E: 30<br>D: 70 | | |
| Low entrance | E: 30<br>D: 70 | | |
| Step | E: 70<br>M: 30<br>D: 0 | E: 15<br>M: 70<br>D: 15 | E: 10<br>M: 20<br>D: 70 |

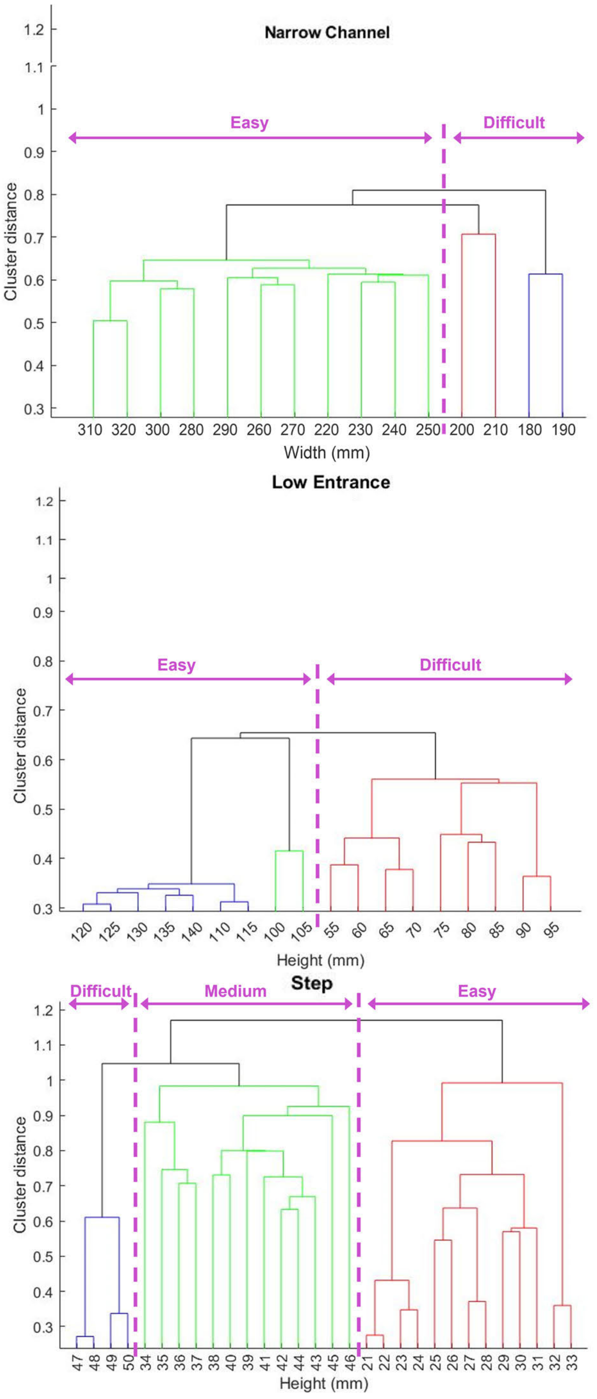


**FIGURE 6.** Dendrograms of the meaningful dimension clusters for the three obstacles, as found by the hierarchical clustering algorithm (which is the reason why the order is not monotonic in numbers). From top to bottom: narrow channel (width), low entrance (height), and step (height). Dotted purple lines indicate separate complexity groups.

'difficult'. Therefore, the 'easy' and 'medium' groups were merged.

After the initial training, the policy was very good for the easy groups, especially for the wider narrow channels and higher low entrances. To overcome these obstacles, the robot needs very little change (if at all) in its configuration. The performance at the end of the initial stage was poor for all other obstacles, i.e., the medium and difficult groups, for which the robot needs to change its configuration to pass. This finding strengthens the basic hypothesis on which the method is based, that indeed, the algorithm can identify groups of obstacle variants based on robot behavior and that these different obstacle variants pose different difficulty levels for the robot motor behavior learning process.

