machines

MDPI

Article

Self-Generating Evaluations for Robot's Autonomy Based on Sensor Input

Yuma Sakamoto and Kentarou Kurashige *

Division of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan; 22043022@mmm.muroran-it.ac.jp
* Correspondence: kentarou@muroran-it.ac.jp; Tel.: +81-143-46-5400

Abstract: Reinforcement learning has been explored within the context of robot operation in different environments. Designing the reward function in reinforcement learning is challenging for designers because it requires specialized knowledge. To reduce the design burden, we propose a reward design method that is independent of both specific environments and tasks in which reinforcement learning robots evaluate and generate rewards autonomously based on sensor information received from the environment. This method allows the robot to operate autonomously based on sensors. However, the existing approach to adaption attempts to adapt without considering the input properties for the strength of the sensor input, which may cause a robot to learn harmful actions from the environment. In this study, we propose a method for changing the threshold of a sensor input while considering the strength of the input and other properties. We also demonstrate the utility of the proposed method by presenting the results of simulation experiments on a path-finding problem conducted in an environment with sparse rewards.

Keywords: reinforcement learning; self-generating evaluations; self-generating of rewards; sensor-based innate mechanism; generation of pleasure and discomfort

Citation: Sakamoto, Y.; Kurashige, K. Self-Generating Evaluations for Robot's Autonomy Based on Sensor Input. *Machines* **2023**, *11*, 892. https://doi.org/10.3390/machines11090892

Academic Editor: Dan Zhang

Received: 7 July 2023
Revised: 4 September 2023
Accepted: 5 September 2023
Published: 6 September 2023

1. Introduction

In recent years, robots have become increasingly popular [1,2]. Unmanned robots that can perform steadily and act decisively to reduce the risk of human injury during rescue operations are high in demand, particularly in complex and rapidly changing environments, such as disaster sites [3,4]. Therefore, robots must be capable of operating in complex and changeable environments as well as in simple ones. These robots are also at risk of failure; therefore, they must identify and select the optimal behavior in their environment to avoid failure. Studies have been conducted on the autonomous adaptation of robots and electronic devices to their environment [5,6]. These methods are based on the senses of organisms operating in complex environments. Within this context, studies have explored the use of reinforcement learning to learn more optimal behaviors through trial and error in response to the environment. In reinforcement learning [7–9], the environment offers rewards for the actions of the robot. A reward is a value that indicates the task accomplishment for an action. Reinforcement learning typically requires designing appropriate rewards by assuming the characteristics of the environment and tasks in advance [10,11]. When an environment or task changes, the relationships between the environment, robot actions, and rewards must also be redesigned [12]. This imposes a considerable burden on designers. Inverse reinforcement learning and internal reward learning are traditional methods that can reduce design burden. Inverse reinforcement learning uses learning optimization metrics based on the behavior of skilled users [13,14]. Internal rewards are generated based on internal changes in the robot [15–17], which are separate from the external rewards derived from the environment.

We have explored self-generating evaluations (SGE) based on sensor trends to reduce the burden of reward design in reinforcement learning [18]. SGE is an internal reward

model in which an agent generates rewards based on its sensor base. The SGE has a value for habituation, which is considered a stimulus threshold. The SGE changes its stimulus threshold to adapt to its environment. However, if the degree of adaptation to the stimuli does not change appropriately, the robot may learn harmful behavior. In this study, we propose a method for adjusting the adaptation speed of a threshold according to the strength and degree of the input property deviations. This method reduces the degree of adaptation to harmful stimuli and allows the robot to learn risk-averse behavior in a risky environment. Simulation experiments were conducted in two environments, one with path learning and the other with sparse rewards, to compare the proposed method with an existing method [19].

2. Existing Method on SGE

SGE is based on the learning process of an organism in a real environment [20]. Organisms can independently determine whether their current state is good or bad based on external stimuli, even if they cannot receive evaluations or rewards from the outside world. Sensations to stimuli such as pain are particularly unpleasant because they threaten vital activities [21]. Therefore, the organisms continue to work by learning to avoid such stimuli and dangers. In view of the above requirements, the robots must remain active while performing many tasks. Further, the robots must avoid dangerous stimuli. Therefore, we considered robots that can adapt to a wide variety of environments by generating rewards for environmental stimuli using their onboard sensors. SGE differs slightly from internal rewards, which are rewards generated based on the internal state of the agent [22], such as curiosity, whereas SGE generates rewards based on inputs from the sensors. Therefore, SGE does not depend on specific environments or tasks. The robot is instructed to generate a reward for reinforcement based on the SGE-generated evaluation values. This reduces the burden on designers because they no longer need to redesign the reward function according to the environment and tasks. Currently, SGE has three evaluation indices that are unique and versatile and are used in different environments based on the sense of discomfort of an organism. Sensor evaluation is based on "strength of input", "predictability of input", and "time with no input". For each sensor input, each of these indicators is calculated as a value between zero and one. Firstly, we explain the first index, namely, the strength of the input. An organism responds to a stimulus that is larger than a certain threshold. The larger the stimulus, the greater the potential threat, and consequently, the more pain or discomfort the organism experiences. Robots are particularly susceptible to failure when they are subjected to strong forces. Therefore, we assume that the robot experiences discomfort due to its large input. Next, we explain the second index, namely, the predictability of the input. The robot predicts the next sensor input it receives and provides a lower evaluation when the deviation from the predicted input is greater. For example, when the sensory input is stable after being touched, an animal, such as a housecat, may not feel dissatisfied. However, if the input is unstable, it can behave in a difficult or unpredictable manner. Conversely, when a person is unexpectedly verbally addressed from behind, they may feel surprised or stressed because they are momentarily unable to predict the next stimulus they anticipate. We consider the possibility that unpredictable inputs and environmental changes, even in robots, can lead to harmful situations because the robot may be unable to adequately process the input. Therefore, it is assumed that the robot experiences discomfort when exposed to unpredictable inputs. Finally, we explain the third index, namely, the time with no input, which denotes the state when the external input is so small that it cannot be sensed or when the sensors fail, causing the system to be unable to pick up any input. Organisms are exposed to environmental stimuli. Studies have shown that living organisms typically experience mental disorders after several days in environments in which they are deprived of sensory input [23,24]. Because a robot cannot distinguish between sensor failure and lack of input from the environment, we believe that a prolonged period without sensor input is harmful to the robot because it continues to be unaware of

