Estimation of Structure and Physical Relations among Multi-modal Sensor Variables for Musculoskeletal Robotic Arm

Kentaro Harada[1] and Yuichi Kobayashi[2]

***Abstract*— Autonomous robots that work in the same environment as humans must operate safely and adapt to handle various tools and deal with partial malfunctions. We propose an approach for estimating the robot structure and apply this approach for building a controller of dynamic motions. The robot structure is estimated by evaluating the mutual information (MI) among the sensor variables. Variables with high values of MI are edge-connected and the controller is automatically constructed based on the estimated structure. The proposed approach can accommodate changes in the robot parameters and dynamic motions. We verify the proposed method by using a simulator of a musculoskeletal arm driven that is driven by artificial muscle for mechanical safety.**

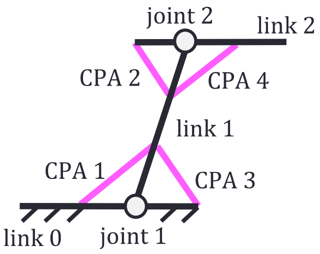

Fig. 1. A model of a musculoskeletal arm.

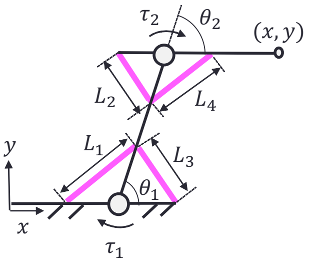

Fig. 2. Variables of arm robot

I. Introduction

Autonomous robots ysed for housework and nursing care have been attracting much attention. It is preferred that those robots have mechanical safety for surroundings and adaptivity to handle various tools and to overcome their partial malfunctions. For mechanical safety, soft material ($e.g.$, [1]) and elastic actuators ($e.g.$, [2], [3], [4]) have been developed, and many attempts have been made to enhance the robots' ability to adapt to changes of robot environment and robot body itself.

Machine learning by approximating the robot dynamics has been applied to control the robot arm ($e.g.$, [5], [6]). These approaches enable robots to adapt to unknown dynamics. However, one shortcoming of this approaches is that the robots cannot effectively adapt to partial changes such as tool use and malfunction, because the learning process has to restart from scratch. Thus, learning will be more effective if the robots can utilize information about property of the system that is valid even though dynamics is changed.

From the perspective of learning unknown tools, several approaches have been proposed [7], [8], [9]. These approaches realize acquisition of tool properties based on robots ’ observation and experience, but regard tool attachments as a factor that is not a part of the learning. If a learning model includes tool attachment and desorption, it will be more generally applicable and will not require human intervention.

Understanding the structure of a robot is an important building block for realizing a seamless learning framework that is adaptive for both of tool use and partial malfunction. Simultaneous motion of a robot and an object was used to build a chain of controllable parts in the robot in [10]. This realizes a unified method for motion generation to avoid obstacles, push objects, and use tools. Furthermore, learning of the kinematic structure of a robot arm using a Bayesian network was proposed in [11]. It has been shown that the system can adapt to the lack of observations and observation noise. However, a common shortcoming of these approaches [10], [11] is that they only deal with kinematic relations among observed variables and hence their application is restricted to quasi-static motion.

In this study, we present an approach for understanding the robot structure and for building controller for dynamic motion of robot. In the proposed approach, the relations among sensor variables are evaluated using mutual information (MI). The relations between sensor variables with high MI are approximated using observation. These relations are used for building the controller based on a chain-construction algorithm.

We consider the musculoskeletal arm robot shown in Fig. 1. The artificial muscle is for the safe of the robot because of its elasticity. We verify the proposed method in applications to tendon-driven musculoskeletal arm to show its applicability.

II. Problem Definition

A. Structure of the arm robot

We discuss the control of the musculoskeletal arm robot shown in Fig. 1. This robot arm has two joints with three links and the base link is fixed. Each of the two joints is actuated by two artificial muscles. The coiled polymer actuator (CPA) [4] is the model of the artificial muscle and the latter responds to heat by contracting. This actuator is generally heated by passing electrical current through wires.

[1] Kentaro Harada, Graduate school of integrated science and technology, Shizuoka University, Hamamatsu, Japan harada.kentaro.16@shizuoka.ac.jp

[2] Yuichi Kobayashi, Department of Mechanical Engineering, Shizuoka University, Hamamatsu, Japan kobayashi.yuichi@shizuoka.ac.jp

TABLE I

SENSOR VARIABLES

$\dot{\boldsymbol{\theta}}$ [rad/s]	$\boldsymbol{d}_1 \in \mathbb{R}^2$	$\boldsymbol{\tau}$ [N/m]	$\boldsymbol{d}_7 \in \mathbb{R}^2$
$\dot{\boldsymbol{x}}$ [m/s]	$\boldsymbol{d}_2 \in \mathbb{R}^2$	$\dot{\boldsymbol{L}}$ [m/s]	$\boldsymbol{d}_8 \in \mathbb{R}^4$
$\boldsymbol{\theta}$ [rad]	$\boldsymbol{d}_3 \in \mathbb{R}^2$	$\boldsymbol{L}$ [m]	$\boldsymbol{d}_9 \in \mathbb{R}^4$
$\boldsymbol{x}$ [m]	$\boldsymbol{d}_4 \in \mathbb{R}^2$	$\ddot{\boldsymbol{L}}$ [m/s^2]	$\boldsymbol{d}_{10} \in \mathbb{R}^4$
$\ddot{\boldsymbol{\theta}}$ [rad/s^2]	$\boldsymbol{d}_5 \in \mathbb{R}^2$	$\boldsymbol{F}$ [N]	$\boldsymbol{d}_{11} \in \mathbb{R}^4$
$\ddot{\boldsymbol{x}}$ [m/s^2]	$\boldsymbol{d}_6 \in \mathbb{R}^2$	$\boldsymbol{T}$ [°C]	$\boldsymbol{d}_{12} \in \mathbb{R}^4$

However, in this case, we use the temperature of muscles to control the robot for simplicity.

