

Paper:

Proposal of Method “Motion Space” to Express Movement of Robot

Kentarou Kurashige*, Naoki Kitayama*, and Masafumi Kiyohashi**

*Muroran Institute of Technology

27-1 Mizumoto-cho, Muroran city, Hokkaido 050-8585, Japan

E-mail: kentarou@csse.muroran-it.ac.jp, kit.hetare@gmail.com

**Sun Information & Service, Co, Ltd.

3-15-9 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

E-mail: m.kiyohashi@gmail.com

[Received February 19, 2012; accepted June 20, 2012]

In recent years, the use of robots has been spreading to various fields. Further, requirements for the use of robots are increasing. A method is therefore necessary for allowing persons who are not experts in using robots to actually operate a robot. We intend to develop a method for robot operation whereby a user need not have technical knowledge. In this paper, we focus on methods whereby a user of a robot assigns movement to the robot and the robot reproduces movement. One of the most widely used techniques that is used nowadays involves teaching playback. Teaching playback is a method in which a teacher moves a robot using controllers and lets the robot record movement and then play it back. Robots functioning via teaching playback cannot however adapt to a changing environment. The environment in which human beings live generally change. Teaching playback is therefore not usable in variable environments. Methods for generating movement robustly in environments have been studied. Designing the movement of a robot by using these methods cannot be done, however without understanding complicated formulas. Only movement designers having technical knowledge can use these techniques. We propose new knowledge of movement to solve the problems present in these methods. Knowledge of movement is information involving the generation of movement in a robot. In conventional methods, knowledge of movement was a complicated formula. By using our method, a robot incorporates knowledge based on information obtained by moving a robot just like in the teaching playback method. We expect that by using our methods a user can move a robot in the desired manner.

Keywords: motion space, knowledge of movement, method to express movement of robot, teaching playback

1. Introduction

In recent years, the development of robots that work in an environment where human beings live, such as a home or offices has increased. These environments include situations that are difficult to predict beforehand. It is thought that a person using a robot at home may not have technical knowledge. We therefore intend to develop methods for operating robots, wherein no expertise or a particular skill set is required. In this paper, we focus on methods that assign movement to a robot and allow the robot to perform that movement.

One of the most widely used techniques currently being used is teaching playback [1]. Teaching playback is a method in which the movement of an actuator is recorded when a robot moves and this movement is played back repeatedly later. A control panel called a teach pendant is generally used for recording movement. We make a robot repeat simple movements using this method. Nowadays, teaching playback is the method that is used in most factories. A robot functioning using teaching playback, however can perform only movement taught by a user. Hence, a robot functioning using teaching playback has difficulty in adapting to environmental change. A person teaching movement to a robot therefore has to anticipate every change in the environment, so using teaching playback in a changing environment is difficult.

We intend to develop a method for operating a robot robustly in a changing environment. There are some studies on the intelligence of robots regarding this. In the field of cognitive robotics [2, 3], various studies are conducted on the intelligence of robots. Here, some methods have been considered for designing knowledge to bring about movement in a robot. There is a method, for example, that uses a neural oscillator [4, 5]. Using this method, a robot is synchronized with input for a neural oscillator. The neural oscillator provides movement to the robot and is not stopped by agitation. There is also a method that uses attraction caused by an attractor [6, 7] for expressing the movement of a robot. Through these methods, a robot is able to adapt to environmental change and is able to execute a task flexibly. It is necessary, however, for a



designer to design knowledge of movement before using these methods. In these methods, a designer of movement designs knowledge of movement using dynamic formulas, so only designers with technical knowledge can use these methods. It is difficult for users who do not have technical knowledge to design knowledge of movement using these methods. There is, however, a method called the Active Learning Method (ALM) for learning objective movement using trial and error, such as in reinforcement learning [8]. Using this method, the controller of the robot obtains the most suitable movement using past information on movement that the controller acquired for a database by using trial and error. As a result, the robot obtains movement for achieving a certain purpose automatically. When users operate robots by using the ALM, however they must prepare an evaluation function. An evaluation function is used for choosing movements that are obtained by trial and error. Technical knowledge is necessary, however, to design the evaluation function. It is difficult for users who do not have technical knowledge to do so. We thus consider a method that can generate knowledge of movement by using a method such as teaching playback. If such a method exists, a user without any technical knowledge will be able to operate a robot with an intelligent system. We therefore consider a method in which a user can teach movement immediately like in teaching playback. Further, this method should allow robot to combine various information on movement into one database on knowledge of movement.

The purpose of this study is to develop a new method for expressing the movement of a robot. In this study, we consider a method for expressing knowledge of movement so that we can use a method such as teaching playback. Flexible movement can be achieved from this knowledge of movement. In addition, we presume that a user can handle knowledge of movement in a form in which correction is easy.