its current environment. Therefore, we consider a robot that experiences discomfort due to the absence of input over a long period of time.

2.1. Evaluation Indices

We now explain the calculation of each evaluation index in SGE. First, we describe the evaluation frequency of SGE. SGE evaluates each input using evaluation indices. In addition, SGE considers the fact that a sensor receives multiple inputs within a single action at time t. Therefore, if the sensor receives n inputs within one action at time t, SGE evaluates the input n times.

Next, the evaluation value $E_{S\,i}(t_n)$ for the strength of the n^{th} input to sensor i at time t is defined by Equation (1). It is represented by a sigmoid function in terms of the mean of the input values $\mu_i(t_n)$, the maximum value of the input variate max_i, the constant N_i, and the variable $\delta_i(t_n)$. The size of the constant N_i determines the change in the evaluation value $E_{S\,i}(t_n)$. A sensor with an appropriate range was mounted on the robot. This sensor was deemed an appropriate one because it fully considers the threat to the recognition and action abilities of the robot. Therefore, the maximum value of the selected sensor was assumed to be max_i. The constant N_i was also designed in advance because the appropriate value varies depending on the sensor and device.

$$E_{S\,i}(t_n) = \left[1 + exp\left\{\frac{\mu_i(t_n) - \left(\frac{max_i + \delta_i(t_n)}{2}\right)}{N_i(max_i - \delta_i(t_n))}\right\}\right]^{-1} \tag{1}$$

The mean of the input values $\mu_i(t_n)$ and variable $\delta_i(t_n)$ are used in Equation (1). As SGE is designed for real machines, multiple inputs are received within a single action. Equation (2) yields the mean value $\mu_i(t_n)$ of the n^{th} input to sensor i at time t. The value $\mu_i(t_n)$ is used to treat multiple input information simultaneously as inputs in a single action. In addition, averaging is performed to reduce the effect of noise on each input.

$$\mu_i(t_n) = \frac{1}{n}\sum_{j=1}^{n} input_i(t_j) \tag{2}$$

The variable $\delta_i(t_n)$ is the threshold value for the strength of the input, which changes with the sensor inputs from the environment and denotes the adaptation of the senses in organisms. Habituation in organisms is the process of adapting to stimuli by gradually becoming unresponsive to repeated stimuli through repeated exposure. However, for salient stimuli, habituation is likely to make the response stronger rather than slower. In other words, the threshold $\delta_i(t_n)$ on which the evaluation is based changes with the input from the environment, allowing for an autonomous evaluation appropriate to the environment. We define the threshold $\delta_i(t_n)$ as an adaptation. Equation (3) describes the adaptation $\delta_i(t_n)$ for each time step wherein data are received from the sensor. β_i is the speed of adaptation to the input. Because β_i is constant, $\delta_i(t_n)$ changes at the same speed for all inputs.

$$\delta_i(t_n) = \delta_i(t_{n-1}) + \beta_i\{\mu_i(t_n) - \delta_i(t_{n-1})\} \tag{3}$$

Next, we explain how to calculate the evaluated values for the predictability of the input are defined in Equations (4)–(6). The prediction equation (Equation (4)) by generating prediction using support vector regression (SVR) [25,26] is used to generate the n^{th} predictability $f_i(t_n)$ during an action at time t_n. The input value of SVR is a constant number of input up to just before time t, which is $t_1, t_2, \ldots, t_{n-1}$ of sensor i. Calculating the prediction errors over the entire time series allows agents to consider stimuli based on their experiences in the environment. Therefore, when the environment does not change significantly, the prediction error can be reduced, and high ratings can be calculated. In Equation (4), ω is a one-dimensional coefficient vector, ϕ is the mapping function, and b is a bias term. Next, the prediction error $D_i(t_n)$ between the predicted and input values at

time t_n is calculated using Equation (5). Finally, the prediction error $D_i(t_n)$ and constant S_i are used to calculate the evaluation value $E_{P\,i}(t_n)$, as expreesed in Equation (6). The evaluation value $E_{P\,i}(t_n)$ is higher for smaller prediction errors $D_i(t_n)$ and lower for larger prediction errors $D_i(t_n)$.