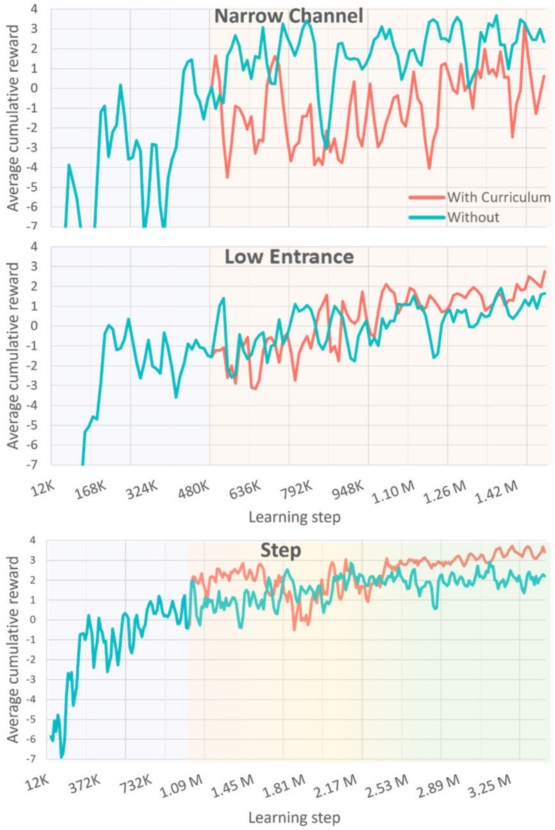


**FIGURE 7.** Average cumulative reward (Y axis) as function of the number of learning steps (X axis), during the training with and without a curriculum. Orange graphs indicate learning with a curriculum, and cyan graphs indicate learning without a curriculum. The initial learning is the same for both methods, and therefore the lines were merged; shown as cyan graphs on a gray background. According to the curriculum, for the narrow channel and the low entrance, there was only one level, represented by an orange background. For the step obstacle, the curriculum consisted of three levels, represented by the gradated background (orange to green). The learned tasks differ between the learning methods. When training without a curriculum the obstacle occurrences are drawn uniformly from the entire range throughout the training. When learning with a curriculum, there are more difficult tasks as the learning progresses. Therefore, the overall learning success cannot be directly deduced from these graphs. Accordingly, a testing stage and a statistical analysis were conducted following the learning.

The curriculum levels and their distribution (Table 2) were determined based on the complexity groups and success rates. The robot is assumed to have learned the task when the success rate is above 95%. For the narrow channel and the low entrance, there were two groups, and in addition, for these obstacle types, the success rates for the 'easy' group were above 95%. Therefore, the curriculum designed for these two