We consider a method for acquiring knowledge through teaching. We focus on the frequency of movement for acquiring knowledge. When a user instructs a robot several times, a slight difference appears in each movement. In the case of a user who is not an expert in operating robots, the difference is considered to be greater. During multiple instructions, however, it is considered that the frequency at which a robot assumes a state necessary for movement becomes higher if the user targets one movement. If a robot makes movement to take a frequent state using these plural movements, a robot can generate a movement required by the user. It is considered that even a person who is not very good at instructing can teach movement through repeated instruction. We propose knowledge of movement of the robot Motion Space.

In Section 2, we explain knowledge of movement and propose Motion Space as a method descriptive of knowledge of movement. In Section 3, we explain the method for moving a robot using the knowledge of movement explained in Section 2. We propose a system that generates knowledge on the movement of a robot using Motion Space and also generates robot movement. In Section 4,

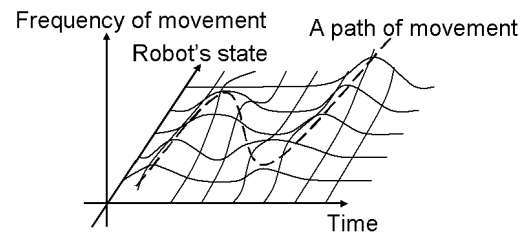


Fig. 1. Frequency of movement. Movement has a path of movement on the space of time and robot's state. Frequency of movement appears around this path in space. If one point in space is near the path, the point has high frequency. High frequency implies a high probability of movement.

we describe an experiment performed to determine the effectiveness of the system suggested in Section 3. We use an actual robot and perform an experiment to teach and reproduce movement. In Section 5, we summarize this paper and discuss future work.

2. Motion Space

2.1. Knowledge of Movement

We focus on a method for expressing the movement of a robot. We treat information on expressing how the robot moves actuators to perform a movement as knowledge of movement in a robot. If the environment does not change, we can operate a robot by deciding the actuator's output beforehand. In an environment where human beings live, however, changes inevitably occur. After an environment has changed, the robot may not reproduce movement as before. This is because the current environment differs from the environment that the movement of the robot was initially based on. Human beings can, however, perform the same movement in a changing environment, so we thought about the process involved in human beings for generating movement. Human beings have information on movement into their brain, and it is thought that they generate movement using this information when they perform movement. We call this information knowledge of movement. We suggest Motion Space as knowledge of movement for a robot.

2.2. Concept of Motion Space

In this section, we consider a method for expressing knowledge of movement. We describe the movement of a robot using the probability of the state of the robot. We presume that a user teaches the same movement to a robot repeatedly. Because movement is taught manually, input movement differs slightly each time. When the user collects information on many movements performed by the robot, a difference in frequency in movement is observed. **Fig. 1** shows frequency of movement. We consider frequency to be related to the probability of the movement of the robot. If the robot traces points of high frequency,

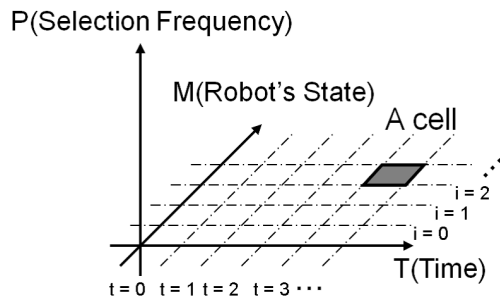


Fig. 2. An example of Motion Space when the robot state is at the first origin. The space is constructed using time and the robot's state as parameters. The robot's state is the value of the sensor or actuator. Motion Space is divided by time and the robot's state. Each part of divided Motion Space is called a cell. A cell has the value of the selection frequency.

it generates the movement desired by the user. Load is exerted on the robot when it changes its own state, however, so the robot may turn over. It is therefore necessary to be able to gradually update the state of the robot. The robot can continue movement even if its posture is changed by agitation.

2.3. Definition of Motion Space

In this section, we explain how knowledge of movement is realized. We prepare a space comprising axis M (M_1, M_2, \dots, M_l) expressing the state of the robot, with axis T expressing time and axis P expressing the value of the selection frequency. The state of the robot is determined by the values of the sensors or actuators. M is a matrix expressing the state of the robot. When the number of sensors and actuators is l , M comprises M_1, M_2, \dots, M_l .

The selection frequency is the frequency at which movement input into a robot passes in a space. The size of Motion Space is determined by the time taken to teach a movement, the number of actuators in the robot, and the number of states of actuators. When there is one actuator in a robot, for example, Motion Space consists of two dimensions. One dimension is for the actuator and the other dimension is for time. We show an example of Motion Space when the robot has one actuator in **Fig. 2**. When there are two actuators in a robot, Motion Space consists of three dimensions. Two dimensions are for actuators, and one dimension is for time. Regardless of the number of dimensions of Motion Space, we can express only one movement as one orbit in Motion Space.