$$f_i(t_n) = \omega^T \phi(t_n) + b \tag{4}$$

$$D_i(t_n) = |input_i - f_i(t_n)| \tag{5}$$

$$E_{P\,i}(t_n) = exp\left\{ \frac{-D_i(t_n)^2}{S_i} \right\} \tag{6}$$

Finally, we describe the calculation of the evaluation index for a time step without any sensor input. The evaluation is performed using Equation (7) when the agent does not clearly know whether the sensor is malfunctioning or inactive with no stimulus. The evaluation value $E_{T\,i}(t_n)$ for the n^{th} input value to sensor i at time t is calculated using the variable $d_i(t_n)$ and constant k_i, as shown below.

$$E_{T\,i}(t_n) = exp\left\{ \frac{-d_i(t_n)}{k_i} \right\} \tag{7}$$

Equation (8) defines the variable $d_i(t_n)$ when no input is detected. By setting γ_i, the variable $d_i(t_n)$ is set to decay with time. The evaluation value $E_{T\,i}(t_n)$ decreases as the condition of no input persists, that is, as $d_i(t_n)$ increases.

$$d_i(t_n) = \begin{cases} \gamma_i d_i(t_{n-1}) & (input_i(t_n) \neq 0) \\ d_i(t_{n-1}) - 1 & (input_i(t_n) = 0) \end{cases} \tag{8}$$

2.2. Integration of Evaluation

This section describes the integration of the evaluations. The sensor inputs are evaluated using the three evaluation indices. Only a single value can be used for sensor i at a given time because that value corresponds to the evaluation value of sensor i. Hence, the evaluation values calculated using the three types of evaluation indices should be integrated. Additionally, evaluation indicators that calculate high evaluation values are actively excluded to emphasize the danger from the sensor information. Equation (9) defines the evaluated value of $E_i(t_n)$ using the values calculated in Equations (1), (6) and (7). Equation (10) defines ω_S, ω_P, and ω_T, which are the weights of the values of the evaluation indices. The weight of the evaluation indicator x is 0 if its value is higher than 0.5 and 1 if its value is lower than 0. Therefore, Equation (10) excludes the higher values when calculating the danger-weighted assessment. Equation (11) yields the sum of the weights and values of the power roots in Equation (9). When the value of the power root is $m \neq 0$, the evaluation is performed using Equation (9). In the condition $m = 0$, in which the evaluation values for all three indices are higher than the reference value, sensor evaluation is not used because it emphasizes danger.

$$E_i(t_n) = \begin{cases} \sqrt[m]{E_{S\,i}(t_n)^{\omega_s}} + \sqrt[m]{E_{P\,i}(t_n)^{\omega_P}} + \sqrt[m]{E_{T\,i}(t_n)^{\omega_T}} & (m \neq 0) \\ 0 & (m = 0) \end{cases} \tag{9}$$

$$\omega_x = \begin{cases} 1 (E_{x\,i} < 0.5) \\ 0 (E_{x\,i} \geq 0.5) \end{cases} \tag{10}$$

$$m = \omega_S + \omega_P + \omega_T \tag{11}$$

2.3. Reward Generation

SGE considers that a sensor receives multiple inputs within a single action; therefore, the condition under which a single action occurs differs from that under which a single

input is received. Hence, sensor i must be evaluated to generate a reward for a single action. The rating E_i for an action is updated as the rating M_{E_i}, and a rating for that action is generated, as expressed in Equation (12). The values are updated using the discount factor γ_e for a single input. The updated evaluation value M_{E_i} is initialized as zero at the start of the action.

$$M_{E_i} \leftarrow (1 - \gamma_e) M_{E_i} + \gamma_e E_i(t_n) \tag{12}$$

The rewards generated by SGE are for a single action, which is similar to conventional reinforcement learning. Therefore, the reward generated by SGE is calculated based on the valuation M_{E_i} of one action at time t. Thus, the range of the evaluation value is $0 \sim 1$. However, to accurately calculate the reward for the evaluation value, the reward range is normalized to $-1 \sim 1$. Equation (13) is used to calculate the reward as follows:

$$r = 2M_{E_i} - 1 \tag{13}$$

3. Adaptation by Considering the Input Properties

This study proposes a method that considers the properties of the input for the value of adaptation in the evaluation index of the input strength to update the adaptation speed of the input strength. We used the danger of strength and the degree of deviation as the input properties. We define the danger of strength as the degree of failure caused by the input and the degree of deviation as the difference between the values of adaptation. The stronger the sensor input, the higher the danger. This is because it is more likely to cause the robot to malfunction. Thus, the more harmful the sensor input, the more carefully it must adapt to it to avoid danger. The extent to which the received sensor input values should be considered must also be determined. The adaptation value in SGE approaches the mean value of the inputs from the environment obtained through the sensor. The greater the difference between the sensor input and adaptation value, the more likely it is that the sensor input received is an infrequent input from the environment. Therefore, the adaptation value is updated in a manner that minimizes the impact of sensor inputs that are rarely received.

Figure 1 shows a flowchart depicting the calculation of the input strength using this method. First, the robot receives a stimulus from the environment at sensor i. The output values of sensor i are input to evaluate the strength, adaptation, and adaptation speed. Next, the adaptation speed v_i is calculated using the input value of sensor i. Subsequently, the value of adaptation δ_i is calculated using the input value of sensor i and the adaptation speed. Finally, the strength evaluation value of the sensor input $E_{S\,i}(t_n)$ is calculated using the input value of sensor i and the adaptation value. This process is performed each time an input is received.