The sensor variables are listed in TABLE. I. $\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{L}$, $\boldsymbol{\tau}$, $\boldsymbol{F}$ and $\boldsymbol{T}$ denote the position of the end-effector, the angle of each joint, the muscular length, the torque applied at the joints and muscular force, and the temperature of each muscle, respectively. Noise is not added on each sensor variable. These sensors are configured redundantly assuming that the robot is designed for robustness. The arrangement of the variables on the robot arm is shown in Fig. 2.

It is assumed that the physical meaning of each sensor variable is unknown to the robot except for the time-derivative information among the same sensor values ($e.g.$, relations among $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, and $\ddot{\boldsymbol{\theta}}$). This implies that the mechanical structure of the robot is also unknown. Additionally, the parameters related to the dynamics of the robot arm are unknown. Therefore, each sensor variable, which is written such as $\boldsymbol{x}$, $\boldsymbol{\theta}$, is treated as vectors $\boldsymbol{d}_1$, $\boldsymbol{d}_2$. The control input of the robot system is known (in this case, $\boldsymbol{\mu} = \boldsymbol{d}_{12} = \boldsymbol{T}$). $\boldsymbol{d}_1 = \dot{\boldsymbol{\theta}}$ and $\boldsymbol{d}_3 = \boldsymbol{\theta}$ are also known as the state variables. The robot system aims to generate a controller that realizes a desired value of the end-effector position $\boldsymbol{d}_{\mathrm{target}} = \boldsymbol{d}_4 = \boldsymbol{x}$. Thus, the task for the robot is to achieve $\boldsymbol{d}_{\boldsymbol{d}_4} = \boldsymbol{d}_{\mathrm{des}}$.

B. Robot arm dynamics

The dynamics of the robot arm is described as follows:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{\theta})$ and $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ denote the inertia matrix and a matrix including the centrifugal force, Coriolis force and friction, respectively. The gravity term $\boldsymbol{g}(\boldsymbol{\theta})$ is zero because the motion of the robot arm is assumed in the horizontal plane.

The relation between the control input $\boldsymbol{\mu} = \Delta\boldsymbol{T}$ and torque $\boldsymbol{\tau}$ is described as follows:

$$\boldsymbol{F} = \boldsymbol{A}\Delta\boldsymbol{T} + \boldsymbol{K}\Delta\boldsymbol{L} + \boldsymbol{C}\dot{\boldsymbol{L}} \tag{2}$$

$$\boldsymbol{\tau} = \boldsymbol{G}^T(\boldsymbol{\theta})\boldsymbol{F}, \tag{3}$$

where $\boldsymbol{A} = \mathrm{diag}(a_1, \ldots, a_4)$[N/℃], $\boldsymbol{K} = \mathrm{diag}(k_1, \ldots, k_4)$[N/m], $\boldsymbol{C} = \mathrm{diag}(c_1, \ldots, c_4)$[N/m·s], $\Delta\boldsymbol{T}$[℃] and ΔL[m] denote the thermal, elastic, and damping constants, the temperature change, and the displacement of muscular length from ordinary length, respectively. $\boldsymbol{G}(\boldsymbol{\theta})$ Is the Jacobian matrix that shows the relation between muscle length and joint angle defined as follows:

$$\dot{\boldsymbol{L}} = \boldsymbol{G}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \quad \boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^{4\times 2}. \tag{4}$$

The arm is actuated by the muscular temperature $\Delta\boldsymbol{T}$. The CPA will be torn off if it is excessively heated; therefore, $\Delta T_i,\ i = 1, \cdots, 4$ is constrained so that $\boldsymbol{T}_i$ does not exceed the maximum temperature, which is 200 °C in this case.

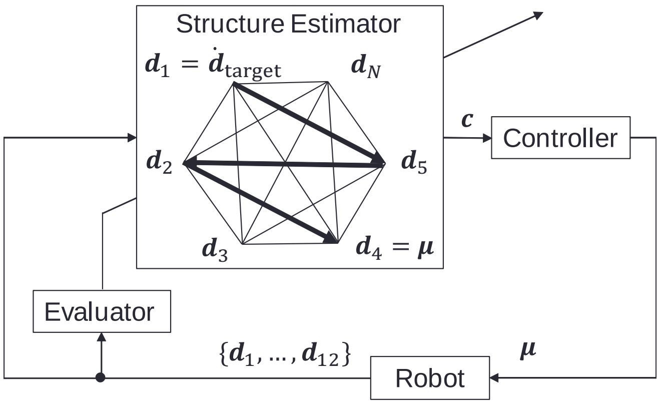

Fig. 3. Overview of the operation

III. Learning Control Including Structure Estimation

A. Overview of the structure estimation and control

To achieve the reaching action under the condition of unknown structure, we propose an automatic construction of a controller based on identifying the dependency among the variables. The overview of the proposed process is shown in Fig. 3.

