2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Generalization of Bernstein's Problem toward Autonomous Action Development of Artificial Muscle Based Robots

Masakazu Suzuki[1]

Abstract— **The Bernstein's problem, asking how to cope with the redundant degrees-of-freedom of musculoskeletal systems in motion control, is straightforwardly extended to more general redundancy resolution problems. What element actions should be coordinated to achieve a complex task, or to realize multi-robot cooperative work? In this article they are unified and formulated as Generalized Bernstein Problem for developmental realization of complex actions, and then the *ease* criterion toward a smart solution for the problem is proposed.**

I. Introduction

How can one effectively control the musculoskeletal system with redundant degrees of freedom (DOF) and realize desirable motions? This question is known as the Bernstein's problem on motor control[1]. It is barely possible to avoid considering the problem for industrial robots of relatively simple link mechanism with conventional geared-motor based joints, since the motion specification is relatively clear and the relation between the target joint angle and the corresponding motor axis rotation angle is simple. But it is impossible to avoid the problem of redundant DOF for artificial muscle based human friendly robots, especially for robots with biarticular muscles and/or multi-DOF joints with redundant muscles. The Bernstein's problem has been investigated based on dynamics study mainly for basic actions such as reaching and walking[9],[2]. By the way the problem of redundant DOF is not limited within the control of musculoskeletal systems. Which actions should be combined and/or merged to realize a complex task? How should a robot team cooperate to accomplish a large scale task? These problems are also formulated in a very similar way. In this work, therefore, these are formulated in a unified expression and referred to as the Generalized Bernstein Problem, GBP. By applying the human wisdom utilized in sophisticatedly accomplishing large scale complex tasks, it is expected to better understand the control methodology of musculoskeletal systems by the brain and nervous system, whose analysis and formulation might be quite difficult. Or the wisdom may be used as a substitution, or merged to the method by brain and nervous system.

The mammals' antagonistic muscle based joint mechanism has excellent features compared to conventional robotic joints. The McKibben artificial muscle[4] and other various artificial muscles[13],[14] are therefore vigorously investigated. The author's group has developed another type of vertebrates' muscle like actuator: Strand-Muscle Actuator (StMA), and applied it to joint control of some robots[8],[11].

[1]M. Suzuki is with Faculty of Engineering, Tokai University, Hiratsuka, Japan suzuki at keyaki.cc.u-tokai.ac.jp

On the other hand, many learning methodologies for autonomous realization of complex actions have been proposed. The Intelligent Composite Action Control[3],[5],[10] (ICAC) used in this work is one of such methodologies. In the ICAC all the actions, from fundamental movement to complex tasks, are represented in a unified manner and realized step by step from the bottom by repeating optimal action composition. Based on the methodology various complex actions such as legged robot soccer action and cooperative soccer actions by several wheeled robots have been realized. In this article the Generalized Bernstein Problem is formulated for developmental realization of complex actions of Strand-muscle based robots through the ICAC.

The contents of the article are as follows: In section II the method for developmental realization of complex actions of artificial muscle based robots through the ICAC is outlined as preliminaries. The basic features of StMA and the joint control scheme with the StMAs are described first. Next the ICAC is briefly introduced and it is shown how a wide variety of actions from muscle extension/contraction to cooperative soccer actions by multiple robots are handled in a unified manner. In section III, the main section, the Bernstein's problem for motor control is generalized so that a wide variety of action realization problem based on the ICAC can be handled within a unified formulation. The Generalized Bernstein Problem is formulated, and as a simple example, the joint rotation of StMA-based 3-DOF joint with redundant muscles is investigated.