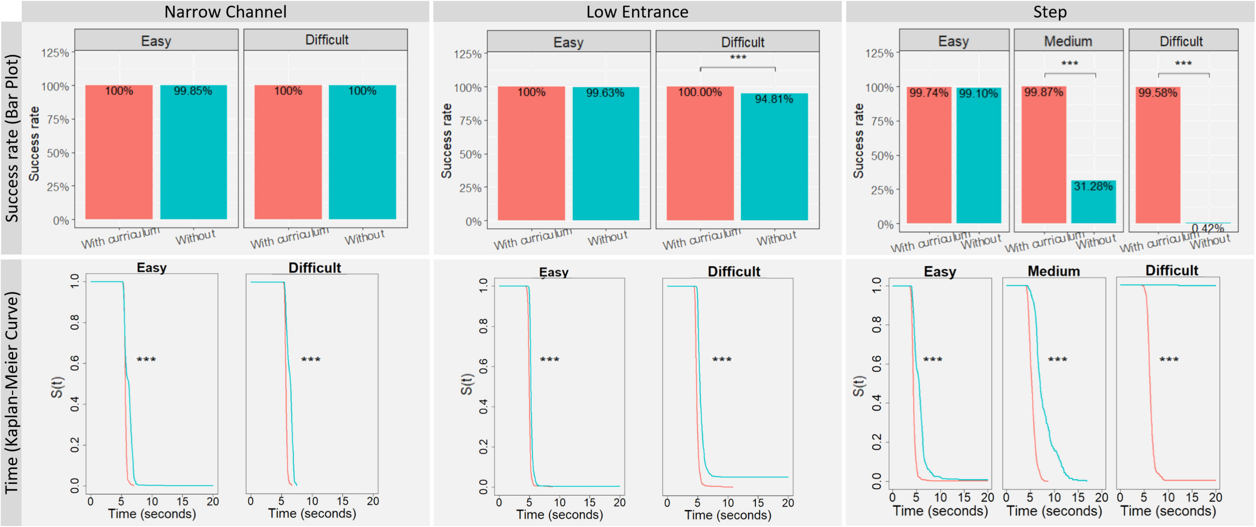


**FIGURE 8.** Success rate (top) and time (bottom) statistics, by complexity and obstacle type. From left to right: narrow channel, low entrance, and step. Dark pink indicates learning with curriculum, and cyan indicates learning without curriculum. Significance is shown for both the bar-plots and the Kaplan-Meier curves estimating survival, i.e., the estimated time it takes the robot to overcome the obstacle. Significance values are indicated: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

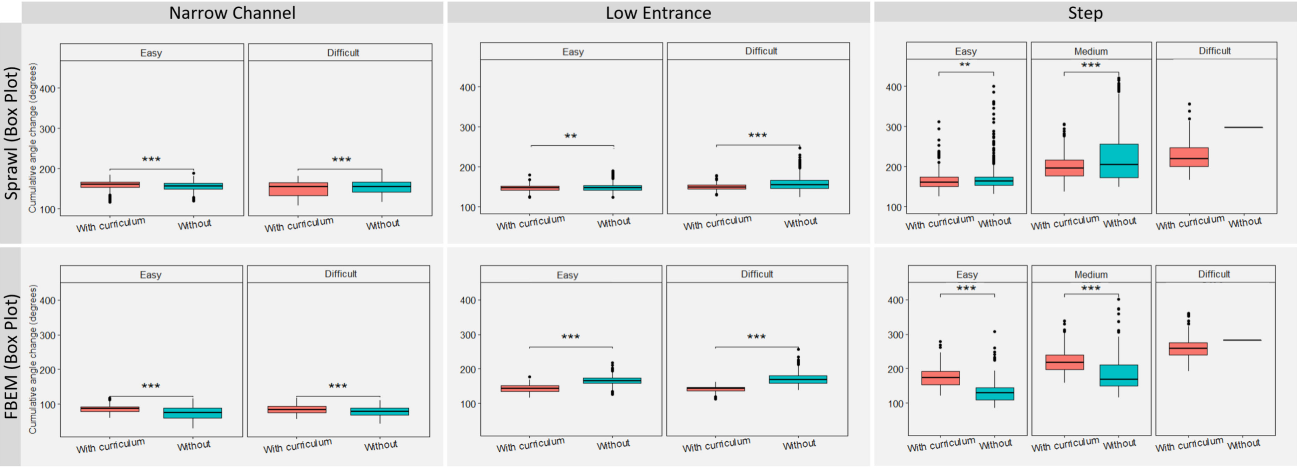


**FIGURE 9.** Box plots of sprawl angle (top) and FBEM angle (bottom) by obstacle and complexity group. From left to right: narrow channel, low entrance, and step. Red indicates learning with curriculum, and cyan indicates learning without curriculum. Significance values: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

obstacle types had only a single level in which the ‘difficult’ group was dominant, i.e., had a high probability weight. For the step obstacle, there were three groups, and the success rates for all three groups were below 95%. Therefore, a curriculum with three levels, ‘easy,’ ‘medium’, and ‘difficult’, was designed.

### B. CURRICULUM LEARNING

The convergence of the reward functions is presented in Figure 7. For the step obstacle, the cumulative rewards converged for both learning methods. For curriculum learning, the reward increased after the initial learning (after 1 million learning steps), as the curriculum starts with the ‘easy’ task level. The cumulative reward decreased briefly when moving to the next (more complex) task level (after about 1.8 million learning steps). The effect of curriculum learning is less apparent in the convergence of the narrow channel and the low entrance. For both these obstacles, when learning with a curriculum, there is an initial decrease in the cumulative reward when curriculum learning starts (after the initiation stage), which coincides with the level’s emphasis on the ‘difficult’ group.

