Conquest Oriented Robot Knowing Its Own Availability

Sho Yamauchi
Future University Hakodate, 116-2, Kameda Nakanocho,
Hakodate-shi, Hokkaido, 041-0803, Japan,

Keiji Suzuki
Future University Hakodate, 116-2, Kameda Nakanocho,
Hakodate-shi, Hokkaido, 041-0803, Japan,
E-mail: yamauchi@fun.ac.jp, kjsuzuki@fun.ac.jp
www.fun.ac.jp

Abstract

Robot is designed for achieving specific task. However, robot is able to do much more things than it is expected by its own hardware. In this paper, concept of conquest oriented robot is proposed to know its availability systematically and autonomously. Algorithm of conquest oriented robot and its experimental result in simulation field for the first step is shown in this paper.

Keywords: autonomous, conquest, robot, availability

1. Introduction

Robot is designed for achieving specific task. Cleaner robot is designed to collect dusts in room and industrial robot is designed to assemble some kind of merchandise in factory. These tasks are designed by human and the only thing robots do is complete the tasks. Previously, robot is just a tool to achieve some tasks that are hard for human or tool for automation. However, robot can do much more things by their body even though the thing is not expected to do by human. For example, Cleaner robot with two wheels can push something as well as cleaning. This means that the cleaner robot might bring something heavy and might communicate with human by pushing their hands, for example. Robot has lots of possibility by their given body and this possibility is not limited to the purpose expected by human. From these points of view, concept of conquest oriented robot is proposed in this paper. Base algorithm to find what they can do by its body is proposed for the first step and experimental result using the algorithm is shown.

2. Concept of conquest oriented robot

The purpose of conquest oriented robot is finding, increasing and maintaining available things of the robot.
Robot has its own available event in its existing environment (Fig. 1). In other words, conquest oriented robot tries to increase event that robot can execute with high probability, increase the probability and maintain its events and probability.

Comparison between previous robot design and conquest oriented robot is shown in Fig. 2. Robot is designed to reach only one event, it means "task", previously. However, if the task is changed, robot must be re-

Published by Atlantis Press

designed to achieve new task. On the other hand, conquest oriented robot researches its own availability previously and it can change its task easily if the new task is within its availability.

What is an advantage of conquest oriented robot? If robot knows its own availability, robot is able to do maximum work even though it does not know the task preliminarily. If the task is unavailable, robot is able to know it is unavailable by its given hardware. In this case, human should prepare another robot that has suitable hardware.

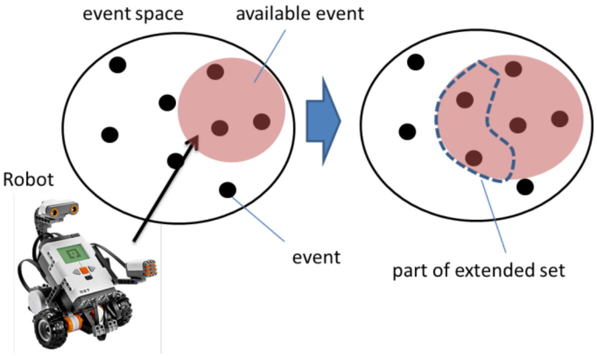

Fig. 1. Conquest oriented robot concept. Event space means a set of all the events that might happen in current environment. Conquest oriented robot tries to extend a set of event that is able to happen at any time.

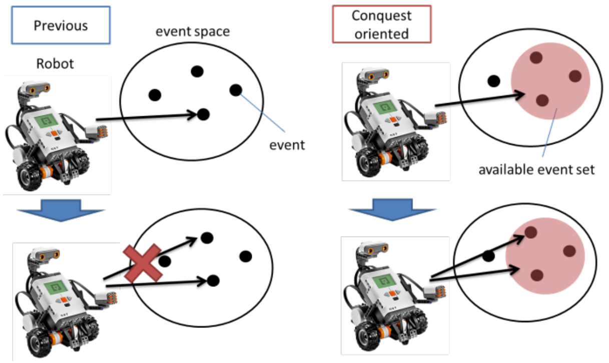

Fig. 2. Previous robot design and conquest oriented design. In previous robot design, the specific event is focused and robot tries to achieve the specific task. However, conquest oriented robot tries to find “what it can do” and maintain those things.