II. Developmental Action Realization of Artificial Muscle Based Robots

A. Actions of StMA-based robots

1) Strand-muscle actuator and its control scheme: A Strand-Muscle Actuator (StMA) is composed of a motor and a strand-muscle that consists of two or more muscle fibers (Fig.1, left). The motor side end is referred to as the actuator's driving end, and the other side is the effected end. Twisting the fiber strand by the motor and straining it with tension, the muscle contracts. Joint rotation is then realized by antagonistically mounted plural actuators as in Fig.1, center. A multi-DOF joint with high failure tolerance is realized by equipping several redundant actuators in parallel as in Fig.1, right. With the antagonistic muscles the joint stiffness control is easy as well as joint angle control. The muscle itself functions as a speed reducer and a stiffness regulating mechanism. Hence downsizing and weight saving are easily achieved. In addition it is possible to select/increase muscles according to the actual use because

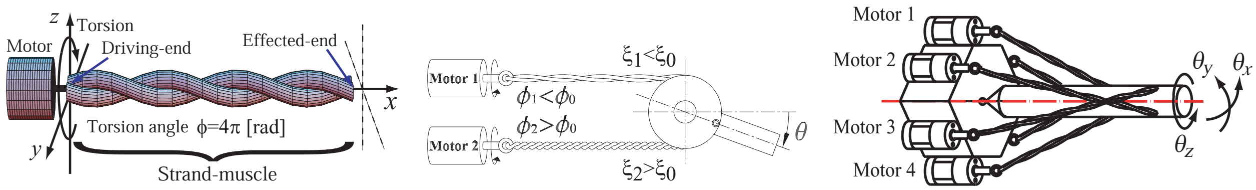

Fig. 1. Strand-Muscle Actuator with two muscle fibers (left), 1-DOF joint of pulley-type (center), and 3-DOF joint (right) actuated by antagonistic Strand-Muscle-Actuators

the muscles are easy to equip/replace, which leads to another good feature: the joint characteristics can be easily improved, *i.e.*, variable/extendable joint mechanism might be possible.

The muscle state $\boldsymbol{x}_{\mathrm{M}}$ of a strand-muscle is defined with ξ, the muscle contraction from the natural muscle length, and T, the muscle tension, as

$$\boldsymbol{x}_{\mathrm{M}}(t) = [\,\xi(t) \quad T(t)\,]^T \in \boldsymbol{R}^2 \tag{1}$$

The muscle state without load nor torsion ($\boldsymbol{x}_{\mathrm{M}} = [0\ 0]^T$) is referred to as the muscle's natural state. The contraction ξ is the sum of the contraction by torsion and the extension by tension. The muscle stiffness S_{M} is defined as $S_{\mathrm{M}} = \partial T/\partial \xi$. It should be noted that larger torsion angle gives smaller muscle stiffness under a fixed tension. Although the muscle stiffness control enhances the action dexterity, it enlarges the redundant degrees of freedom, which leads to the learning difficulty.

Although the StMA can realize flexible and complex motions, its mathematical model is difficult to establish because of various nonlinear factors such as fiber surface friction. The relation between ξ, T and ϕ is referred to as the actuator's basic mechanical characteristics, but practically, the relation between ξ, T and motor drive time t_{M}, $\xi(t_{\mathrm{M}}, T)$, is more useful for control. The basic actuator characteristics $\xi(t, T)$ expresses the contraction ξ of a muscle of natural length L from the twistless state under constant muscle tension T generated by driving the motor for t [sec] with impressed voltage V_c. This formula is experimentally obtained and utilized as the control knowledge for StMA-based joint control. Note that L and V_c are usually fixed and omitted to show for simplicity. The actuator can be controlled based on the experimental control equation with sufficient accuracy[8].

2) Muscle Extension/Contraction - **MEC** *:* The most primitive motions usually fixed depending on the hardware and/or the control scheme are referred to as the *base actions*, or *base motions*. The base motion of Strand-Muscle-Actuator-driven Robots (StMARs) is **MEC**.

The **MEC** that changes a strand-muscle in natural state to the one in a specified state with a specified muscle contraction speed is referred to as a basic **MEC**. Practically, necessary is a general muscle extension/contraction that changes the muscle state from an arbitrary initial state to a requested target state with a specified speed. The control of such a general **MEC** is derived from the control knowledge of basic **MEC**s[16].

