Environmental Adaptation for Wheel-legged Robot Using External Force Given to Legs

Yuki Nishimura† and Sadayoshi Mikami²

†Department of Complex and Intelligent Systems, Future University Hakodate, Hokkaido, Japan
(Tel : +81-80-1870-1694; E-mail: g2116037@fun.ac.jp)
²Department of Complex and Intelligent Systems, Future University Hakodate, Hokkaido, Japan
(Tel : +81-138-34-6561; E-mail: s_mikami@fun.ac.jp)

Abstract: In recent years, autonomous mobile robots working in unknown environments are required to select apposite actions depending on their environments by themselves. In this research, we develop a system for a wheel-legged robot which grasps the situation of surrounding environment from the transition of its internal sensor state and selects the action to escape from the situation where it cannot move by itself. In our previous research, the effectiveness of the proposed system was demonstrated by the experiments with a simulator using physics engine. However, it was not shown whether the proposed system is also effective in actual environments. In this paper, we show the effectiveness of the proposed system by the experiments with a real robot and online learning in actual environments. The results of the series of experiments show that the system converges into appropriate actions in sufficiently short learning period, which means that the proposed system is effective in the actual environment.

Keywords: Autonomous systems, Environmental adaptation, Wheel-legged robot, Internal state, Reinforcement learning

1. INTRODUCTION

1.1 Background

Currently, autonomous mobile robots that run in unknown environments are actively investigated. These robots equip with the moving mechanisms having high running performances. Although these robots use such a mechanism to move, they still have difficulties in continuously running in many situations. For instance, the robot has a risk that it cannot run by getting in contact with obstacles or sinking in sand. In such cases, the situation demands human operators to help the robot to escape by remote control. However, such a real-time communication is often unavailable, or it may be difficult that human operators understand what are happening around the robot and give apposite commands. Therefore, it should be necessary to propose a method for autonomous mobile robots to escape from the situation that they cannot move by themselves.

1.2 Related works

Environmental adaptation for mobile robots generally consists of two steps: the first step is to identify the environment and the second step is to plan robot’s actions. First, a robot agent (controller) identifies the environment. The agent grasps a state of the robot's surrounding environment by using cameras or sensors. Then, the agent plans robot's actions based on the environmental data (e.g., map data or other environmental information) that is identified in the first step. However, many of conventional methods have problems included in either step.

Most general environmental identification methods include an image-based method with cameras or vision systems [1, 2]. Camera images are useful to identify the surrounding environment due to large amount of information included in these. For example, robots can recognize shapes of obstacles, features of terrains and so on by using this information. However, these methods have some drawbacks of credibility of cameras. Camera images are affected by lighting conditions or weather conditions. Hence, the systems cannot recognize obstacles or terrains when these conditions change. Lenser et al. [3] proposed a method to resolve this problem. This method detects environmental change automatically with an algorithm to adapt to the change of lighting conditions. Therefore, the system can identify the environment correctly even when lighting conditions change. However, even if such algorithms are used, physical problems of cameras cannot be resolved. For instance, accuracy of camera is decreased by dirt (e.g., dust or mud) on the camera lens. Also, for another example, camera images cannot accurately describe the situation in muddy water. Therefore, the image-based system cannot identify the state in the environment where robot legs sink in soft surfaces (e.g., shallow rivers or snow surfaces).

Many of sensor-based environmental identification methods use optical methods (e.g., LIDAR) [4, 5]. Optical sensor based methods can precisely measure and recognize objects (e.g., obstacles or walls). Therefore, robots can avoid obstacles more accurately. However, these methods also suffer from dirt or unclear environments like the methods with cameras.

To overcome the problems of optical sensors, the method using internal sensors is effective. Giguere et al. [6] proposed an environmental identification method with some kinds of internal sensors (i.e., accelerometers, gyroscopes, angle encoders and current estimators). This method can identify the environment stably since the internal sensors have strength against dirt. In addition, this method can adapt to complex environments by combining some of these internal sensor data. However, the system can acquire only actions corresponding to certain environments since it

† Yuki Nishimura is the presenter of this paper.
learns behavior in some environments supposed by authors. Therefore, the system cannot adapt to unknown and unexpected environments.

Internal sensors, such as force and touch sensors that measure forces applied on the body, are most suitable for environmental identification in harsh environments. But conventional internal sensor based methods did not give appropriate plans of robot’s actions for the situations recognized by the methods.

We propose a new method to resolve these problems by combining internal sensors and online learning.