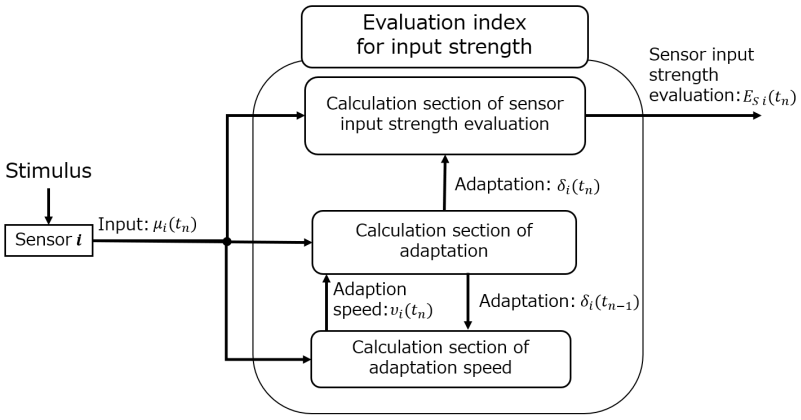


Figure 1. Flowchart depicting an evaluation of the strength of inputs using the proposed method.

The same weighted mean as that used in the adaptation formula of the previous method is used in Equation (14). As depicted in Equation (14), the adaptation time for an input value varies with the adaptation speed v_i. The adaptation speed v_i is updated using Equation (15) each time the sensor receives an input. The adaptation speed is calculated using the mean value of the sensor input μ_i and the difference between the mean value of the sensor input μ_i and adaptation value δ_i. The adaptation speed v_i is smaller when the difference between the mean value of the sensor input μ_i and the adaptation value δ_i is higher. Therefore, according to Equation (14), the smaller the adaptation speed v_i, the smaller the change in the adaptation value δ_i. Thus, careful adaptation can be applied to the sensor input.

$$\delta_i(t_n) = \delta_i(t_{n-1}) + v_i(t_n)\{\mu_i(t_n) - \delta_i(t_{n-1})\} \tag{14}$$

The c is the maximum value calculated by Equation (15). The properties pertaining to the adaptation of the robot transform depending on the value of c. Therefore, when its value is high, becoming accustomed to the strength of the sensor input is relatively easy, and vice versa. The maximum value of the input variate max_i and the mean value of the sensor input μ_i are associated with the strength of the sensor input. The higher the mean value of the sensor input μ_i, the smaller the parameter representing the strength of the sensor input. The strength of the sensor input varies for each sensor. For a typical evaluation, we normalize the value in the range 0–1. The values corresponding to the variables max_i, $\mu_i(t_i)$, and $\delta_i(t_{n-1})$ represent the rarity of the sensor input. Sensor rarity is an input with a high probability of being an outlier with a significant deviation from the adaptation. The value decreases as the difference between the mean of the sensor input μ_i and adaptation value δ_i increases. In other words, the narrower the sensor input in the environment, the smaller the value. This value is also normalized to 0–1 because the strength of the sensor input varies for each sensor.

$$v_i(t_n) = c\frac{\{max_i - \mu_i(t_n)\}\{max_i - |\mu_i(t_n) - \delta_i(t_{n-1})|\}}{{max_i}^2} \tag{15}$$

4. Experiments

4.1. Simulation Experiments with Path Learning

The purpose of this experiment was to verify whether the proposed method could be used to recognize and avoid dangers more effectively. We compared the results obtained using the previous and proposed methods. The learning paths of each robot and the transition of the evaluation index for the strength of the sensor input for each robot were compared.

4.1.1. Experimental Setup

In this experiment, we performed a simulation to learn a path on a grid map, as shown in Figure 2. The robot was asked to learn a path from the starting point to the goal point. The robot can move one square per action and perform four types of actions: up, down, left, and right motions. The time and energy required for all actions performed by the robot were the same. The robot was equipped with a temperature sensor to detect the temperature, a collision sensor to identify wall collisions, and a position sensor to identify its position. If a robot performs an action that causes a wall collision, it returns to its original square. A high-temperature area is harmful because it increases the internal temperature of the robot, causing an internal breakdown. In this experiment, we set the temperature above 70 °C as dangerous. Therefore, the robot must learn paths to avoid hazardous/hot areas. The experiment was conducted ten times in the same setting for each of the previous and proposed methods.

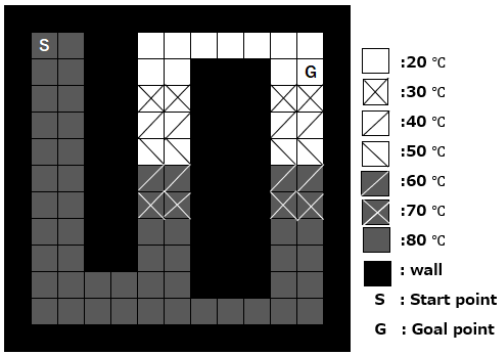


Figure 2. Grid map used in the experiment 1.

Table 1 lists the environmental settings for this experiment, and Table 2 lists the experimental settings of the robot. The temperature sensor received seven different inputs corresponding to the outside temperature from 20 °C (lower bound) to 80 °C (upper bound) every 10 °C.

Table 1. Environmental settings for the experiment 1.