A basic **MEC** is specified by the target muscle state and the average muscle contraction speed $\dot{\xi}^{\mathrm{r}}$ from natural state to the target state. The control for **MEC** is executed by simple on/off control that is specified by the duty ratio d and the motor drive time t_{M}. Therefore the action parameter $\boldsymbol{\alpha}_{\mathrm{MEC}}$ and the control parameter $\boldsymbol{\gamma}_{\mathrm{MEC}}$ are defined as

$$\boldsymbol{\alpha}_{\mathrm{MEC}} = [\,\xi^{\mathrm{r}} \quad T^{\mathrm{r}} \quad \dot{\xi}^{\mathrm{r}}\,]^T,\ \boldsymbol{\gamma}_{\mathrm{MEC}} = [\,d \quad t_{\mathrm{M}}\,]^T \tag{2}$$

It is impossible in general to realize an action arbitrarily specified by three action parameters by adjusting two control parameters. $\boldsymbol{\alpha}_{\mathrm{MEC}}$ is restricted by a constraint dependent on the situation where **MEC** is executed, and its degrees-of-freedom is no greater than two in practice. In fact not only motor drive but some physical constraint is necessary to realize independently specified T and ξ. When the antagonistic StMAs drive a joint, some constraint $f(\xi, T) = 0$ will be imposed between T and ξ.

3) Joint Rotation - **JR** *:* The joint state $\boldsymbol{x}_{\mathrm{J}}$ of a StMA-based joint is defined with the joint angle $\boldsymbol{\theta}$ and the joint stiffness $\boldsymbol{S}_{\mathrm{J}}$ as

$$\boldsymbol{x}_{\mathrm{J}}(t) = [\,\boldsymbol{\theta}(t) \quad \boldsymbol{S}_{\mathrm{J}}(t)\,] \in \boldsymbol{R}^{n_{\mathrm{DOF}} \times 2} \tag{3}$$

where n_{DOF} is the DOF of the joint. Although the action parameter of **JR** was the target joint angle in the previous work[5], here we consider more general **JR** whose action/control parameters are, for example, defined as

$$\boldsymbol{a}_{\mathbf{JR}} = [\,\boldsymbol{x}_{\mathrm{J}}^{\mathrm{r}} \quad \boldsymbol{\omega}^{\mathrm{r}}\,] = [\,\boldsymbol{\theta}^{\mathrm{r}} \quad \boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}} \quad \boldsymbol{\omega}^{\mathrm{r}}\,] \in \boldsymbol{R}^{n_{\mathrm{DOF}} \times 3}, \tag{4}$$

$$\boldsymbol{c}_{\mathbf{JR}} = [\,\boldsymbol{\xi} \quad \boldsymbol{T} \quad \dot{\boldsymbol{\xi}}\,] \in \boldsymbol{R}^{M \times 3} \tag{5}$$

where $\boldsymbol{\theta}^{\mathrm{r}}$, $\boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}}$, $\boldsymbol{\omega}^{\mathrm{r}}$ in the action parameter are the target joint angle, the target joint stiffness when $\boldsymbol{\theta}^{\mathrm{r}}$ is realized, the target average joint angular velocity until $\boldsymbol{\theta}^{\mathrm{r}}$ is achieved, respectively. And, M is the number of strand-muscles of the joint, $\boldsymbol{\xi}$, $\boldsymbol{T}$, $\dot{\boldsymbol{\xi}} \in \boldsymbol{R}^M$ in the control parameter are the muscle contraction and tension of each muscle to realize $\boldsymbol{x}_{\mathrm{J}}^{\mathrm{r}}$, the average muscle contraction speed to realize $\boldsymbol{\omega}^{\mathrm{r}}$, respectively. The inheritance of muscle stiffness is important. Because the action parameter of **MEC** includes the muscle tension, the action parameter of **JR** naturally includes joint stiffness. This is further inherited by limb stiffness control in posture-to-posture motions.