We divided Motion Space into parts and assigned selection frequency to each part. We call each of these parts a cell. We express the cell address of the M_l axis in i_l and the T axis in t , so the selection frequency is represented by $p(t, i_1, i_2, \dots, i_l)$. Each cell has one selection frequency.

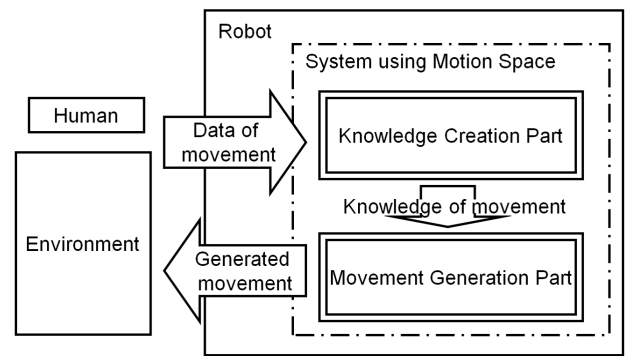


Fig. 3. Outline of the system using Motion Space.

3. Method for Operating Robots Using Motion Space

3.1. Outline of a System Using Motion Space

In this section, we propose a system for moving a robot using Motion Space. First, we explain the cycle of operation using our system. A user teaches movement in our system through a robot. The system collects information on input knowledge on movement and records it. Next, the user sends a command for movement generation to the system. The system generates robust movement using recorded knowledge.

We proposed Motion Space as knowledge of movement in the previous section. We divide our system into two parts. One part involves creating knowledge from input. We call this part knowledge creation. The other part involves generating movement from the knowledge created. We call this part movement generation. **Fig. 3** shows an outline of our system. The knowledge creation part creates knowledge of movement using input from a user. Created knowledge is used for movement generation. The movement generation part helps operate the robot through knowledge of movement. Knowledge of movement is transferred from the knowledge creation part to the movement generation part. This knowledge of movement is Motion Space.

3.2. Knowledge Creation Part

The knowledge creation part converts information on movement to knowledge of movement. A change in actuator level indicates information on movement. A user moves the robot and instructs a movement. The system then converts the movement into a form of Motion Space. The system adds the knowledge of movement to Motion Space where conventional knowledge of movements was recorded. By repeating this process, knowledge of movement tends toward movement desired by a user. We treat changes in the actuator level of a robot in a period of time as one movement of a robot.

Next, we explain the process for the knowledge creation part. **Fig. 4** shows the structure of the knowledge creation part. When the system obtains data on the robot's movement, the system obtains the robot's state at

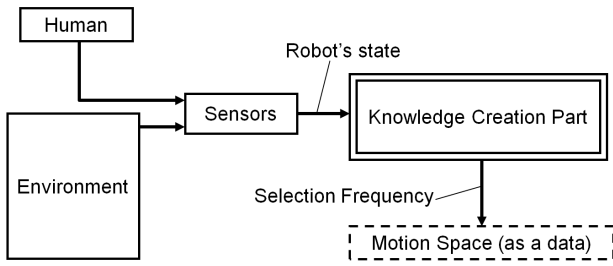


Fig. 4. Details of the system around the knowledge creation part.

all times. The system obtains the robot's state from the robot's sensors. At this time, the system converts the robot's state to selection frequency. The system calculates the values of selection frequency for all cells and these selection frequencies are placed in temporary Motion Space. A cell nearer to input movement is assigned a higher selection frequency. The selection frequency in temporary Motion Space takes the shape of a mountain with its center as input movement. The selection frequency of the movement is calculated using Eq. (1).

$$u_{t,i} = \frac{C}{r+1} \quad \dots \quad (1)$$

$u_{t,i}$ is the selection frequency of a cell with step t and robot state i in temporary Motion Space. C is the maximum value of the selection frequency in each cell, which is obtained by creating knowledge once. r is the number of cells from the center cell nearest to input movement. r is an integer from 0 to $\lfloor q/2 \rfloor$. q is the number of cells that add selection frequency once, and q must be an odd number. The calculated selection frequency in temporary Motion Space is integrated with master Motion Space. Eq. (2) is the selection frequency in temporary Motion Space integrated with cells of master Motion Space.

$$p_{t,i} \leftarrow p_{t,i} + u_{t,i} \quad \dots \quad (2)$$

$p_{t,i}$ is the selection frequency of a cell with step t and robot state i in master Motion Space. When the system starts, the selection frequency in all cells $p_{t,i}$ is 0. The user adds the knowledge of movement to this Motion Space repeatedly. Selection frequencies are piled up in sequence in Motion Space. This Motion Space is used in the movement generation part.