The number of trials	1000
The number of actions per trial	200
Learning method	Q-learning
Learning rate in Q-learning α	0.3
Discount rate in Q-learning γ	0.99
The action selection method	ε-greedy
ε in the ε-greedy method	0.01
Goal reward	1
Reward for goal failure per trial	-1
Reward for colliding with a wall	-1
Reward per action	-0.005

Table 2. Experimental settings of the robot for the experiment 1.

The maximum value of the temperature sensor	100
The minimum value of the temperature sensor	0
Parameter N	0.08
k_i in the evaluation for the time with no input	250
γ_i in the evaluation for the time with no input	0.99
Parameter of the proposed method c	0.001
Parameter of the proposed method β	0.001
Input for every action taken	100

4.1.2. Experimental Results and Discussion

We determined the mean values and standard errors of the evaluation index for the strength of the sensor input in the experiment. The results of the previous and proposed methods for the 1000th trial of the experiment are shown in Figure 3. The learning paths of the agents obtained using the previous and proposed methods are shown in Figures 4 and 5, respectively. The paths shown are learning paths for which the greedy algorithm is run after completing all the trials and are representative of the learning paths for each method. In the case of the previous method, the agent learned the path through the high-temperature area (Figure 4a) eight out of ten times and through the area of normal-temperature (Figure 4b) two out of ten times. In the case of the proposed method, the agent learned the path through the normal temperature area (Figure 5) ten out of ten times.

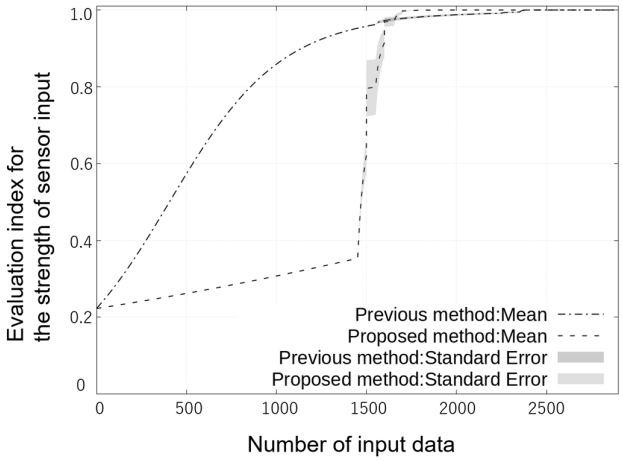
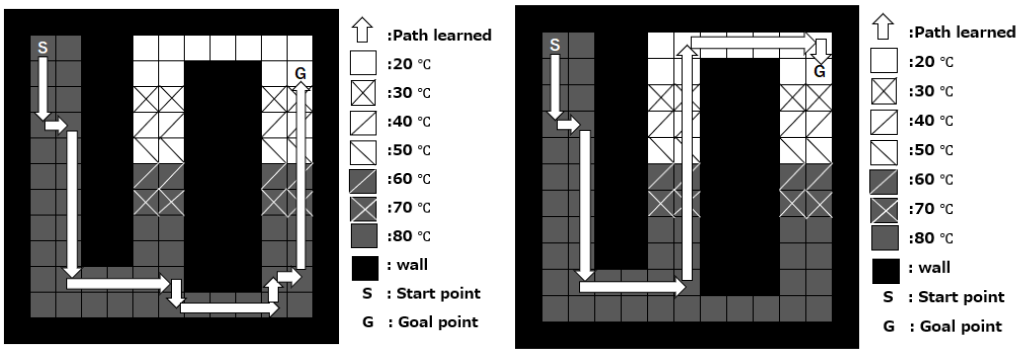


Figure 3. Evaluation of the strength of the sensor input for the previous and proposed methods for the 1000th trial in the experiment.



(**a**) (**b**)

Figure 4. (**a**) Path learned using the previous method (eight out of ten times); (**b**) path learned using the previous method (two out of ten times).

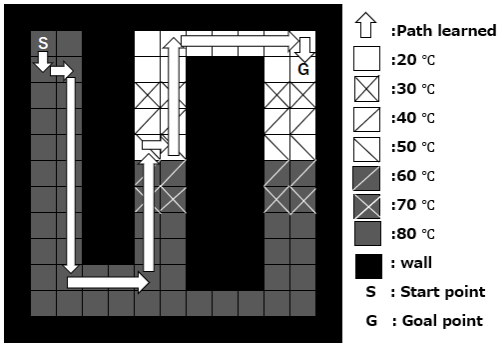


Figure 5. Path learned using the proposed method (ten out of ten times).

In the previous method, the robots exhibited a high probability with respect to passing through the high-temperature areas (80 °C), although high-temperature areas were harmful to the robots. This is because it is difficult for the robot to identify the areas as dangerous due to the high value of adaptation to high temperatures in the danger area. Although the paths through the areas with high and normal temperatures (20 °C) are learned for evaluating strength of the sensor input, the small standard error for the previous method suggests that the same evaluations are made in the danger and safe areas corresponding to high and normal-temperatures, respectively. Thus, the robot learns two distinct paths.