Consider here a simple 1-DOF joint ($n_{\mathrm{DOF}} = 1$) with $M = 2$. We investigate the control parameter $\boldsymbol{c}_{\mathbf{JR}} \in \boldsymbol{R}^{2\times 3}$ to change the joint state from $\boldsymbol{x}_{\mathrm{J}}^0$ to $\boldsymbol{x}_{\mathrm{J}}^{\mathrm{r}}$ with $\boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}} = \boldsymbol{S}_{\mathrm{J}}^0$. When $\boldsymbol{\theta}^{\mathrm{r}}$ and $\boldsymbol{S}_{\mathrm{J}}^0$ are given, $\boldsymbol{\xi}$ and $\boldsymbol{T}$ are uniquely determined from the following geometric and static relations

$$\boldsymbol{f}_{\xi}(\boldsymbol{\xi}) = \boldsymbol{0},\ \boldsymbol{f}_{T}(\boldsymbol{T}) = \boldsymbol{0},\ \boldsymbol{f}_{\theta}(\boldsymbol{\xi}) = \boldsymbol{\theta}^{\mathrm{r}},\ \boldsymbol{f}_{S}(\boldsymbol{\theta}^{\mathrm{r}}, \boldsymbol{T}) = \boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}} \tag{6}$$

which are specified by the muscle layout and the mechanical structure such as link size and connection. The duty ratio for each muscle can be determined, for example, so that all the muscles complete the necessary contraction simultaneously. The target muscle contraction speed is then straightforwardly determined, which yields $\dot{\boldsymbol{\xi}}$.

Singular postures should be taken into account depending on the muscle layout, and some optimization for an adequate criterion[11] should be done in order to determine the muscle control when the joint has redundant muscles.

B. Intelligent Composite Action Control

The Intelligent Composite Action Control, ICAC, aims for realizing intelligent robots that can adaptively/evolutionarily realize complex tasks just by giving them the control for fundamental motions. In the ICAC a complex action is formulated as a parameterized combination of several simpler element actions. They have adjustable control parameters, and then the desirable performance is achieved by optimizing the parameters, by repeated action practices, for example. This process is referred to as the optimal action composition. To utilize the empirical knowledge obtained through the optimization, the knowledge is stored in an action intelligence unit. The units for various actions are then hierarchically connected to form the action intelligence network, which plays the central role for action development.

1) Action intelligence: An intelligent robot decides what action to do in the situation and how to do the action after situation awareness. The situation awareness is done by the recognition intelligence, and the action decision is done by the action intelligence. The robot action intelligence is, in a general sense, a mapping from situation to control. Given environmental information specifying the situation the mapping generates the control parameter that specifies the control. The action intelligence is composed of *action planning intelligence* and *action control intelligence*, though it is not always easy to clearly separate. Since it is not easy to construct the action intelligence with full performance for a large-scale task, the intelligence will be constructed in an evolutionary manner, that is, *evolutionary action generation*, *evolutionary applicability extension*, and *evolutionary intelligence refinement*. These are the ICAC's main concrete goals to achieve.

2) Complex actions based on repeated action composition: Suppose that *element actions* $\boldsymbol{E}_{\boldsymbol{\alpha}_k}, k = 1, 2, \cdots, N$, each of which is specified by action parameter $\boldsymbol{\alpha}_k$ and realized by control $\boldsymbol{u}_{\boldsymbol{E}k}(\boldsymbol{\gamma}_k, t)$ specified by control parameter γ_k, are acquired and available. Then consider a complex action $\boldsymbol{C}_{\boldsymbol{a}}$ specified by action parameter $\boldsymbol{a}$ and realized by the combined-form control:

$$\boldsymbol{u}_{\boldsymbol{C}}(\boldsymbol{c}, t) = \sum_{k=1}^{N} \boldsymbol{u}_{\boldsymbol{E}k}(\boldsymbol{\gamma}_k, t) \tag{7}$$