First, the robot collects data from the sensor variables by moving randomly. The structure is estimated from the collected samples using the MI and a path-finding method. A controller is generated based on the estimated structure. The position control starts after the controller is generated. The generated controller is evaluated and updated online using the collected samples of the sensor variables. If the controller does not achieve the target, it is abandoned and a new controller is generated.

B. Least squares mutual information

Mutual Information (MI) is a measurement of the dependency among pairs of random variables which provides estimates of the decreasing randomness of a variable by observing another variable. Different types of MI are defined by [12]. In this study, an approximate value of the squared-loss MI (SMI) by least squares MI (LSMI) [13] is used because SMI has moderate sensibility for dependency and does not require adjustable parameters. The definition of SMI $I(x, y)$ between two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is given as follows :

$$I(\boldsymbol{x},\boldsymbol{y}) = \iint p(\boldsymbol{x})p(\boldsymbol{y})(r(\boldsymbol{x},\boldsymbol{y})-1)^2 d\boldsymbol{x}d\boldsymbol{y} \tag{5}$$

$$r(\boldsymbol{x},\boldsymbol{y}) = \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}, \tag{6}$$

where $I(\boldsymbol{x},\boldsymbol{y})$ denotes the MI between $\boldsymbol{x}$ and $\boldsymbol{y}$. $p(\boldsymbol{x},\boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$ denote the joint probability density function and marginal probability density functions of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. $r(\boldsymbol{x},\boldsymbol{y})$ is the probability density ratio. The characteristic of SMI is described as follows:

$$I(\boldsymbol{x},\boldsymbol{y}) \geqq 0 \tag{7}$$

$$I(\boldsymbol{x},\boldsymbol{y}) = 0 \Leftrightarrow p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}). \tag{8}$$

A high SMI suggests that the data in the pairs are highly dependent.

LSMI calculates an approximate value of SMI by estimating the probability density ratio $r(\boldsymbol{x},\boldsymbol{y})$ of (5) which is difficult to calculate directly. The probability density ratio is modeled as follows:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}), \tag{9}$$

where $\boldsymbol{\alpha}$ denotes the optional parameter vector and $\boldsymbol{\phi}$ denotes the non-negative basis function vector: $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ have the same dimensions as sample number of $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{\phi}$ is defined by the cross variation of the candidates. LSMI retrives $\boldsymbol{\alpha}$ according to:

$$\tilde{\boldsymbol{\alpha}}_{\mathrm{LSMI}} = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}}(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{h} + \lambda\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}) \tag{10}$$

$$\boldsymbol{H} = \frac{1}{d^2}\sum_{i,i'=1}^{d}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})^{\mathrm{T}} \tag{11}$$

$$\boldsymbol{h} = \frac{1}{d}\sum_{i=1}^{d}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})^{\mathrm{T}}, \tag{12}$$

where $\lambda > 0$ is the regularization parameter.

The general solution of (10) is described as follows:

$$\tilde{\boldsymbol{\alpha}}_{\mathrm{LSMI}} = (\boldsymbol{H} + \lambda\boldsymbol{I})^{-1}\boldsymbol{h}. \tag{13}$$

Using the solution of (13), LSMI is approximated as follows:

$$\tilde{I}(\boldsymbol{x},\boldsymbol{y}) = \tilde{\boldsymbol{\alpha}}_{\mathrm{LSMI}}^{\mathrm{T}}\boldsymbol{h} - 1. \tag{14}$$

If the LSMI value between sensor variables $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$ is high, it may be possible to infer that $\boldsymbol{d}_a$ can be controlled by manipulating the value of $\boldsymbol{d}_b$ or vice versa. Considering the both possibilities, chains of controller are constructed as described in the next section.

C. Structure estimation using LSMI and Dijkstra's Algorithm

The structure of the robot system is estimated by evaluating dependencies of the sensor variables. The controller is constructed as a "chain" starting from the derivative of the target variable $\dot{\boldsymbol{d}}_{\mathrm{target}}$ and terminates at the control input $\boldsymbol{\mu}$ based on information of the dependencies. The chain is obtained using Dijkstra's algorithm based on the LSMI values. Dijkstra's algorithm is a method used solve the minimum cost path problem [14].

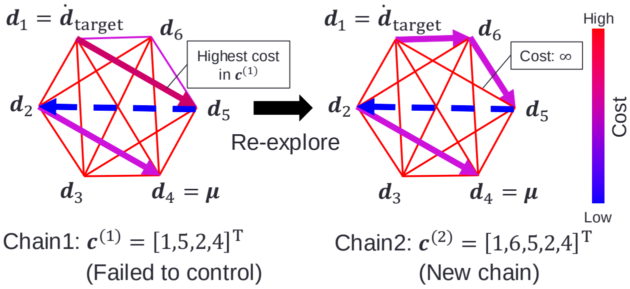
$\boldsymbol{d}_1 = \dot{\boldsymbol{d}}_{\mathrm{target}}$
$\boldsymbol{d}_6$
Highest cost in $\boldsymbol{c}^{(1)}$
$\boldsymbol{d}_2$
$\boldsymbol{d}_5$
$\boldsymbol{d}_3$
$\boldsymbol{d}_4 = \boldsymbol{\mu}$
Re-explore
$\boldsymbol{d}_1 = \dot{\boldsymbol{d}}_{\mathrm{target}}$
$\boldsymbol{d}_6$
Cost: ∞
$\boldsymbol{d}_2$
$\boldsymbol{d}_5$
$\boldsymbol{d}_3$
$\boldsymbol{d}_4 = \boldsymbol{\mu}$
High
Cost
Low
Chain1: $\boldsymbol{c}^{(1)}$ = [1,5,2,4]$^{\mathrm{T}}$
(Failed to control)
Chain2: $\boldsymbol{c}^{(2)}$ = [1,6,5,2,4]$^{\mathrm{T}}$
(New chain)

Fig. 4. Example of the resultant dependency graph. Derivative relations and chain are expressed by broken lines and arrows.