The robot in the proposed method exhibited a slower adaptation to inputs from high-temperature areas. Therefore, the value of the evaluation index for the strength of the sensor input from the high-temperature area is low, and the robot continues to identify danger. The values of the evaluation index differ between the high-temperature and normal-temperature areas. We believe that the robot can learn a safer route with a higher evaluation. In addition, the standard error of the robot for the proposed method is high because the correct behavior for approaching the goal varies between the experiments. Therefore, the standard error increases because the input strengths are evaluated differently. Thus, the robot can learn a safer path rather than the safest path.

4.2. Simulation Experiments with Sparse External Rewards

The purpose of the experiment described in this section was to verify that the proposed method improved the ability to identify harmful areas, even in an environment with few external rewards. The proposed method allows the adaptation speed to vary with sensor input and the evaluation index for the strength of the sensor input to be appropriate for the environment. Therefore, it is expected to increase the number of actions in safe areas, thereby stabilizing the robot activities. We compared the proportion of visits by each robot obtained using the previous method with the that of the proposed method for each coordinate in the environment.

4.2.1. Experimental Setup

In this experiment, we performed a simulation in which the robots remained active on a grid map, as shown in Figure 6. There are seven types of stimuli in the environment, some of which are dangerous to the robots. Therefore, the robots should avoid dangerous environments and continue to operate in safe environments. The robot can move one square per action and perform four different types of actions: up, down, left, and right motions. The time and energy required for all of the actions performed by the robot were the same. The robot was equipped with an array of sensors to receive inputs from the environment. The robot was equipped with temperature, collision, and positional sensors, as was the case before. If the robot performs an action that causes a wall collision, it returns to its the original square. The robot returns to the starting point and performs the next trial when the number of actions in a trial is met or when it becomes inactive.

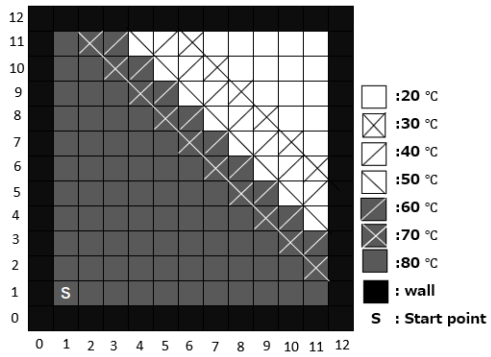


Figure 6. Grid map used in the experiment 2.

Table 3 lists the environmental settings for this experiment, and Table 4 lists the experimental settings of the robot.

The temperature sensor receives seven different inputs corresponding to the outside temperature from 20 °C (lower bound) to 80 °C (upper bound) every 10 °C.

Robots are more likely to become inactive when their internal temperature increases because of environmental temperature. Therefore, we set the agent to become inactive if it continued to receive certain inputs from the environment. The inactivity points that increase for each action with input from the environment at 60 °C, 70 °C, and 80 °C are

defined by Equation (16). The decision to deactivate the agent was made on a per-action basis. The initial value of the probability of agent inactivity in each trial was zero.

$$\text{Inactivity point} = \begin{cases} 5.0 \times 10^{-7} (80\ ^\circ\text{C}) \\ 1.0 \times 10^{-7} (70\ ^\circ\text{C}) \\ 0.5 \times 10^{-7} (60\ ^\circ\text{C}) \end{cases} \quad (16)$$

Table 3. Environmental settings for the experiment 2.

The number of trials	2000
The number of actions per trial	200
Learning method	Q-learning
Learning rate in Q-learning α	0.3
Discount rate in Q-learning γ	0.99
The action selection method	ε-greedy
ε in the ε-greedy method	0.01
Reward for continued 200 actions	1
The reward of inactivity	-1
Reward for colliding with a wall	-1

Table 4. Experimental settings of the robot for the experiment 2.

The maximum value of the temperature sensor	100
The minimum value of the temperature sensor	0
Parameter N	0.08
k_i in the evaluation for the time with no input	250
γ_i in the evaluation for the time with no input	0.99
Parameter of the proposed method c	0.001
Parameter of the proposed method β	0.001
Input for action taken	100

4.2.2. Experimental Results and Discussion

We compared the extent to which learning from the previous and proposed methods changed the percentage of robot visits to the environment. Figure 7 shows the percentage of robots visits to the environment from the first trial to the 500th trial (with insufficient learning) and the percentage of robots visits to the environment from the 1501st trial to the 2000th trial (with sufficient learning). The results of the previous method are shown in Figure 7. Figure 7a shows the percentage of visits made using the previous method by the robot with insufficient learning, and Figure 7b shows the percentage of visits made using the previous method by the robot with sufficient learning. The results corresponding to the proposed method are shown in Figure 8. Figure 8a,b show the percentage of visits by the robot with insufficient and sufficient learning, respectively.

The robots used in the previous method visited the entire environment during the initial stages of the experiment. The percentage of robot visits to the upper-right safe area to avoid danger in the final phase of the experiment did not differ significantly from that in the earlier phase, demonstrating that they affected the environment. We assume that the robot can easily adapt to any input; hence, only a slight difference exists between the values of the evaluation indices for the potentially dangerous high-temperature areas (70 °C and 80 °C) and those obtained from normal-temperature areas (20 °C and 30 °C). Furthermore, no difference was observed in the generated reward. We believe that the robot learns about the environment by recognizing that the inputs from the high- and normal-temperature areas are equivalent.