3.3. Movement Generation Part

The movement generation part operates a robot using knowledge created by the knowledge creation part. The system generates movement depending on the state of the robot. While the system generates movement, it simultaneously checks the state of the robot. In this study, the system monitors the state of each actuator of the robot, which is regarded as the state of the robot. The system compares actual movement with ideal movement of the robot and generates the next movement to obtain ideal movement. Actual movement is movement that the sys-

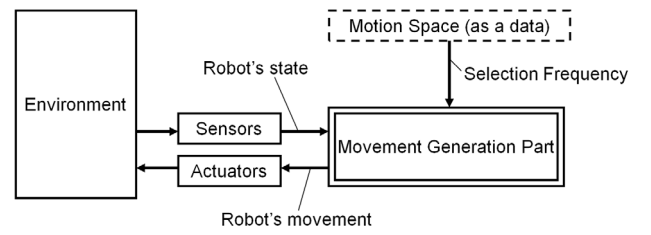


Fig. 5. Details of the system around the movement generation part.

tem obtains from the sensors of an actual robot. Ideal movement is movement that is desired by the user of the robot. When the system generates movement, the system gradually brings the state of the robot close to an ideal state. A user decides the degree of approach so that load does not affect the robot too much. **Fig. 5** shows the structure of the movement generation part.

Here, we explain the process of how the system generates the robot's movement. Before the user of a robot begins operation, parameters that influence a robot are determined. To gradually change a robot's state, virtual acceleration movement in Motion Space is used. In this part, the State Point has a weight m and every cell has an attraction. The State Point moves because of acceleration. The user of the robot decides the range of candidate cells r depending on the robot. This is decided so that minimal load is exerted on the robot. When the robot moves, the system checks the present state of the robot and calculates the next state every ΔT seconds.

We explain the calculation cycle for every step. First, the system calculates the attraction power for the State Point. The State Point shows the current state of the robot. The system checks the selection frequency of cells within the range of r in the row of the next step and decides the direction in which power is added next. The center of range r is a position of the State Point in the row of the next step of calculation. The system chooses a target cell having the highest selection frequency from this range r . **Fig. 6** shows the process involved in choosing a target cell. The system calculates attraction power a_t based on the chosen cell. We consider attraction power to be higher when the distance between the State Point and target cell is longer and the selection frequency of the target cell is higher than those of surrounding cells. Eq. (3) calculates the power of the attractor.

$$F_t = k_t \times (N_{t+1} - M_t) \quad \dots \quad (3)$$

F_t is the power of the attractor in step t . k_t is a comparison of the values of the selection frequency target cell and surrounding cells. This comparison is achieved by using Eq. (4). N_{t+1} is the robot's state at step $t+1$. M_t is the current state of the robot, which is obtained from sensors.

$$k_t = \frac{p_{t,max}}{\sum_{i=1}^n p_{t,i}} \times \alpha \quad \dots \quad (4)$$

$p_{t,i}$ is the selection frequency of cell time t and robot state

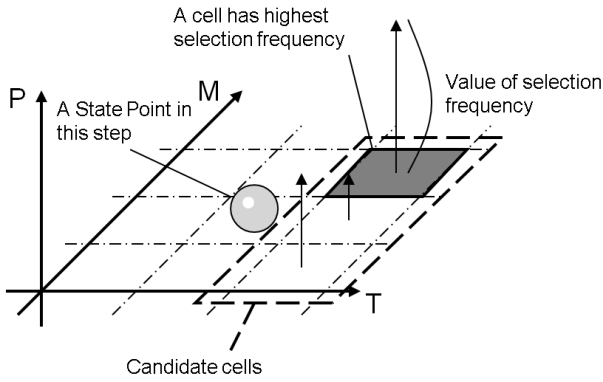


Fig. 6. Process for choosing a target cell. A target cell is chosen from the range of candidacy of target cell r . A target cell has the highest selection frequency in range r .

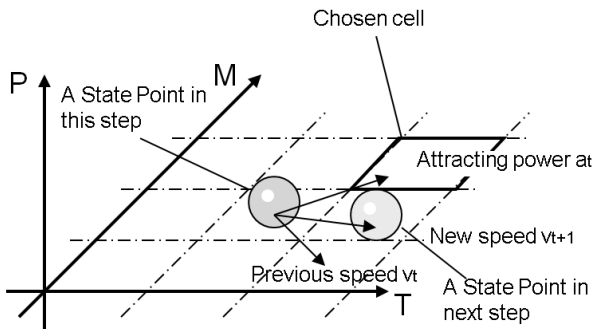


Fig. 7. Method for calculating virtual speed of the State Point. The State Point is attracted by the cell that has the highest selection frequency. The State Point is accelerated and attains a new speed.

i . $p_{t,max}$ is the selection frequency of the target cell. α is the value that decides the balance between the power of distance and the power of difference in height.