3. Action Fragment and Variable Action Generation

As a unit of robot’s available event, action fragment is introduced. Action fragment is a part of time-series data of robot’s action as shown in Fig. 3. Action fragment **F** is defined as

.

$$\mathbf{F} = [\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_n] \qquad (1)$$

$$\vec{x}_i = \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \\ \vdots \\ x_i^{(m)} \end{bmatrix} = \begin{bmatrix} a_i^{(1)} \\ a_i^{(2)} \\ \vdots \\ a_i^{(n_a)} \\ b_i^{(1)} \\ b_i^{(2)} \\ \vdots \\ b_i^{(n_b)} \end{bmatrix} \qquad (2)$$

where $a_i^{(k)}$, $b_i^{(k)}$are k-th actuator output and sensor input at time i respectively .

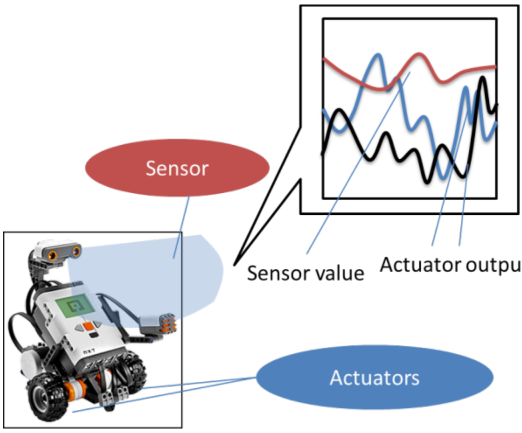

Fig. 3. Action fragment of conquest oriented robot. Action fragment is made of time series data of sensors and actuators.

Use Fourier series expansion to generate various types of time-series action for robot. Equation is as follows.

$$f(\omega t) = \frac{a_0}{2} + \sum_{n=1}^{n_{\max}} (a_n \cos n\omega t + b_n \sin n\omega t) \quad (3)$$

If coefficients a_n, b_n are changed periodically, various types of wave would be generated and this wave is used as actuator output of robot for generating various actions. To find new action fragment, robot has to detect whether the action fragment is different from all other experienced action fragments. It means that criterion to measure the difference between two action fragments is needed. For this purpose, criterion **G** is introduced as follows.

$$G(\mathbf{F}_A,\mathbf{F}_B)=\frac{AMSS(\mathbf{F}_A,\mathbf{F}_B)}{s_{\max}} \tag{4}$$

$$s_{\max}=AMSS(\mathbf{F}_{long},\mathbf{F}_{long}) \tag{5}$$

$$\mathbf{F}_{long}=\underset{\mathbf{F}\in\{\mathbf{F}_A,\mathbf{F}_B\}}{\arg\max}\{len(\mathbf{F})\} \tag{6}$$

where $\boldsymbol{F_A}, \boldsymbol{F_B}$ are compared action fragments. Function len() returns length of action fragment's time series data. Function AMSS() is also returns the value of AMSS between two action fragments. AMSS (Angular Metrics for Shape Similarity) [Ref. 1] is method to compare two time series data and calculate similarity. AMSS assume time series data as vector and calculate similarity by cos similarity and dynamic programing. It can be applied to time series data that are different length. In criterion **G**, similarity is normalized and all the results can be compared with each other. The value of criterion **G** is called similarity rate in this paper. Robot stores new action fragment if new action fragment is different enough comparing with all the action fragments stored already as follows.

For all *i*, if similarity rate r_i satisfies condition

$$r_i=G(\mathbf{F},\mathbf{F}_i)<c_{acc\_l} \tag{7}$$

then store action fragment **F**
where $c_{acc\_l}(0<c_{acc\_l}\leq 1)$.