For constructing the controller, Dijkstra's algorithm is used to find a chain with large dependencies among the sensor variables. For this purpose, the dependency information is converted to cost. Let $\zeta_{a,b}$ denote the cost between sensor variables $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$. Costs are defined as follows:

$$\zeta_{a,b} = \begin{cases} 0, & \text{if } \boldsymbol{d}_b \text{ is derivative of } \boldsymbol{d}_a \\ \infty, & \text{if } \boldsymbol{d}_b \text{ is integral of } \boldsymbol{d}_a \\ \frac{1}{I(\boldsymbol{d}_a,\boldsymbol{d}_b)}, & \text{otherwise} \end{cases} . \tag{15}$$

As described in section II, the relation between sensor variables is assumed to be known when one variable is a time-the derivative of the other.

Let $\boldsymbol{c}^{(j)} \in \mathbb{N}^{P^{(j)}}$ denote a vector comprising indices of sensor variables in the chain (j), where $P^{(j)}$ denotes the number of the edges obtained by the above mentioned method.

$$\boldsymbol{c}^{(j)} = \left[c_1, c_2, \ldots, c_P\right]^{\mathrm{T}} \tag{16}$$

The first and last element of $\boldsymbol{c}$ become the data number of the derivative of the target variable and control input: thus, $\boldsymbol{d}_{\boldsymbol{c}_1} = \dot{\boldsymbol{d}}_{\mathrm{target}}$, $\boldsymbol{d}_{\boldsymbol{c}_P} = \mu$.

An example of the result is given in Fig. 4. Note that the number of sensor variables in Fig. 4 is not corresponding to TABLE I. The controller is generated from the chain by the method described in III-D. If the generated controller fails to converge the target variable $\boldsymbol{d}_{\mathrm{target}}$ to the desired value $\boldsymbol{d}_{\mathrm{des}}$, another chain is explored by setting the cost of the edge with the highest value in the current chain to infinity. In Fig. 4, the re-explored chain $\boldsymbol{c}^{(2)}$ directly connects $\boldsymbol{d}_1$ and $\boldsymbol{d}_5$, because of the updated cost of infinity and due to owing that $\zeta_{1,5}$ is the highest in the chain $\boldsymbol{c}^{(1)}$.

D. Generation of the controller

A controller is created from the constructed chain. For the control, relation between the two variables in the chain $\boldsymbol{c}^{(j)}$ is used. The rule for constructing the controller is described in Algorithm 1, where the notation of $\boldsymbol{c}^{(j)}$ is omitted for simplicity. In the algorithm, relation between variables $\boldsymbol{d}_{\boldsymbol{c}_i}$ and $\boldsymbol{d}_{\boldsymbol{c}_{i+1}}$ is estimated by the linear approximation $\boldsymbol{R}_{\boldsymbol{c}_i,\boldsymbol{c}_{i+1}}$, where $\boldsymbol{R}_{\boldsymbol{c}_i,\boldsymbol{c}_{i+1}}$ denotes the matrix of the linear approximation which relates $\left[\boldsymbol{d}_{\boldsymbol{c}_i}^{\mathrm{T}}, 1\right]^{\mathrm{T}}$ to $\boldsymbol{d}_{\boldsymbol{c}_{i+1}}$. The details of estimating $\boldsymbol{R}_{\boldsymbol{c}_i,\boldsymbol{c}_{i+1}}$ are described in III-E.

Algorithm 1 Generate the controller

```
Input: chain $c$,
Initialize: control input $\mu \leftarrow d_{\mathrm{des}} - d_{\mathrm{target}}$
for $i = 1 : P$ do
    Check the relation between $c_i$ and $c_{i+1}$
    if Relation is unknown then
        $\mu \leftarrow R_{c_i, c_{i+1}} \left[\mu^{\mathrm{T}}, 1\right]^{\mathrm{T}},\ (d_{c_{i+1}} = R_{c_i, c_{i+1}} d_{c_i})$
    else if Relation is derivative then
        $\mu \leftarrow \mu - d_i$
    end if
end for
```