1.3 Previous work

In our previous work [7], we proposed a concept of control method that uses force sensing to determine appropriate actions. In the proposed method, the robot grasps the surrounding environment from the transition of its leg torque (internal state) at first. Then, the robot learns actions to escape from the situation that it cannot move. Reinforcement learning is used in the proposed system as an algorithm to learn actions. This algorithm has often been used to acquire behavior for mobile robots where appropriate actions are not directly known by the sensing results [8]. Our previous work showed an effectiveness of the proposed method through the experiments with a simulator using physics engine [7]. However, its effectiveness in the actual environment had not been shown. For example, it was not clear how we should measure the torques applied on legs during moving, and whether it is able to improve performances in real environments.

1.4 Purpose of this paper

In this paper, we show that the proposed method is also effective in actual environments. For this purpose, the actual wheel-legged robot having the same structure as the previous simulations is shown at first. Also, the proposed system modified for the actual robot is described. Then, the configurations of experimental environments for rough terrain are explained. Finally, we show the experiments of robot agent for learning in the experimental environment. The results are analyzed and described in the last chapter.

2. WHEEL-LEGGED ROBOT

A wheel-legged robot has many advantages for running in rough terrain [9-11]. A wheel-legged robot has legs which have wheels attached at the end of these. By this structural characteristic, this type of robot has advantages of both a legged robot and a wheel robot. First, a legged robot is highly adaptive to various terrains by changing the height of its legs. On the other hand, a wheel robot can run faster than a legged robot on a smooth road. A wheel-legged robot can adapt to terrain and run faster on a smooth road. In addition, a wheel-legged robot has higher adaptation abilities by combining these moving mechanisms. For instance, a wheel-legged robot can hold its legs up while moving its wheels, which realizes smooth movement on a rough terrain.

However, choosing an appropriate pair of motions for a given terrain is very difficult. A snowy surface is a typical example. In the case of a soft snow, a wheel movement may cause that the robot's body sinks down by scraping the snow out. Instead, a leg-only movement may support the body floating up on the snow and move forward successfully. On the other hand, a hard snow may cause that the legs slip on its surface. In this case, a wheel-only movement may be better because it can push the surface firmly.

As described, a wheel-legged robot has high adaptation ability to terrain, but it has a problem of action selection. In this research, the system to solve this problem is shown.

Fig. 1 shows the wheel-legged robot developed for this research. This robot has 6 legs with 2 degrees of freedom, which allow the robot to perform standard gaits such as a tripod gait or a crawl gait. Wheels are mounted at the end of each leg, which are driven by motors and produce the force to push the body.

Fig. 1 Wheel-legged robot aimed at this research

3. SYSTEM ARCHITECTURE

3.1 Reinforcement learning with internal sensor state

In this research, the robot agent learns the appropriate escaping actions by using reinforcement learning. The agent observes external force given to the robot’s legs based on the transition of these torques (internal state), and learns the actions based on this internal state.

An overview of the learning flow is shown in Fig. 2. The state, the action and the reward represented in this figure are important elements of the reinforcement learning. The action value function, which represents the values of each action in every state, is used to select an action. An online learning method is used to update the action value function in order to learn in real time (i.e., the terminal state does not exist). The definitions of these reinforcement learning elements in this system are described in detail from the next section.
3.2 State identification by external force observation

A state is defined based on the leg torques. The leg torques are changed by external force which is caused by the interaction between the environment and the body motion. For example, the leg torque increases when the leg gets in contact with an obstacle as shown in Fig. 3 even though the robot tries to run forward. As in this example, the robot's surroundings situation can be recognized by the transition of the leg torques. In this paper, we only focus on the torques of front two legs. A torque value of each leg is measured as follows: (1) All the legs are moved to form a special posture as shown in Fig. 4 and stop in this posture. (2) Then, the front and rear four legs are driven forward by motors for a short while. (3) During the forward drive, the torques applied on the front two legs are measured from the motor current. The detail of this process is summarized as follows:

i. The robot moves the legs to the special posture to measure leg torques.
ii. Then, front and rear four wheels rotate slowly for a short while (approximately 200 milli-second).
iii. The agent records front two leg torques.
iv. The measurement is repeated 20 times at 10 milli-second intervals.
v. Finally, the agent calculates the average of the recorded torque values on each of the front legs.

In our method, electrical current passing through a wheel driving motor is measured as a leg torque. The current values at the front legs obtained through this procedure are used as a state value by discretizing with the thresholds as shown in Fig. 5. The number of states should be kept as small as possible for quick learning convergence. For this purpose, we checked possible range of the current in our hardware (from halt to free rotation) as the preparation. Then, we divided the current values in 6 ranges around the area where the value sharply changes according to the applied torque. The current values are measured in each of left and right legs. By this, the number of states is 36 in total, which is considered as small enough for online reinforcement learning.

3.3 Action

In this research, leg actions and wheel actions are defined individually, and the action is defined as
combinations of leg actions and wheel actions. For quick convergence, the number of leg and wheel actions should be kept as small as possible like the definition of the state.