For the success rate (Figure 8, top), the differences between the methods depend on obstacle and complexity group. For the narrow channel, there was no difference in the success rate with or without a curriculum. For the low entrance, there was no difference in the success rate with or without a curriculum for the ‘easy’ group, but the success rate was higher when learning with a curriculum for the 'difficult' group ($\chi^2 = 26.73$, $p < 0.001$). For the step obstacle, there was no difference in the success rate with or without a curriculum for the 'easy' group, but the success rate was higher when learning with a curriculum for the 'difficult' ($\chi^2 = 468.07$, $p < 0.001$) and 'medium' ($\chi^2 = 809.76$, $p < 0.001$) groups. The difference between the methods was very significant in the 'difficult' group as the robot overcame the step obstacle in only one of the trials when learning without a curriculum but in 99.6% of the trials when learning with a curriculum.

For all obstacles and all complexity groups, the survival curves for learning with a curriculum converged to zero earlier ($p < 0.05$) than without the curriculum, i.e., after learning with a curriculum, it took the robot less time to overcome the obstacle (Figure 8, bottom).

For all obstacles, there were differences between the cumulative sprawl angle for all groups [narrow channel: 'difficult' ($t_{1794} = -4.77$, $p < 0.001$), 'easy' ($t_{1794} = 6.04$, $p < 0.001$); low entry: 'difficult' ($t_{2125} = -11.47$, $p < 0.001$), 'easy' ($t_{2125} = -2.75$, $P < 0.01$); step: 'medium' ($t_{2807} = -4.77$, $p < 0.001$), 'easy' ($t_{2807} = -2.98$, $p < 0.01$) (Figure 9, top). For all obstacles, there were differences between the cumulative FBEM angle for all groups [narrow channel: 'difficult' ($t_{1794} = 5.27$, $p < 0.001$), 'easy' ($t_{1794} = 14.38$, $p < 0.001$); low entry: 'difficult' ($t_{2125} = -39.13$, $p < 0.001$), Easy ($t_{2125} = -32.71$, $p < 0.001$); step: 'medium' ($t_{2807} = 16.42$, $p < 0.001$), 'easy' ($t_{2807} = 29.20$, $p < .001$) (Figure 9, bottom)]. For the 'difficult' group of the step obstacle, the sprawl and the FBEM angles could not be compared because there was only one successful trial when learning without a curriculum.

## V. CONCLUSION

The robot's behavior (e.g., sprawl and FBEM angle motion profiles) differed when learning with and without a curriculum. The behaviors learned with a curriculum enabled the robot to overcome the three obstacle types more successfully and more rapidly than the behaviors learned without a curriculum. The advantage of learning with the prescribed curriculum was more significant for the more challenging obstacle variants (narrower channels, lower entrances, and higher steps). For example, for the low entrance, the success rate in the 'difficult' level was 95% when learning without the curriculum and 100% when learning with the curriculum. Even more markedly, for the 'medium' level of the step obstacle, the success rate was only 31% when learning without the curriculum and 100% when learning with the curriculum. Finally, and most strikingly, for the 'difficult' level of the step obstacle, learning without a curriculum produced only one successful trial (out of 30 repetitions), whereas almost all trials (99.6%) were successful for learning with a curriculum.

The developed method is especially suitable for developing curricula for tasks where the division into sub-groups and their difficulty are unclear to the human operator. Such tasks are commonly encountered when using reconfigurable robots since reconfigurable robots present multiple motion possibilities. These motion capabilities may be critical for search and rescue missions where the robot encounters challenging surroundings requiring dexterous motion. The developed method builds on analyzing the task from the robot's perspective and is, therefore, suitable for situations where the operator's intuition regarding the required operation may not be sufficient.

The task groups are clustered based on the distance between the motion models learned based on data from the initial training stage. The model features are related to the robot's dynamics and, in some sense, to task requirements. Indeed, for tasks or robots that require different features, measuring the distance between models is not supported by the current method. In the current work, the features (the robot's degrees of freedom) were pre-determined. In future work, features can be automatically determined based on methods such as principal component analysis or regression analysis [40].