Here we give an example with $\boldsymbol{c} = [1, 5, 2, 4]$ and the known relationship that $\boldsymbol{d}_2$ is the derivative of $\boldsymbol{d}_5$, as shown in Fig. 4. First, the control input $\boldsymbol{\mu}$ is set as $\boldsymbol{\mu} = \boldsymbol{d}_{\mathrm{des}} - \boldsymbol{d}_{\mathrm{target}}$, thus omitting the feedback gain matrix. Then, we check the relation between $\boldsymbol{d}_{\boldsymbol{C}_1} = \boldsymbol{d}_1$ and $\boldsymbol{d}_{\boldsymbol{C}_2} = \boldsymbol{d}_5$, and the relational matrix $\boldsymbol{R}_{1,5}$ is multiplied to the current control input $\boldsymbol{\mu} = \boldsymbol{R}_{1,5}(\boldsymbol{d}_{\mathrm{des}} - \boldsymbol{d}_{\mathrm{target}})$ because the relation is unknown. We then check the next relation between $\boldsymbol{d}_{\boldsymbol{C}_2} = \boldsymbol{d}_5$ and $\boldsymbol{d}_{\boldsymbol{C}_3} = \boldsymbol{d}_2$, which is relation of derivative, and the control input becomes $\boldsymbol{\mu} = \boldsymbol{R}_{1,2}(\boldsymbol{d}_{\mathrm{des}} - \boldsymbol{d}_{\mathrm{target}}) - \boldsymbol{d}_5$. For the relation between $\boldsymbol{d}_{\boldsymbol{C}_3} = \boldsymbol{d}_2$ and $\boldsymbol{d}_{\boldsymbol{C}_4} = \boldsymbol{d}_4$, the controller is built same as in between $\boldsymbol{d}_1$ and $\boldsymbol{d}_5$. Finally, the control input becomes $\boldsymbol{\mu} = \boldsymbol{R}_{2,4}\{\boldsymbol{R}_{1,2}(\boldsymbol{d}_{\mathrm{des}} - \boldsymbol{d}_{\mathrm{target}}) - \boldsymbol{d}_5\}$.

E. Local estimation of unknown relation

For the online estimation of the unknown relation $\boldsymbol{R}$ used in the controller, $\boldsymbol{R}$ is locally estimated by the locally weighted regression (LWR) [15] that weights the data samples on the basis of the Euclidean distance between each sample and query input because $\boldsymbol{R}$ is generally non-linear, as in the case of the Jacobian in (4). The relation between the pair of sensor variables $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$ is approximated by the locally linear relation as follows:

$$\boldsymbol{d}_b = \boldsymbol{R}_{a,b} \left[\boldsymbol{d}_a^{\mathrm{T}}, 1\right]^{\mathrm{T}},\ \boldsymbol{R} \in \mathbb{R}^{\beta \times (\alpha+1)}, \tag{17}$$

where α and β denote the column size of $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$, respectively. The element of one under $\boldsymbol{d}_a$ is added to include an the offset parameter in the linear approximation.

To apply least squares regression, Eq. (17) is transformed as follows:

$$\boldsymbol{D}_b = \boldsymbol{D}_a \boldsymbol{r}_{a,b}, \quad \boldsymbol{r}_{a,b} \in \mathbb{R}^{(\alpha+1)\beta \times 1} \tag{18}$$

$$\boldsymbol{D}_a = \left[\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_n\right]^{\mathrm{T}} \in \mathbb{R}^{n\beta\ \times\ (\alpha+1)\beta} \tag{19}$$

$$\boldsymbol{D}_b = \left[\boldsymbol{d}_b^{(1)\mathrm{T}}, \ldots, \boldsymbol{d}_b^{(n)\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{n\beta\ \times\ 1} \tag{20}$$

$$\boldsymbol{\Phi}_n = \underbrace{\left.\begin{bmatrix} \boldsymbol{d}_a^{(1)\mathrm{T}} & 1 & & \ldots & \boldsymbol{O} \\ \vdots & & \ddots & & \vdots \\ \boldsymbol{O} & \ldots & & \boldsymbol{d}_a^{(n)\mathrm{T}} & 1 \end{bmatrix}\right\}\beta,}_{(\alpha+1)\beta} \tag{21}$$

TABLE II
PARAMETERS OF THE ROBOT ARM

l [m]	1	l_{g} [m]	$\frac{1}{2}l_2$
m [kg]	1	a [N/℃]	1
k [N/m]	1	c [N/m·s]	1

where $\boldsymbol{r}_{a,b}$, $\boldsymbol{D}_a$, $\boldsymbol{D}_b$ and n denote the column vector of the regression coefficient that aligns each line of $\boldsymbol{R}_{a,b}$ into one column, the data matrix of $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$, and the number of samples, respectively. $\boldsymbol{d}^{(n)}$ is the n-th sample in $\boldsymbol{d}$. The regression coefficient $\boldsymbol{r}_{a,b}$ is estimated as follows:

$$\boldsymbol{r}_{a,b} = (\boldsymbol{D}_a^{\mathrm{T}} \boldsymbol{W} \boldsymbol{D}_a)^{-1} \boldsymbol{D}_a^{\mathrm{T}} \boldsymbol{D}_{\boldsymbol{b}}. \tag{22}$$

$$\boldsymbol{W} = \mathrm{diag}[\underbrace{w_1, \cdots, w_1}_{\beta}, \cdots, \underbrace{w_n, \cdots, w_n}_{\beta}] \in \mathbb{R}^{n\beta \times n\beta} \tag{23}$$

where $\boldsymbol{W}$ denotes the weight matrix. Each weight is defined as follows:

$$w_n = \exp\left(-\frac{1}{2}h(\boldsymbol{s}^{(n)} - \boldsymbol{s}_{\mathrm{query}})^{\mathrm{T}}(\boldsymbol{s}^{(n)} - \boldsymbol{s}_{\mathrm{query}})\right) \tag{24}$$

$$\boldsymbol{s} = [\boldsymbol{\theta}^{\mathrm{T}},\ \dot{\boldsymbol{\theta}}^{\mathrm{T}}]^{\mathrm{T}} \tag{25}$$

where $\boldsymbol{s}^{(n)}$, $\boldsymbol{s}_{\mathrm{query}}$ and h denote the state vector $\boldsymbol{s}$ when sample $\boldsymbol{d}^{(n)}$ was collected, the current state vector, and the scaling parameter of the weighting range, respectively. Elements of $\boldsymbol{r}_{ab}$ are reorganized to the original shape of $\boldsymbol{R}_{a,b}$ for estimation by Eq. (17). h is decided from prepared candidates by the leave-one-out cross-validation method that chooses one candidate that associates the minimal squared error between the current sensor data $\boldsymbol{d}_b$ and the estimated value by Eq. (17). The total number of samples is fixed to N_{sample}, and the oldest sample is discarded whenever a new sample is collected after reaching the maximum number.