Next, the system calculates acceleration for the State Point based on a calculated attractor. The system accelerates the State Point and changes the robot's state for a moving robot. The State Point has a virtual speed and moves in Motion Space. The system changes this virtual movement by providing acceleration to the State Point. **Fig. 7** shows the process involved in changing virtual movement by adding acceleration. Eq. (5) calculates virtual acceleration.

$$\mathbf{a}_t = \frac{k_t \times (\mathbf{N}_{t+1} - \mathbf{M}_t)}{m} \dots \dots \dots (5)$$

\mathbf{a}_t is acceleration for the State Point in step t . m is the virtual weight of the State Point.

Finally, the system calculates the state the robot aims at based on the accelerator. The system calculates speed \mathbf{v}_{t+1} of the ball at time $t+1$ using Eq. (6).

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta T \mathbf{a}_t \dots \dots \dots (6)$$

\mathbf{v}_t is the virtual speed of the State Point in step t . ΔT is the time taken to move the robot from the previous step to the current step. The system calculates the targeted state

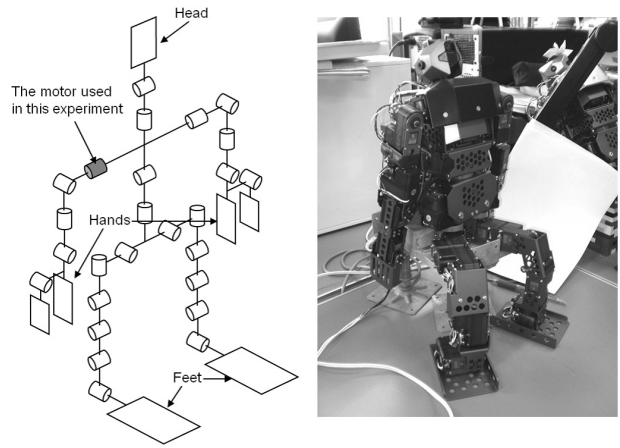


Fig. 8. Structure of the robot in this experiment (left) and appearance of this robot (right). The gray motor on the left in this figure is the motor used in this experiment.

of the robot at step $t+1$ using Eq. (7).

$$\mathbf{M}_{t+1} = \mathbf{M}_t + \Delta T \mathbf{v}_{t+1} \dots \dots \dots (7)$$

\mathbf{M}_{t+1} is the matrix that expresses the desired state of the robot. The system repeats calculation at each step.

4. Experiment for Teaching Movement to Actual Robots

4.1. Summary of this Experiment

In this section, we summarize an experiment to move a robot using the system proposed in this paper. Through this experiment, we show that a system using Motion Space is suitable for generating robot movement. We discuss two experiments performed. First, we validate whether the robot can form Motion Space using the proposed system. Next, we check whether the robot can generate movement robustly using Motion Space formed in the previous step. In this experiment, we use an actual robot. The actual environment is suitable for confirming the usefulness of the proposed system. We describe the robot used in this experiment in Section 4.2 and describe the method for investigating robustness in Section 4.3.

4.2. Experiment Settings

First, we describe the robot used in this experiment. We use a humanoid robot called Specys. The physical structure and appearance of Specys is shown in **Fig. 8**. In this experiment, we use the robot on a stand. We use only one servomotor as the robot state. Expansion to a multi-input and multi-output system is easy to do for this system. If we can apply the system to a robot having a single input and single output, we can apply this system to a robot having multi-input and multi-output. We therefore experiment on a robot having single input and single output. We place the servomotor on the right shoulder of the robot. The placement of the servomotor is shown in

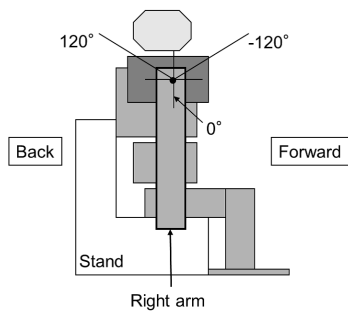


Fig. 9. Movable range of the servomotor used in this experiment. This graph is a summary of the sagittal plane of the robot in Fig. 8. The servomotor on the right shoulder moves between -120° and 120° . The angle decreases when the robot moves the arm forward, and vice versa.

Table 1. Parameters used in this experiment. All symbols described correspond to parameters.

Parameter	Symbol	Value
Moving time [s]	–	10.0
Step size of time [s]	ΔT	0.5
Step size of angle [deg]	ΔM	12.0
Number of cells that add selection frequency	q	11
Number of candidate cells	r	9
Virtual weight of State Point	m	0.07
Inclination factor	α	1.0
Maximum value of selection frequency	C	1.0

Fig. 8. In this experiment, we treat the servomotor angle as the robot state. The servomotor moves between -120° and 120° (Fig. 9). The smallest unit of angle is 0.1° . We adjust the robot so that only the right arm moves.