The current study examined the learning scores periodically to determine a learning level and task distribution. The period length and distribution coefficients at each level were empirically determined and clearly affected the results. For example, neglecting to retain some weight to previously learned groups causes forgetting and hinders overall performance. On the other hand, giving too much weight to previously learned groups slows up the learning process with unnecessary learning instances. The periodic estimation and group distribution should be compared to stepwise group selection (as in [15], [16], [23], [24], [25], and [26]), which obviates the need for determining period and distribution coefficients.

The developed method was tested in the current work in simulation only. Prior work by the authors has successfully tested discrete policies learned in simulation using Q-learning with a physical setup [35]. Not only were the learned policies successful in overcoming the physical obstacles, but they even outperformed the performance attained by human operators. This seamless transfer from the simulation into the physical setup is partly due to the advanced maneuverability of the RSTAR. The current physical model and controller are unsuitable for run-time operation with network-based control. We are currently constructing a physical setup suitable for testing these issues.

## ACKNOWLEDGMENT

The authors would like to thank Prof. Yisrael Parmet for assisting in the statistical analysis and Nissim Abuhatzira for his technical assistance. They thank Inez Mureinik for her assistance in editing the manuscript.

## REFERENCES

[1] F. Pasandideh, J. P. J. D. Costa, R. Kunst, W. Hardjawana, and E. P. de Freitas, "A systematic literature review of flying ad hoc networks: State-of-the-art, challenges, and perspectives," *J. Field Robot.*, vol. 40, no. 4, pp. 955–979, 2023, doi: 10.1002/rob.22157.

[2] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, pp. 362–386, Apr. 2020, doi: 10.1002/rob.21918.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero, "Extending the OpenAI gym for robotics: A toolkit for reinforcement learning using ROS and gazebo," 2016, *arXiv:1608.05742*.

[5] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013, doi: 10.1177/0278364913495721.

[6] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *J. Mach. Learn. Res.*, vol. 21, pp. 1–50, Jan. 2020.

[7] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner, "TACO: Learning task decomposition via temporal alignment for control," 2018, *arXiv:1803.01840*.

[8] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, p. 9, Dec. 2016.

[9] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, no. 56, pp. 1633–1685, 2009.

[10] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," 2020, *arXiv:2009.07888*.

[11] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 6, pp. 1526–1565, 2022, doi: 10.1007/s11263-022-01611-x.

[12] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, vol. 382, Jan. 2009, pp. 41–48, doi: 10.1145/1553374.1553380.

[13] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 4, 2018, pp. 2458–2471.

[14] S. Narvekar, "Curriculum learning in reinforcement learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 5195–5196, doi: 10.24963/ijcai.2017/757.

[15] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2216–2226, Jun. 2018, doi: 10.1109/TNNLS.2018.2790981.

[16] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher–student curriculum learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3732–3740, Sep. 2020, doi: 10.1109/TNNLS.2019.2934906.

[17] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[18] B. Ivanovic, J. Harrison, A. Sharma, M. Chen, and M. Pavone, "BaRC: Backward reachability curriculum for robotic reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 15–21, doi: 10.1109/ICRA.2019.8794206.

[19] M. Duarte, S. Oliveira, and A. L. Christensen, "Hierarchical evolution of robotic controllers for complex tasks," in *Proc. IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL)*, San Diego, CA, USA, Nov. 2012, pp. 1–6, doi: 10.1109/DEVLRN.2012.6400828.

[20] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," 2017, *arXiv:1707.05300*.

[21] J. Andreas, D. Klein, and S. Levine, "Modular multitask reinforcement learning with policy sketches," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 166–175.

[22] D. Zhao, J. Zhu, Z. Guo, and B. Zhang, "Curriculum learning for deep generative models with clustering," 2019, *arXiv:1906.11594*.

[23] P. Klink, C. D'Eramo, J. Peters, and J. Pajarinen, "Self-paced deep reinforcement learning," in *Proc. 34th Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–12.