F. Evaluation of the controller

To look for new chain and to quit the current one, the generated controller is evaluated by performing 100 s simulations and checking whether the robot arm follows desired position in time or not. All chains that were found with Dijkstra's algorithm are evaluated although some of them achieve to follow the target position.

IV. EXPERIMENT

A. Experimental conditions

The control method is verified by using a simulator. The parameters of the robot arm are listed in Table II. l, l_g, and m denote the length of each link, the length from the gravity point to the end of the link, and the mass of each link, respectively. The control cycle is 0.05 s. The maximum number for storing the samples for LSMI and LWR is $N_{\mathrm{sample}} = 300$.

B. Result of structure estimation

The relational costs among the sensor variables are shown in TABLE III. All 27 chains obtained with with Dijkstra's algorithm are shown below. The chains that achieved position control are shown with $\diamond$.

- $\boldsymbol{c}^{(1)} = [2, 6, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(2)} = [2, 6, 10, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{L}} \rightarrow \boldsymbol{T})$

TABLE III

RELATIONAL COSTS AMONG SENSOR VARIABLES

from \ to	$\dot{\theta}$	$\dot{x}$	θ	x	$\ddot{\theta}$	$\ddot{x}$	τ	$\dot{L}$	L	$\ddot{L}$	F	T	colorbar
$\dot{\theta}$		0.5	∞	0.5	0.0	6.3	1.5	0.4	1.1	0.4	1.3	1.4	3
$\dot{x}$	0.5		0.6	∞	1.3	0.0	0.7	0.5	79.4	0.6	0.7	0.7	
θ	0.0	0.6		0.2	0.0	0.8	0.7	0.4	0.5	0.7	0.8	0.8	
x	0.5	0.0	0.2		0.6	0.0	9.7	0.4	0.4	3.3	3.6	46.6	
$\ddot{\theta}$	∞	1.3	∞	0.6		1.0	1.3	0.5	131	0.4	3.0	1.2	2
$\ddot{x}$	6.3	∞	0.8	∞	1.0		1.4	1.6	466	0.5	0.8	0.8	
τ	1.5	0.7	0.7	9.7	1.3	1.4		0.5	9.8	0.5	0.5	1.0	
$\dot{L}$	0.4	0.5	0.4	0.4	0.5	1.6	0.5		∞	0.0	0.6	0.6	
L	1.1	79.4	0.5	0.4	131	466	9.8	0.0		0.0	2196	2236	1
$\ddot{L}$	0.4	0.6	0.7	3.3	0.4	0.5	0.5	∞	∞		0.5	0.5	
F	1.3	0.7	0.8	3.6	3.0	0.8	0.5	0.6	2196	0.5		0.2	
T	1.4	0.7	0.8	46.6	1.2	0.8	1.0	0.6	2236	0.5	0.2		0

- $\boldsymbol{c}^{(3)} = [2, 6, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(4)} = [2, 8, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(5)} = [2, 6, 10, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{L}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(6)} = [2, 8, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(7)} = [2, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(8)} = [2, 3, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(9)} = [2, 3, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(10)} = [2, 6, 10, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{L}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- ◇ $\boldsymbol{c}^{(11)} = [2, 1, 5, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \ddot{\boldsymbol{\theta}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(12)} = [2, 1, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(13)} = [2, 1, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(14)} = [2, 3, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(15)} = [2, 6, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \ddot{\boldsymbol{x}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- ◇ $\boldsymbol{c}^{(16)} = [2, 1, 5, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \ddot{\boldsymbol{\theta}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(17)} = [2, 1, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(18)} = [2, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(19)} = [2, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(20)} = [2, 8, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- ◇ $\boldsymbol{c}^{(21)} = [2, 1, 5, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \dot{\boldsymbol{\theta}} \rightarrow \ddot{\boldsymbol{\theta}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(22)} = [2, 3, 4, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{x} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(23)} = [2, 3, 9, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(24)} = [2, 3, 4, 7, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{x} \rightarrow \boldsymbol{\tau} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(25)} = [2, 3, 4, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{x} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(26)} = [2, 3, 9, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{T})$
- $\boldsymbol{c}^{(27)} = [2, 3, 9, 11, 12]$ $(\dot{\boldsymbol{x}} \rightarrow \boldsymbol{\theta} \rightarrow \dot{\boldsymbol{L}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{T})$

There are enormous pattern of edges that connect $\boldsymbol{d}_2$ and $\boldsymbol{d}_{12}$. However, chains are narrowed in 27 patterns because the edges that has infinity cost is added every time a new chain is constructed. Although chain 1, 2, 3, 5, 10, 11, 15, 16, and 21 seem to plausibly associates physical meanings of the variables, other chains cannot be protested as not plausible because every variable is related indirectly.