We found that the robots using the proposed method visited a higher proportion of the environment on the upper-right side, a safer area. In the final phase of the experiment, visitation rates to the high-temperature areas of 70 °C and 80 °C were lower, and the percentage of visits to the safe environment in the upper-right was higher. We believe that

this is because the rewards generated by the self-generation of evaluations allow them to learn and continue to operate in safer areas, even in an environment where external rewards are sparse. Thus, we assume that the robot learns by showing a lower evaluation of the sensor inputs in areas with higher temperatures than in the normal-temperature areas.

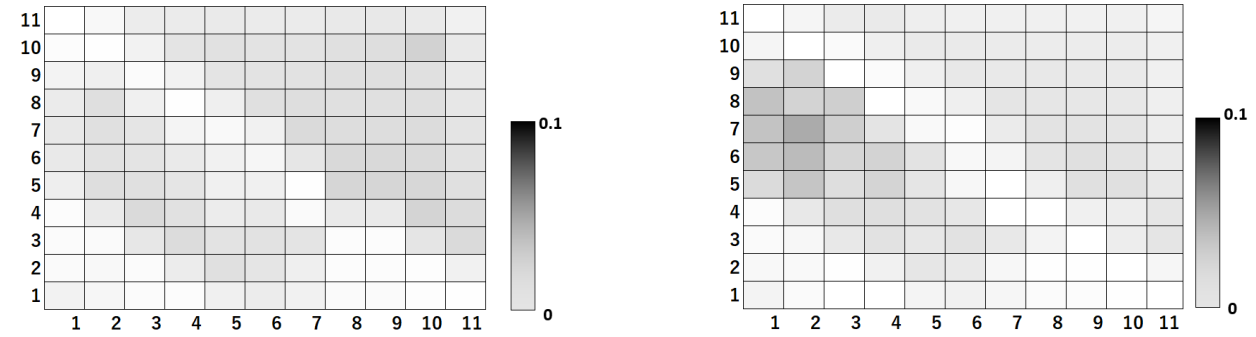


Figure 7. (**a**) Percentage of robot visits to the environment from the first trial to the 500th trial in the previous method; (**b**) percentage of robot visits to the environment from the 1501st trial to the 2000th trial in the previous method.

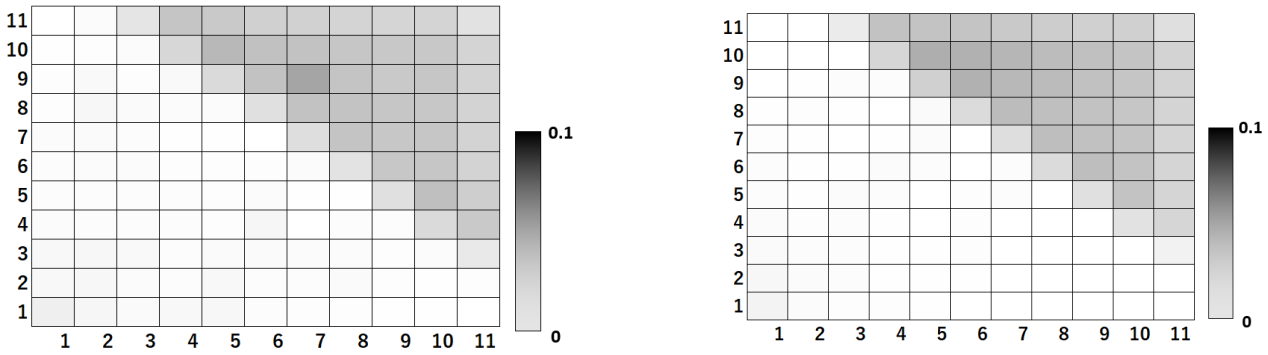


Figure 8. (**a**) Percentage of robot visits to the environment from the first trial to the 500th trial in the proposed method; (**b**) percentage of robot visits to the environment from the 1501st trial to the 2000th trial in the proposed method.

5. Conclusions

This study focused on the issue of robots learning potentially harmful behavior when the adaptation speed of the adaptation value, which is the threshold for the strength of the sensor input, is the same for all sensor input values. To address this issue, we varied the adaptation speed to consider the danger and degree of deviation for different properties of the sensor input. This is expected to improve the robot's adaptation to the environment by carefully changing harmful sensor inputs. The results of our simulation experiments on path learning in an environment with safe and harmful paths and learning in an environment with sparse rewards confirm that the proposed method enables the robot to learn actions to avoid harmful environments with high failure rates.

In the future, we intend to allow the robot to distinguish and evaluate sensor inputs that are lower than the threshold for the strength of the sensor input to evaluate safer sensor inputs. SGE yields the same evaluation for all sensor input values below the threshold for the strength of the sensor input. Therefore, learning safer behavior is difficult because the evaluation cannot distinguish between larger and smaller sensor inputs within a sensor input value lower than the threshold strength of the sensor input. By changing the evaluation equation for each strength region using a piecewise linear function, it is possible

to distinguish between large and small strengths with sensor input values smaller than the strength threshold of the sensor input. Therefore, we believe that safer actions can be learned, and the accuracy of danger-evading actions can be improved. In addition, we intend to consider generic evaluation indices that can be applied to various sensors and environments and conduct simulation experiments and resultant data collection that more closely resemble real-world environments to demonstrate the utility of SGE.