Next, we explain parameters of Motion Space and the system used in this experiment. Table 1 lists parameters. Moving time is the time between the start and end of movement. Step size of time is a time interval divided into cells. Step size of angle is an angle interval divided into cells. Number of cells that add selection frequency is the number of cells that the system adds selection frequency to. Number of candidate cells is the number of cells in the target cell range. Virtual weight of the State Point is a virtual value that decides the ease of State Point movement. Inclination factor is a value that decides the balance between power of distance and power of difference in height. Maximum value of selection frequency is the maximum selection frequency given to a cell once. Table 1 also lists symbols corresponding to parameters. Values in this table were used in this experiment. Using this setting, time is divided by 21 and the motor angle is divided by 20. Motion Space is composed of 420 cells.

4.3. Experiment Procedure

In this paper, we perform two experiments. First, we perform an experiment to generate Motion Space. Using the human hand, we input three movements to the pro-

posed system through a robot. Using the human hand, information is provided on three different movements. Information is also provided on each movement in the system. By comparison with input data, we confirm that Motion Space is formed by the proposed system.

Next, we perform an experiment to generate movement using Motion Space. We investigate two points here. One is whether the proposed system generates movement robustly. The other is whether the proposed system generates the movement desired by the user of the robot. The robot is considered to return to the movement desired by the user when the state of the robot is changed by agitation if the system can generate movement firmly. In this experiment, we therefore decide that the state of the robot changes at 0 s. We operate the robot five times and change the initial state each time. Five initial angles are used for the servomotor of the robot and the type of movement generated by the proposed system from those states is checked; 0° is the original state and -120° , -60° , 60° , and 120° are changed states. The first state is assumed to be the original movement and the other four states are assumed to be changed by agitation. In addition, we compare output to the average of input and check whether the movement desired by the user is achieved.

4.4. Experiment for Making Motion Space

In this section, we perform an experiment to generate Motion Space. First, we obtain input data from the actuator of the robot. Information is provided on three different movements using input through the human hand. Next, we make an index of the robot movement desired by the user using these inputs. Finally, we validate the Motion Space generated by the proposed system.

Figures 10, 11, and 12 show the results of this experiment. Input data provided through the robot are shown in Fig. 10. The horizontal axis represents time and the vertical axis represents the motor angle of the robot. Three input data points have a region where they are near and far from each other. The region where each movement value is near to each other is movement that is strongly required by the user. These inputs for the proposed system are shown in Fig. 10.

Figure 11 shows the average and standard deviation of input. The region where standard deviation is small is where there are few differences in input. Such a trend appears near 0 s, which is the beginning of movement, from 3 s to 5 s, and near 10 s. We use the average and standard deviation shown in Fig. 11 as an indicator of the movement desired by the user.

Figure 12 shows Motion Space generated by using information on the three movements shown in Fig. 10. The value of selection frequency is expressed by the density of the color. The lighter cell color, the higher the selection frequency. Fig. 12, shows that cells having a high selection frequency appear in the region of 0 s, which is the beginning of movement, the region from 3 s to 5 s, and in the region of 10 s. This trend is in accordance with the trend in the amount of standard deviation of input. Motion Space was therefore formed as expected.

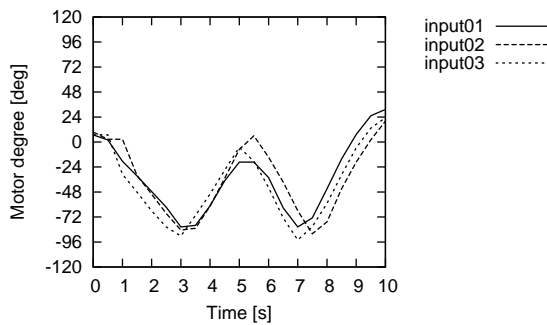


Fig. 10. Input angles for the robot, i.e., Input01, input02, and input03. Input includes some unevenness because these are input manually.

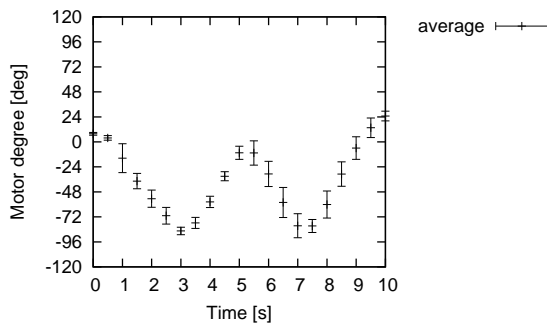


Fig. 11. Average and standard deviation of input in Fig. 10.

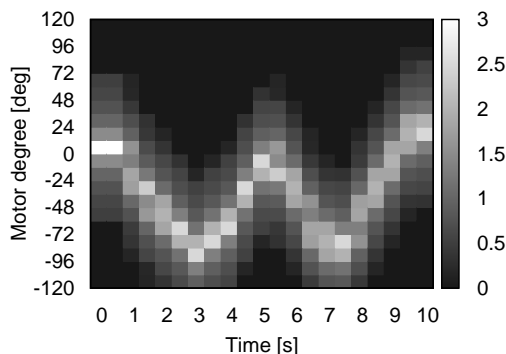


Fig. 12. Map of selection frequency in Motion Space in this experiment. This is a map that expresses the height of the selection frequency of all cells in Motion Space after the input shown in Fig. 10. The height of the selection frequency is expressed by the density of color. The darker the color, the lower the selection frequency.