C. Result of Position Control

Position control was achieved when using controllers generated from chain $\boldsymbol{c}^{(11)}$,$\boldsymbol{c}^{(16)}$, and $\boldsymbol{c}^{(21)}$. However calculation cycle of LWR exceeds the real time of control cycle. The trajectory of the end-effector while using the controller generated from chain $\boldsymbol{c}^{(16)}$ is shown in Fig. 5. The generated controller from $\boldsymbol{c}^{(16)}$ is described as follows:

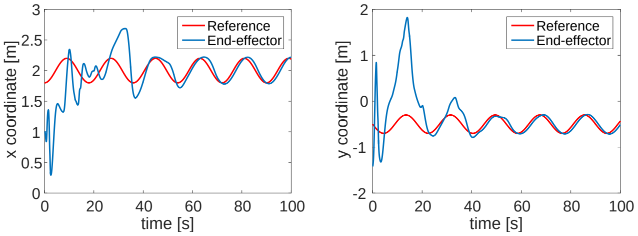

Fig. 5. x & y trajectory of end-effector controlled by $\boldsymbol{c}^{(16)}$

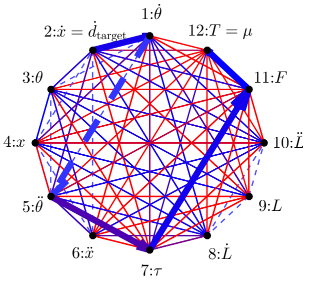

Fig. 6. Dependency graph: $\boldsymbol{c}^{(16)}$

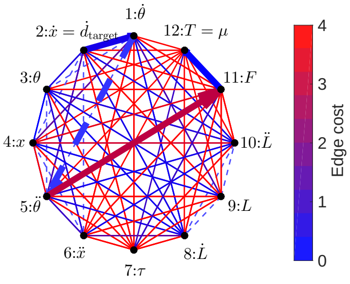

Fig. 7. Dependency graph: $\boldsymbol{c}^{(21)}$

$$\boldsymbol{\mu} = \boldsymbol{R}_{11,12}\boldsymbol{R}_{7,11}\boldsymbol{R}_{5,7}\{\boldsymbol{R}_{2,1}(\boldsymbol{d}_{\rm des} - \boldsymbol{d}_{\rm target}) - \boldsymbol{d}_1\}. \quad (26)$$

Eq. (26) is interpreted as following equation:

$$\boldsymbol{\mu} = \boldsymbol{A}^{-1}(\boldsymbol{G}^{\rm T}(\boldsymbol{\theta}))^{-1}\boldsymbol{M}(\boldsymbol{\theta})\{\boldsymbol{J}^{-1}(\boldsymbol{\theta})(\boldsymbol{x}_{\rm des} - \boldsymbol{x}) - \dot{\boldsymbol{\theta}}\}, \quad (27)$$

Eq. (27) is equivalent to the conventional musculoskeletal controller [16]: thus the proposed method successfully generated a reasonable controller.

The dependencies of $\boldsymbol{c}^{(16)}$ and $\boldsymbol{c}^{(21)}$ are shown in Fig. 6 and 7, respectively. A chain not including $\boldsymbol{d}_7 = \boldsymbol{\tau}$ which is included in $c^{(16)}$ is found in Fig. 7. This result indicates that the robot will keep controlling the position of the end-effector by finding alternative chains if sensor malfunctions, *e.g.*, $\boldsymbol{d}_7$ happens and affects the quality of $\boldsymbol{R}_{5,7}$ and the control.

D. Adaption for change of arm parameters

Adaptivity of the proposed method is verified by changing parameters of link 2 in TABLE IV in 50 s while the end-effector position is being controlled by chain $\boldsymbol{c}^{(16)}$. The change in the appearance of the robot arm is shown in Fig. 8. This condition considers the case of the robot arm holding a tool. In addition, we compare the proposed method with the case of relearning from scratch.

The trajectory of the end-effector is shown in Fig. 9. The end-effector position eventually converges to the desired trajectory even though the parameters have changed. The proposed method converges faster than relearning from scratch because it is not relearning all the approximated relations but relearning the ones affected by the changes, in this case $\boldsymbol{R}_{1,2}$ and $\boldsymbol{R}_{5,7}$.

TABLE IV

CHANGED PARAMETERS OF LINK 2

	before	after
l_2 [m]	1	1.5
m_2 [kg]	1	3
l_{2g} [m]	$\frac{1}{2}l_2$	$\frac{2}{3}l_2$

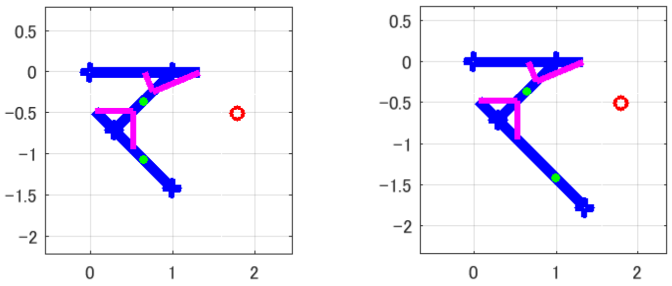

Fig. 8. Change of parameters on the robot arm. Blue and pink lines denote links and muscles of the robot arm. Green dots on links denote gravity points of each movable link. Red circle is the target position.

V. Conclusions and Future Work

We presented an approach for estimating the robot structure and building controllers of the dynamic motion of robots in order to adapt for changes. The structure was estimated from observing sensor variables by using MI and a path-finding method. The controller was built based on the estimated structure. We applied the proposed method to a musculoskeletal arm simulator and verified the behaviour of the generated controllers. As a result, Adaptivity of proposed method for change of structure and parameters was proved.