specified by control parameter $\boldsymbol{c}$, where $\sum$ is a composition operator that specifies the way of composition, $i.e.$, sequential, parallel, or hybrid, and other details of composition that contribute to efficient/dexterous actions. The complex action realized by $\boldsymbol{u}_{\boldsymbol{C}}$ is referred to as the *composite action* composed of $\boldsymbol{E}_{\boldsymbol{\alpha}_k}, k = 1, 2, \cdots, N$, and is represented as $\boldsymbol{C}_{a}(\boldsymbol{c})$, or simply as $\boldsymbol{C}_{\boldsymbol{a}}$. The composition is described using the composition operator as

$$\boldsymbol{C}_a(\boldsymbol{c}) = \sum_{k=1}^{N} \boldsymbol{E}_{\boldsymbol{\alpha}_k}(\gamma_k) \tag{8}$$

The optimal composite action realized on a certain complexity level is then regarded as a new element action on a higher level. Hence whether an action is *element* or *composite* is just relative.

For StMARs, every joint rotation, $\mathbf{JR}ij$, that is the base motion for conventional joint type robots, is realized as a composite action composed of several **MEC**s:

$$\mathbf{JR}ij_{\boldsymbol{a}_{\mathbf{JR}}}(\boldsymbol{c}_{\mathbf{JR}}) = \sum_{k=1}^{M_{ij}} \mathbf{MEC}ijk_{\boldsymbol{\alpha}_{\mathrm{MEC}}}(\boldsymbol{\gamma}_{\mathrm{MEC}}) \tag{9}$$

where $\mathbf{MEC}ijk$ represents $\mathbf{MEC}$ for the muscle k of the joint j of the leg/arm i, M_{ij} is the number of muscles contributing to the rotation of the joint j of the leg/arm i.

A posture-to-posture motion, **PTP**, is a parallel composition of **JR**s. A swing-leg action, **SwLeg**, is composed of sequential several **PTP** actions. A stepping-forward action, **Step**, is a parallel composition of **SwLeg**s for each leg, and Walk is composed of sequential several **Step**s. The Approach the ball and the Kick are a Walk and a specialized Swing-Leg, respectively. Both the Pass and the Shoot are composed of the Approach and the Kick. Moreover the cooperative actions by several robots are also realized as their compositions. The Shoot action by a hexapod[5] and Block-centering-shoot by 3 wheeled robots[12] have been realized through the ICAC.

3) Construction of action control intelligence based on action optimization: At each stage of action composition, the desired action is realized by solving the following constrained parameter optimization problem to optimize $\boldsymbol{c}$ so that the action evaluation function $J_{\boldsymbol{C}}$, whose function form is specified by $\boldsymbol{a}$, is minimized under the constraint $\boldsymbol{c} \in \boldsymbol{G}$:

$$\begin{aligned} \mathbf{P}_{\boldsymbol{C}}: \ & \min_{\boldsymbol{c}} J_{\boldsymbol{C}}(\boldsymbol{u}_{\boldsymbol{C}}) \\ & \text{subj. to} \ \ \boldsymbol{c} \in \boldsymbol{G} \end{aligned}$$

Since $J_{\boldsymbol{C}}$ and $\boldsymbol{G}$ are dependent on $\boldsymbol{a}$, the solution should be expressed as $\boldsymbol{c}^*(\boldsymbol{a})$, which means that the control specified by the control parameter $\boldsymbol{c}^*$ is optimal for the composite action $\boldsymbol{C}$ specified by the action parameter $\boldsymbol{a}$. $\boldsymbol{c}^*(\boldsymbol{a})$ obtained for various $\boldsymbol{a}$ will be stored in the optimal control knowledge database. The database is referred to as *knowledge array* and represented as $\boldsymbol{V}_{\boldsymbol{C}}$, which functions as the mapping from $\boldsymbol{a}$ to $\boldsymbol{c}^*$.