[24] P. Klink, C. D'Eramo, J. Peters, and J. Pajarinen, "On the benefit of optimal transport for curriculum reinforcement learning," 2023, *arXiv:2309.14091*.

[25] M. Nesterova, A. Skrynnik, and A. Panov, "Reinforcement learning with success induced task prioritization," in *Advances in Computational Intelligence* (Lecture Notes in Computer Science), vol. 13612, O. P. Lagunas, J. Martinez-Miranda, and B. M. Seis, Eds. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-19493-1_8.

[26] M. Jiang, E. Grefenstette, and T. Rocktaschel, "Prioritized level replay," 2020, *arXiv:2010.03934*.

[27] Y. Liu and G. Nejat, "Robotic urban search and rescue: A survey from the control perspective," *J. Intell. Robotic Syst., Theory Appl.*, vol. 72, no. 2, pp. 147–165, 2013, doi: 10.1007/s10846-013-9822-x.

[28] T. S. A. Attia, "Design and development of a novel reconfigurable wheeled robot for off-road applications," Ph.D. dissertation, Dept. Mech. Eng., Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2018.

[29] G. Freitas, F. Lizarralde, L. Hsu, and M. Bergerman, "Terrain model-based anticipative control for articulated vehicles with low bandwidth actuators," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 382–389, doi: 10.1109/ICRA.2013.6630604.

[30] G. Freitas, F. Lizarralde, L. Hsu, and N. R. S. D. Reis, "Kinematic reconfigurability of mobile robots on irregular terrains," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 1340–1345, doi: 10.1109/ROBOT.2009.5152309.

[31] W. Reid, F. J. Pérez-Grau, A. H. Göktogan, and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a Martian analog surface," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5596–5602, doi: 10.1109/ICRA.2016.7487777.

[32] H. Lee, Y. Shen, C. H. Yu, G. Singh, and A. Y. Ng, "Quadruped robot obstacle negotiation via reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 3003–3010, doi: 10.1109/ROBOT.2006.1642158.

[33] K. Zhang, F. Niroui, M. Ficocelli, and G. Nejat, "Robot navigation of environments with unknown rough terrain using deep reinforcement learning," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Aug. 2018, pp. 1–7, doi: 10.1109/SSRR.2018.8468643.

[34] M. Totani, N. Sato, and Y. Morita, "Step climbing method for crawler type rescue robot using reinforcement learning with proximal policy optimization," in *Proc. 12th Int. Workshop Robot Motion Control (RoMoCo)*, Jul. 2019, pp. 154–159, doi: 10.1109/RoMoCo.2019.8787360.

[35] L. Yehezkel, S. Berman, and D. Zarrouk, "Overcoming obstacles with a reconfigurable robot using reinforcement learning," *IEEE Access*, vol. 8, pp. 217541–217553, 2020, doi: 10.1109/ACCESS.2020.3040896.

[36] O. Simhon, Z. Karni, S. Berman, and D. Zarrouk, "Overcoming obstacles with a reconfigurable robot using deep reinforcement learning based on a mechanical work-energy reward function," *IEEE Access*, vol. 11, pp. 47681–47689, 2023, doi: 10.1109/ACCESS.2023.3274675.

[37] D. Zarrouk and L. Yehezkel, "Rising STAR: A highly reconfigurable sprawl tuned robot," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1888–1895, Jul. 2018, doi: 10.1109/LRA.2018.2805165.

[38] I. Davidowitz, Y. Parmet, S. Frenkel-Toledo, M. C. Banina, N. Soroker, J. M. Solomon, D. G. Liebermann, M. F. Levin, and S. Berman, "Relationship between spasticity and upper-limb movement disorders in individuals with subacute stroke using stochastic spatiotemporal modeling," *Neurorehabilitation Neural Repair*, vol. 33, no. 2, pp. 141–152, 2019, doi: 10.1177/1545968319826050.

[39] H. Lackritz, Y. Parmet, S. Frenkel-Toledo, M. C. Banina, N. Soroker, J. M. Solomon, D. G. Liebermann, M. F. Levin, and S. Berman, "Effect of post-stroke spasticity on voluntary movement of the upper limb," *J. NeuroEng. Rehabil.*, vol. 18, no. 1, p. 81, 2021, doi: 10.1186/s12984-021-00876-6.