First, leg actions are defined as a combination of some leg movements. We define 4 patterns of the leg actions as shown in Fig. 6. Then, wheel actions are defined as either all wheels are given rotation torque or not. Therefore, 2 action patterns are defined as wheel actions. Finally, the action is defined as combinations of these leg actions and wheel actions. In addition to these, a standing-still behavior is excepted from actions. Therefore, the number of actions is 7 in total as shown in Table 1. In this paper, an action is selected from these 7 actions based on the action value function by softmax action selection method where temperature parameter is set to 1.

![Fig.6 Leg Actions](image)

### Table 1 All action list

<table>
<thead>
<tr>
<th>Action number</th>
<th>Gait</th>
<th>Wheel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No movement</td>
<td>On</td>
</tr>
<tr>
<td>2</td>
<td>Crawl I</td>
<td>Off</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>On</td>
</tr>
<tr>
<td>4</td>
<td>Crawl II</td>
<td>Off</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>On</td>
</tr>
<tr>
<td>6</td>
<td>Tripod</td>
<td>Off</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>On</td>
</tr>
</tbody>
</table>

### 3.4 Reward

A reward is defined based on a transition of the torque value between before and after performing an action. The agent should be given a positive reward when the robot succeeds in escaping from stuck state (e.g., the situation as shown Fig. 3) as the result of performing an action. In this case, the measured torque value decreases. On the other hand, the agent should be given a negative reward when the robot fails to escape from stuck state as the result of performing an action. In this case, the torque value increases or does not change.

To satisfy these conditions, the reward at time $t+1$ is obtained by

$$ r_{t+1} = d \sum_{leg \in \{left, right\}} (\tau_{(leg,t)} - k \tau_{(leg,t+1)}) $$

where $r_{t+1}$ stands for a reward and $\tau_{(leg,t)}$ and $\tau_{(leg,t+1)}$ stands for torque values of each leg before and after performing an action. Also, $k$ is a constant to revise torque values after performing an action and $d$ is a constant to reduce large noisy rewards. In this paper, $k$ is set to 1.05 and $d$ is set to 0.1, which will also be applicable to many other system configurations.

### 3.5 Updating action value function

In this research, the action value function is updated by Q-learning algorithm for an online learning. Therefore, after performing an action and observing a state and getting a reward, the action value function is updated by

$$ \delta_{t+1} = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t), $$

$$ Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_{t+1}, $$

where $s_t$ stands for a state and $a_t$ stands for an action at time $t$ (i.e., before performing an action). Also, $s_{t+1}$ stands for a state and $r_{t+1}$ stands for a reward at time $t+1$ (i.e., after performing an action). In addition, $\alpha$ stands for the learning rate and is set to 0.1 and $\gamma$ stands for the discount factor and is set to 0.9. From our experiences, it is known that the small change of these learning factors does not affect the convergence on our system.

### 4. EXPERIMENTS

#### 4.1 Experimental settings

A series of experiments was carried out in an actual environment with the real experimental wheel-legged robot. The performance of the proposed system was evaluated through these experiments. Different types of rough terrains with obstacles were built for the experiments. A procedure of this series of experiments is shown as follows:

i. The agent learns in the training environment.

ii. The effectiveness of acquired actions is verified in the training environment.

iii. The effectiveness of acquired actions leaned in the training environment is evaluated in the test environments. Two test environments are prepared, where the same obstacles as the training environment are placed in different patterns.

In the experiments, the action value function was set to all 0 before starting to learn in the training environment. We conducted the experiments as the episodic events, which is useful to evaluate the performances properly. Fig. 7 shows the learning flow with the episode. One episode starts when the robot begins actions on the "START" line. Then, the episode
ends as the robot reaches on the "GOAL" line. As the way to evaluate the effectiveness of acquired actions, the number of actions per episode was compared before and after learning. Fig. 8 shows the way to count the number of actions in detail. In the “before learning” phase, an action is uniform randomly chosen from all the possible actions in Table 1. In the “after learning” phase, actions are taken by the policy acquired through learning.

Fig. 7 Learning flow with episode

Fig.8 Flow to measure the number of actions

4.2 Learning in the training environment

Fig. 9 shows a rough terrain constructed for the training environment. In this environment, some rectangular parallelepiped objects were fixed on the floor as obstacles. Fig. 10 shows the size of the objects, which is hard to get over by leg movements only. Also, the objects fixed on the floor are heavy enough not to move even when these are pushed by the robot. Fig. 11 shows the situation of the learning process. The agent learned 30 episodes in this environment. As a result, the agent acquired the policy shown in Fig. 12.
4.3 Performance evaluation in the training environment

To compare the number of actions per episode, the robot performed 50 episodes by using “before learning” and “after learning” agents. Fig. 13 shows the comparison of the number of actions per episode between each agent in the training environment. As shown in this graph, the frequency that the robot completes an episode with fewer actions has significantly increased after learning. Also, the independent t-test about the average of the number of actions per episode before learning ($M = 8.00, SD = 2.74$) and after learning ($M = 4.44, SD = 1.15$) is performed to validate the result. As a result, the t-test shows that this decrease of the average is statistically significant ($t(98) = -8.47, p < 0.01$).