Author Contributions: Conceptualization, K.K.; methodology, K.K. and Y.S.; validation, Y.S.; formal analysis, Y.S.; investigation, Y.S.; resources, K.K.; data curation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S.; visualization, Y.S.; supervision, K.K.; project administration, K.K.; funding acquisition, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Z.; Barenji, A.V.; Jiang, J.; Zhong, R.Y.; Xu, G. A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *J. Intell. Manuf.* **2020**, *31*, 469–480. [CrossRef]
2. Matheson, E.; Minto, R.; Zampieri, E.G.G.; Faccio, M.; Rosati, G. Human–Robot Collaboration in Manufacturing Applications: A Review. *Robotics* **2019**, *8*, 100. [CrossRef]
3. Zhang, Q.; Zhao, W.; Chu, S.; Wang, L.; Fu, J.; Yang, J.; Gao, B. Research progress of nuclear emergency response robot. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *452*, 042102. [CrossRef]
4. Li, F.; Hou, S.; Bu, C.; Qu, B. Robots for the urban earthquake environment. *Disaster Med. Public Health Prep.* **2022**, *17*, 181. [CrossRef] [PubMed]
5. He, Z.; Ye, D.; Liu, L.; Di, C.A.; Zhu, D. Advances in materials and devices for mimicking sensory adaptation. *Mater. Horiz.* **2022**, *9*, 147–163. [CrossRef] [PubMed]
6. Graczyk, E.L.; Delhaye, B.P.; Schiefer, M.A.; Bensmaia, S.J.; Tyler, D.J. Sensory adaptation to electrical stimulation of the somatosensory nerves. *J. Neural Eng.* **2018**, *15*, 046002. [CrossRef] [PubMed]
7. Sutton, R.S.; Barto, A.G. *Reinforcement Learning*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
8. Zhu, H.; Yu, J.; Gupta, A.; Shah, D.; Hartikainen, K.; Singh, A.; Kumar, V.; Levine, S. The Ingredients of Real-World Robotic Reinforcement Learning. International Conference on Learning Representations. *arXiv* **2020**, arXiv:2004.12570.
9. Akalin, N.; Loutfi, A. Reinforcement Learning Approaches in Social Robotics. *Sensors* **2021**, *21*, 1292. [CrossRef] [PubMed]
10. Kuhnle, A.; Kaiser, J.-P.; Theiß, F.; Stricker, N.; Lanza, G. Designing an adaptive production control system using reinforcement learning. *J. Intell. Manuf.* **2021**, *32*, 855–876. [CrossRef]
11. Eschmann, J. Reward function design in reinforcement learning. In *Reinforcement Learning Algorithms: Analysis and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 25–33.
12. Everitt, T.; Hutter, M.; Kumar, R.; Krakovna, V. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese* **2021**, *198*, 6435–6467. [CrossRef]
13. Fu, J.; Korattikara, A.; Levine, S.; Guadarrama, S. From language to goals: Inverse reinforcement learning for vision-based instruction following. *arXiv* **2019**, arXiv:1902.07742.
14. Arora, S.; Doshi, P. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artif. Intell.* **2021**, *297*, 103500. [CrossRef]
15. Chentanez, N.; Barto, A.; Singh, S. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2004; Volume 17.
16. Aubret, A.; Matignon, L.; Hassas, S. A survey on intrinsic motivation in reinforcement learning. *arXiv* **2019**, arXiv:1908.06976.
17. Colas, C.; Fournier, P.; Chetouani, M.; Sigaud, O.; Oudeyer, P.Y. Curious: Intrinsically motivated modular multi-goal reinforcement learning. In Proceedings of the International Conference on Machine Learning, Beijing China, 3–7 November 2019; pp. 1331–1340.
18. Hakim, A.A.B.M.N.; Fukuzawa, K.; Kurashige, K. Proposal of Time-based evaluation for Universal Sensor Evaluation Index in Self-generation of Reward. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 1161–1166. [CrossRef]
19. Ono, Y.; Kurashige, K.; Hakim, A.A.B.M.N.; Kondo, S.; Fukuzawa, K. Proposal of Self-generation of Reward for danger avoidance by disregarding specific situations. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Online, 5–7 December 2021; pp. 1–6. [CrossRef]
20. Kurashige, K.; Nikaido, K. Self-Generation of Reward by Moderate-Based Index for Senor Inputs. *J. Robot. Mechatron.* **2015**, *27*, 57–63. [CrossRef]

21. Watanabe, M.; Narita, M. Brain Reward Circuit and Pain. In *Advances in Pain Research: Mechanisms and Modulation of Chronic Pain*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 201–210.
22. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-Driven Exploration by Self-Supervised Prediction. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 2778–2787. [CrossRef]
23. Sugimoto, S. The Effect of Prolonged Lack of Sensory Stimulation upon Human Behavior. *Philosophy* **1967**, *50*, 361–374.
24. Sugimoto, S. Human mental processes under sensory restriction environment. *Jpn. J. Soc. Psychol.* **1985**, *1*, 27–34.
25. Zhong, H.; Wang, J.; Jia, H.; Mu, Y.; Lv, S. Vector field-based support vector regression for building energy consumption prediction. *Appl. Energy* **2019**, *242*, 403–414. [CrossRef]
26. Quan, Q.; Zou, H.; Huang, X.; Lei, J. Research on water temperature prediction based on improved support vector regression. *Neural Comput. Appl.* **2020**, *34*, 8501–8510. [CrossRef]