[40] C. Zhang, H. Zhang, and L. E. Parker, "Feature space decomposition for effective robot adaptation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 441–448, doi: 10.1109/IROS.2015.7353410.

[41] S. Khansari-Zadeh and A. Billard, "Learning stable nonlineardynamical systems with Gaussian mixture models," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 943–957, Oct. 2011, doi: 10.1109/TRO.2011.2159412.

[42] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Adv. Robot.*, vol. 23, no. 15, pp. 2059–2076, 2009.

[43] R. N. Tamura and D. D. Boos, "Minimum Hellinger distance estimation for multivariate location and covariance," *J. Amer. Stat. Assoc.*, vol. 81, no. 393, pp. 9–223, 1986.

[44] M. Kristan, A. Leonardis, and D. Skocaj, "Multivariate online kernel density estimation with Gaussian kernels," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2630–2642, 2011.

[45] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intell. Data Anal.*, vol. 11, no. 6, pp. 583–605, 2007, doi: 10.3233/ida-2007-11602.

[46] D. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 2, pp. 827–832, Jan. 2015, doi: 10.1007/978-1-4899-7488-4_196.

[47] D. A. Cohn, Z. Ghahramani, and I. J. Michael, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, Mar. 1996.

[48] A. Jatain, A. Nagpal, and D. Gaur, "Agglomerative hierarchical approach for clustering components of similar reusability," *Int. J. Comput. Appl.*, vol. 68, no. 2, pp. 33–37, 2013, doi: 10.5120/11553-6832.

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

[50] R. G. Newcombe, "Interval estimation for the difference between independent proportions: Comparison of eleven methods," *Statist. Med.*, vol. 17, pp. 873–890, Apr. 1998.

[51] T. Krishnan, "Survival analysis," in *Essentials of Business Analytics* (International Series in Operations Research & Management Science), vol. 264, B. Pochiraju and S. Seshadri, Eds. Springer, 2019, pp. 439–458, doi: 10.1007/978-3-319-68837-4_14.

[52] D. P. Harrington and T. R. Fleming, "A class of rank test procedures for censored survival data," *Biometrika*, vol. 69, no. 3, pp. 553–566, 1982.

**ZOHAR KARNI** received the B.Sc. and M.Sc. degrees from Ben Gurion University of the Negev, Israel. Her research interests include intelligent systems and machine learning.

**OR SIMHON** received the B.Sc. and M.Sc. degrees from Ben Gurion University of the Negev, Israel. His research interest includes imparting autonomous abilities for reconfigurable robots using reinforcement learning.

**DAVID ZARROUK** (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Technion, Israel, in 2005, 2008, and 2011, respectively.

From 2011 to 2013, he was a Postdoctoral Fellow with the Biomimetic Millisystems Laboratory, UC Berkeley. He is currently an Associate Professor with the Mechanical Engineering Department, Ben Gurion University of the Negev. His research interests include medical robotics, robotics in flexible and slippery surfaces interactions, biomimetics, and minimally actuated mechanisms. He received multiple prizes for excellence in research and teaching, which include a Fulbright Postdoctoral Fellowship, in 2011, Fulbright-Ilan Ramon Postdoctoral Fellowship for most prominent Israeli Fulbright Fellow, in 2011, Hershel Rich Innovation Award, Aharon and Ovadia Barazani prize for excellent Ph.D. thesis, and Alfred and Yehuda Weisman prize for consistent excellence in teaching.

**SIGAL BERMAN** (Senior Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Technion, and the M.Sc. degree in electrical and computer engineering and the Ph.D. degree in industrial engineering from the Ben Gurion University of the Negev. She is currently a Professor with the Department of Industrial Engineering and Management, Ben Gurion University of the Negev, Beer Sheva. She leads the Intelligent Systems Engineering Laboratory (ISEL), where her research focuses on the analysis and engineering of intelligent systems capable of dexterous motion. She develops deterministic and stochastic models for the synthesis of robotic motion and the analysis of human motion.