However, the proposed method is not free of problems in practice. The calculation cost of LWR must become shorter than control cycle. The locally weighted projection regression(LWPR) can deal with this problem because it keeps local models approximated from a part of whole data samples in memory [17]. The controller evaluation method needs to be automated by the stability analysis method or the evaluation of the trajectory of the target variable.

In addition, we need to investigate the more efficient use of the estimated structure. Neuralnetworks and reinforcements are implemented to learn the motion of robots (*e.g.*, [18], [19]); however these method require large training data and time-consuming for relearning because of the difficulty in revising the learned model. The proposed method possibly have lower relearning costs by revising one part of estimated structure where approximation accuracy became worth by change of structure and reusing other part.

Acknowledgement

The authors would like to thank Prof. K. Takagi of Nagoya university for useful advice about model of tendon-driven manipulator. This work was partly supported by JSPS KAKENHI (17K00362).

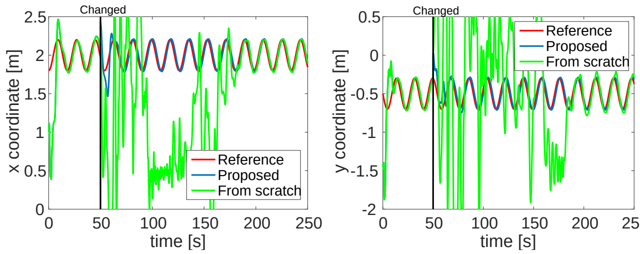

Fig. 9. Trajectory of end-effector when parameter is changed in 50s

References

[1] T. Odashima, M. Onishi, K. Tahara, K. Takagi, F. Asano, Y. Kato, H. Nakashima, Y. Kobayashi, T. Mukai, Z. Luo, S. Hosoe, “A soft human-interactive robot RI-MAN”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[2] X. Li, Y. Pan, G. Chen, H. Yu, “Adaptive HumanRobot Interaction Control for Robots Driven by Series Elastic Actuators”, IEEE Transactions on Robotics 33.1, pp. 169-182, 2017

[3] K. C. Wickramatunge, T. Leephakpreeda, “Study on mechanical behaviors of pneumatic artificial muscle”, International Journal of Engineering Science 48.2, pp. 188-198, 2010

[4] C. Yip, G. Niemeyer, “High-performance robotic muscles from conductive nylon sewing thread”. Proc. IEEE International Conference and Robotics and Automation, pp. 2313-2318, 2015

[5] J. Nakanishi, S. Stefan, “Feedback error learning and nonlinear adaptive control”, Journal of the International Neural Network Society 17.10, pp. 1453-1465, 2004

[6] D. Nguyen-Tuong, J. Peters, “Model learning for robot control: a survey”, Cognitive processing 12.4, pp. 319-340, 2011

[7] J. Sinapov, A. Stoytchev, “Detecting the Functional Similarities Between Tools Using a Hierarchical Representation of Outcomes”, Proc. IEEE International Conference on Development and Learning, pp. 91-96, 2008

[8] J. Sinapov, C. Scheenck, A. Stoytchev, “Learning relational object categories using behavioral exploration and multimodal perception”, Proc. IEEE International Conference on Robotics and Automation, pp. 5691-5698, 2014

[9] E. E. Aksoy, B. Dellen, F. Wörgötter, “Categorizing object-action relations from semantic scene graphs”, Proc. IEEE International Conference on Robotics and Automation, pp. 398-405, 2010

[10] Y. Kobayashi, S. Hosoe, “Planning-space shift motion generation: variable-space motion planning toward flexible extension of body schema”, Journal of Intelligent and Robotic Systems 62.3-4, pp. 467-500, 2011

[11] J. Sturm, C. Plagemann, W. Burgard, “Unsupervised body scheme learning through self-perception. In Robotics and Automation”, Proc. of IEEE International Conference on Robotics and Automation, pp. 3328-3333, 2008

[12] M. Sugiyama, K. Irie, M. Tomono, Machine Learning with Mutual Information and Its application in Robotics. Journal of the Robotics society of Japan, Vol.33, No.2, pp.86-91, 2015

[13] T. Suzuki, M. Sugiyama, T. Kanamori, J. Sese, ”Mutual information estimation reveals global associations between stimuli and biological processes ” , BMC Bioinformatics, 10(1):S52, 2009.

[14] E. W. Dijkstra, “A note on two problems in connexion with graphs”, Numerische mathematik 1.1, pp. 269-271, 1959

[15] C.G. Atkeson, A.W. Moore, S. Schaal. “Locally weighted learning for control”, Artificial Intelligence Review, 11(1), pp.75-113, 1997

[16] S. Wittmeier, et al, “Toward anthropomimetic robotics: development, simulation, and control of a musculoskeletal torso.” Artificial life 19.1, pp. 171-193, 2013

[17] S. Vijayakumar, A. D ’ Souza, S. Schaal, “Incremental online learning in high dimensions”, Neural Computation, 17(12), pp. 26022634, 2005.

[18] D. Shumsheruddin, “Neural network control of robot arm tracking movements”, Neural Network Applications, pp. 129-139, 1992

[19] J. Peters, S. Schaal, “Reinforcement learning by reward-weighted regression for operational space control”, Proc. of the 24th international conference on Machine learning, pp. 745-750, 2007.