4.4 Performance evaluation in the test environments

To verify the performance of adaptation by the actions acquired in the training environment for generic rough terrains, we performed the experiments in different rough terrains. Two test environments were prepared as shown in Fig. 14 (test environment I) and Fig. 15 (test environment II). These environments were built by using the same objects but in different patterns as in the training environment. To compare the number of actions per episode, the robot performed 50 episodes in each test environment by using learned and unlearned agents.

Fig. 13 The number of actions per episode in the training environment

Fig. 14 Test environment I

Fig. 15 Test environment II

Fig. 16 shows the comparison of the number of actions per episode between each agent in the test environment I. As shown in this graph, the frequency of the learned robot to complete the episode with fewer actions has increased like the verification in the training environment. To verify the results, the independent t-test about the difference of the average number of actions per episode between unlearned agent ($M = 9.92, SD = 3.93$) and learned agent ($M = 5.32, SD = 1.35$) was performed. The t-test results show that the decrease of the average is statistically significant ($t(98) = -7.83, p < 0.01$).
I, pp.
ded
al
re
i
ns to
i
r
s
ion, the agent selects the combination
of either or both side leg increase.

c
1
escape the situation
of experiments.

actual
4
statistically significant \( t = 9.38 \) is performed to verify the result.

F
The

The

\( \text{Fig. 16} \) The number of actions per episode in the test environment I

Fig. 17 shows the comparison of the number of actions per episode between each agent in the test environment II. As shown in this graph, the frequency of the learned robot to complete the episode with fewer actions has increased like the validation in the training environment and the test environment I. The independent \( t \)-test about the difference of the average number of actions per episode between unlearned agent \( M = 9.38, SD = 4.65 \) and learned agent \( M = 5.02, SD = 1.36 \) is performed to verify the result. As a result, the \( t \)-test shows that this decrease of the average is statistically significant \( t (98) = -6.36, p < 0.01 \).

The proposed method is expected to be useful in various environments other than the rough terrains such as on sand, snow or icy surfaces. We plan to conduct the experiments in these different environments. Also, the transition of electrical current passing through wheel driving motors may not be able to detect external force properly in some environments. In such cases, a direct force measuring sensor such as a strain gauge attached on a leg should be better. Currently, we are reconsidering the configuration of our wheel-legged robot including this force measuring method.

5. CONCLUSION

5.1 Summary of this paper
In this research, we proposed the system for a wheel-legged robot to escape from the situation where the robot cannot move by itself. The main idea of the proposed system is the usage of the transition of leg torques changed by external force. The result of the series of experiments showed the effectiveness of the proposed system in the actual environment. The results also showed that the proposed system is effective as an online learning method.

5.2 Future work
The proposed method is expected to be useful in various environments other than the rough terrains such as on sand, snow or icy surfaces. We plan to conduct the experiments in these different environments. Also, the transition of electrical current passing through wheel driving motors may not be able to detect external force properly in some environments. In such cases, a direct force measuring sensor such as a strain gauge attached on a leg should be better. Currently, we are reconsidering the configuration of our wheel-legged robot including this force measuring method.

REFERENCES


4.5 Discussion
The effectiveness of the proposed method in the actual rough terrain has been shown through the series of experiments. The agent has acquired the actions to escape the situation when the robot has got stuck. Fig. 12 shows that the agent selects actions of the combinations of leg and wheel actions when the torques of either or both side leg increase. For instance, when the left and right leg states are (5, 2) (i.e., the left leg torque increases), the agent selects the combinational action of crawl gait I and wheel movement. Also, when these states are (1, 5) (i.e., the right leg torque increases), the agent selects the combinational action of crawl gait II and wheel movement. In addition, when these states are (3, 3) (i.e., the both leg torques increase), the agent selects the combinational action of tripod gait and wheel movement. As a result of these actions acquired in the training environment, it has been shown that the ability of adaptation for generic rough terrains has also been acquired.

The small number of episodes needed for the agent to learn sufficiently is also notable. In these experiments, the agent learned 30 episodes in the training environment, and exhibits high performance on test environments. It is considered to be small enough to learn in real time with online learning method. Therefore, the result of the experiments has shown that the proposed method is effective for online learning method in actual environments.