4.5. Experiment for Generating Movement Using Motion Space

In this section, we perform an experiment for generating movement using Motion Space as described in Section 4.4. We investigate whether the proposed system generates movement robustly and whether it is movement desired by the user of the robot. First, we operate the robot using Motion Space. Its movement starts using five initial angles: -120° , -60° , 0° , 60° , and 120° . We compare these outputs and the average and standard deviation of the input in Fig. 10.

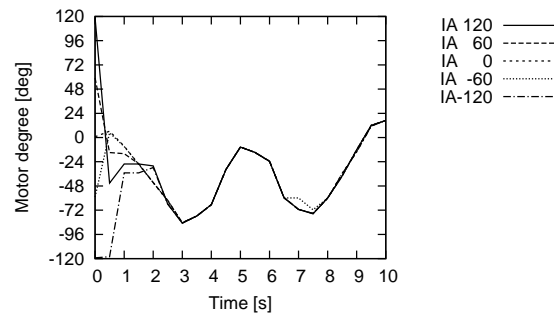


Fig. 13. Output angles of the robot. The IA (initial angle) in captions are initial angles of the robot that generate movement from Motion Space. IA -120° , for example, is a trace of the change of angles of the motor when the motor's initial angle is -120° .

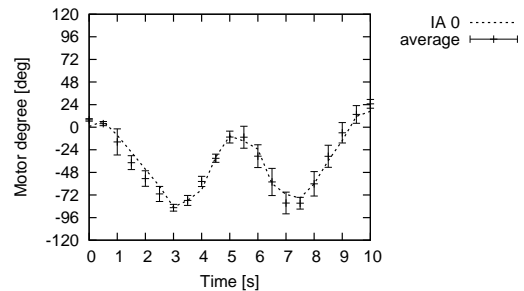


Fig. 14. Output angles and average input in this experiment. The output angle is the case when the starting angle is 0° (IA 0 in Fig. 13). Average is the average shown in Fig. 11.

Figures 13 and 14 show the results of this experiment. Fig. 13 shows the movement starting from the five initial angles. Each movement gradually approaches movement starting from 0° . The five movements converge in one movement at 3 s. In movement with an initial angle of 120° (IA 120), the angle of the motor overshoots. This overshoot is controlled by value of the virtual weight of State Point m . The appropriate value of m depends on the robot to which it is being applied in this system. In this experiment, we allow the robot to move violently, but we can also operate the robot slowly by adjusting certain parameters. When the initial angle is -120° (IA -120), the robot stops during the first 0.5 s because of the absence of cells with a higher selection frequency in the range of r . r is the number of cells in the range of candidate cells. In a case where the initial angle is -60° (IA -60), a difference in movement is observed from 6.5 s to 8 s, as seen in Fig. 13. In this case, the servomotor is stopped by reception error in current angle data at 6.5 s. In this experiment, hardware problems sometime occur because we use an actual robot. If the system receives an abnormal value, the servomotor that sends the abnormal value is stopped for safety reasons by the system. The servomotor then restarts movement at 7 s. At 8 s, this movement is as stable as the other movement.

Finally, we compare input and output in Fig. 14, where "IA 0" is output with an initial angle of 0° in Fig. 13 and

"average" is the average and standard deviation of input. The standard deviation range is an indicator of the accuracy of output. The system generates movement where standard deviation is large. The system generates movement more closely where standard deviation is small. This means that the system begins robot operation closely in the region where the demand of the user is clear. There are some places, however, where output is outside of the standard deviation range. This is because the system generates movement considering the previous state of the robot. Robot movement that the user requests is performed gently so that the robot does not receive an excessive load. In this experiment, we set the parameters of the system. The farther the State Point is away from the target cell, the higher the attraction power. Hence, like the area from 3 s to 5 s in Fig. 14, the robot state may be more attracted even if standard deviation is large. We regulate the power balance using the distance and size of the selection frequency changing α value.

5. Conclusions

In this paper, we have proposed "Motion Space," a method for expressing robot movement. This method of expression enables robot movement to be generated based on the idea of attracting motion. We have explained how to use Motion Space in a robot. We have explained the system that uses Motion Space. We divided our system into two parts. The first part is the knowledge creation part, which creates Motion Space based on robot movement. The other part is the movement generation part, which generates robot movement based on Motion Space and sensor input. We then applied this system to the robot to show how the robot is operated by using our method. The system has been able to reproduce movement that the user has desired. Further, the robot has moved regardless of its state when it starts moving.