The process to store new action fragment is called "accumulate" in this paper. "Accumulate" is considered as the first step of conquest oriented robot.

4. Experiment

Confirm the process of "accumulate" in simple experiment. Consider a robot in 2D physics simulation field shown in Fig. 4. In this simulation field, a differential wheeled robot and two objects are placed. The robot can detect object in front of it and obtain the distance from detected object. It means that this robot has 1 sensor and 2 actuators.

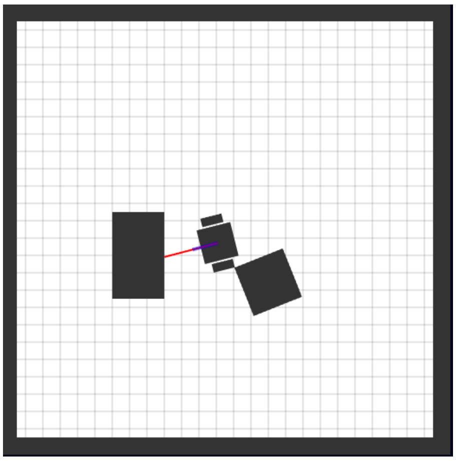

Fig. 4. 2D physics experimental field. Red line indicates the sensor of differential wheeled robot. Two objects are also placed.

Friction between wheels of robot and floor, objects and floor exists. Robot starts "accumulation" in this field and store action fragments. All the fragments have the same fixed length in this time. In this experiment, two types of action generating methods are applied. One changes coefficients in equation (3) periodically. Another does not change coefficients. If coefficients are changed periodically, various types of action of robot would be generated and number of action fragments is supposed to increase. However, action of robot is limited, so the number of action fragment is expected to be saturated eventually. On the other hand, if coefficients in equation (3) are fixed, generated action of robot is far more limited and number of stored action fragment is expected to be smaller. Experimental result is shown in Fig. 5. Horizontal axis denote time step and vertical axis means number of action fragments. "Update" is a case that coefficients in equation (3) are periodically updated. "Fixed" means coefficients in equation (3) are fixed.

As expected, number of action fragment saturated in both case. Also, as we expected, the number of action fragment in "Fixed" case is smaller than "Update". This result means that criterion **G** for similarity rate works properly and "accumulation" phase algorithm works properly.

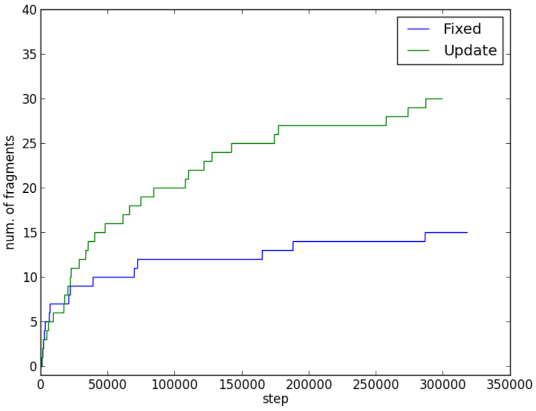

Fig. 5. Experimental result. Horizontal axis denote time steps and Vertical axis denote number of action fragments..

5. Conclusion

The concept of conquest oriented robot that tries to know its own availability is proposed in this paper. For the first step, algorithm of "accumulate", the first phase of conquest oriented robot algorithm, is proposed. AMSS based similarity criterion is also introduced. "Accumulate" phase process is confirmed by experiment using differential wheeled robot in 2D physics simulation experiment field. We develop second and third phase of conquest oriented robot algorithm for the future work.

References

1. Tetsuya NAKAMURA, Keishi TAKI, Hiroki NOMIYA and Kuniaki UEHARA, *AMSS: A Similarity Measure for time Series Data, IEICE TRANSACTIONS on Information and Systems, Vol.J91-D, No.11, pp.2579-2588, 2008.*