We have compared this method with other methods. Using this method, we have operated the robot in the manner that likes teaching playback. Using this method, the robot has generated movement robustly because the robot acquires knowledge based on movement taught by Motion Space. We do not need to use complicated dynamic formulas for designing movement such as those needed in methods to design knowledge and learning methods such as ALM. Motion Space incorporates information on past movement and saves it. We have thus realized the knowledge of movement that we wanted.

Based on the above results, we have considered the type of robot that is suitable for this method. We can use Motion Space method as a technique such as teaching playback in factories. Specifically, we think that this method will be effective in a factory where a product often changes. In addition, Motion Space is available for automating the operation of a radio control robot, for example: when the signal from an operator is cut off, the robot returns to the operator using past operation data saved in Motion Space.

In future work, we will treat various sensors as the state of the robot. Using sensor information on the knowledge of movement, it will become easy for a robot to learn movement adapting change of environment. It is necessary to consider a method for effectively expressing Motion Space with little information because big Motion Space is necessary for learning the exact movement in our method.

References:

- [1] H. Asada and Y. Asari, "The Direct teaching of tool manipulation skills via the impedance identification of human motion," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1269-1274, 1988.
- [2] A. Clark and R. Grush, "Towards a Cognitive Robotics," Adaptive Behavior, January 1999, Vol.7, No.1, pp. 5-16, 1999.
- [3] R. A. Brooks, "A robust layered control system for a mobile robot," IEEE J. of Robotics and Automation, Vol.RA-2, No.1, pp. 14-23, 1986.
- [4] S. Kotosaka and S. Schaal, "Synchronized robot motion by neural oscillators," J. of the Robotics Society of Japan, Vol.19, No.5, pp. 580-583, 2001 (in Japanese).
- [5] T. Kasuga and M. Hashimoto, "Human-Robot Handshaking using Neural Oscillators," Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, pp. 3802-3807, April 18-22, 2005.
- [6] M. Okada, K. Tatani, and Y. Nakamura, "Polynomial Design of the Nonlinear Dynamics for the Brain-Like Information Processing of Whole Body Motion," Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation, Vol.2, pp. 1410-1415, 2002.
- [7] M. Okada and M. Watanabe, "Controller Decomposition and Combination Design of Body / Motion Elements based on Orbit Attractor," Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation, pp. 1364-1369, May 12-17, 2009.
- [8] Y. Sakurai, N. Honda, and J. Nishino, "Acquisition of Knowledge for Gymnastic Bar Action by Active Learning Method," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.7, No.1, pp. 10-18, 2003.



Name:

Kentarou Kurashige

Affiliation:

Research Associate, Department of Computer Science and System Engineering, Muroran Institute of Technology

Address:

27-1 Mizumoto-cho, Muroran-city, Hokkaido 050-8585, Japan

Brief Biographical History:

2002-2005 Research Associate, Fukuoka University
2005- Research Associate, Muroran Institute of Technology

Main Works:

- K. Kurashige, T. Fukuda, and H. Hoshino, "Reusing primitive and acquired motion knowledge for Gait generation of Six-Legged Robot using Genetic Programming," J. of Intelligent and Robotic systems, Vol.38, Kluwer Academic Pub., pp. 121-134, 2003.
- "From Automation To Autonomy," Machine Learning, A. Mellouk and A. Chebira (Eds.), Published by In-Tech, pp. 39-52, Feb. 2009.
- Y. Miyazaki and K. Kurashige, "Estimate of current state based on experience in POMDP for Reinforcement Learning," Proc. of the seventeenth Int. Symposium on Artificial Life and Robotics (AROB 17th '12), pp. 1135-1138, Jan. 19-21, 2012, B-Con Plaza, Beppu, Oita, Japan, 2012.

Membership in Academic Societies:

- The Robotics Society of Japan (RSJ)
- The Japanese Society for Artificial Intelligence (JSAI)
- Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)



Name:

Naoki Kitayama

Affiliation:

Muroran Institute of Technology

Address:

27-1 Mizumoto-cho, Muroran-city, Hokkaido 050-8585, Japan

Brief Biographical History:

2011 Graduated from Department of Computer Science and Systems Engineering, Muroran Institute of Technology

2011- Master's Course, Division of Information and Electronic Engineering, Muroran Institute of Technology

Main Works:

• N. Kitayama and K. Kurashige, "Proposal of method "Motion Space" to express movement of the robot," Proc. of IWACHII2011 CD-ROM, GS1-3, November 19-23, 2011, Suzhou, China, 2011.



Name:

Masafumi Kiyohashi

Affiliation:

Sun Information & Service, Co, Ltd.

Address:

3-15-9 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Brief Biographical History:

2008 Graduated from Department of Computer Science and Systems Engineering, Muroran Institute of Technology