When we consider realizing $\boldsymbol{C}_{\boldsymbol{a}}$ we suppose that the optimal control parameter $\boldsymbol{\gamma}_k$ for each $\boldsymbol{E}_{\boldsymbol{\alpha}k}, k = 1, 2, \cdots, N$ is already stored in $\boldsymbol{V}_{\boldsymbol{E}k}$, and it is also known how to generate $\boldsymbol{u}_{\boldsymbol{E}k}$ from $\boldsymbol{\gamma}_k$. That is, in the ICAC, the optimal action composition is regarded as the problem to optimize control parameter $\boldsymbol{c}$ that specifies the action parameters $\boldsymbol{\alpha}_k$

of the component element actions, which is formulated as follows.

$$\begin{aligned}\mathbf{P}_{C}^{\Sigma}: \ & \min_{c\in\mathcal{C}} J_{C}(u_{C}) \\ & \text{subj. to} \quad u_{C}(c,t)=\sum_{k=1}^{N} u_{Ek}(\gamma_k,t) \\ & \qquad \gamma_k = V_{Ek}(\alpha_k(c)), \ k=1,\cdots,N \\ & \qquad \mathcal{C}=\{c \mid \alpha_k(c)\in\mathcal{D}_{\alpha_k}, \forall k\} \\ & \qquad c\in G\end{aligned}$$

where $\mathcal{D}_{\alpha_k}$ represents the domain of α_k where E_{α_k} is executable.

The knowledge arrays for various actions are connected with the same structure as the action composition to form a network, which is referred to as the knowledge array network(Fig.2). If the action parameter a of the target complex action is given at the top of the network, it goes down through the knowledge arrays and composition operators, and is transferred to the base motions, which promptly determines all of their control parameters and then the robot can swiftly execute the target action.

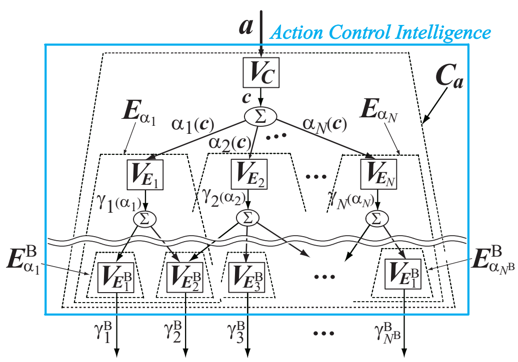

Fig. 2. Knowledge array network as action intelligence

4) Construction of action planning intelligence: For autonomous action planning, the robot must determine what actions to execute, and how to do them, including the number of element actions to combine, without teaching. The action planning intelligence checks the feasibility of several executable actions and proposes the most promising action with most suitable action parameter.

In action optimization with GA the chromosomes get longer as the action gets more complex. And the number of element actions to be combined depends on the situation. Then the variable-chromosome-length GA is introduced, where the crossover operation includes cut-and-connect that changes the chromosome length. With the method versatile cooperative actions according to the situation can be autonomously planned in some examples of 2-vs-1 cooperative soccer action planning problem[12]. The chromosome consists of the number of genes N meaning the number of element actions and the genes $E_k, k=1,2,\cdots,N$, meaning the labels of element actions. When the composition type is simply parallel or sequential, E_k is identified with the element number of the executable action set $\mathcal{E}$:

$$\mathcal{E}=\{E_n, n=1,2,\cdots,N_0\}, \tag{10}$$

and then the chromosome has a simple expression as

$$\tilde{n}=(N,n_1,n_2,\cdots,n_N)\in N^{1+N} \tag{11}$$

$\tilde{n}$ only specifies the sort of element actions and the details of the actions are not specified. Therefore the element actions E_k are concretized and evaluated by randomly choosing several near-optimal actions for m pairs of $(a^j,\tilde{n}^j), j=1,2,\cdots,m$ using the applicable action intelligence V_{Ek} selected based on the estimation of the situation at E_k execution. The composite action composed of m suboptimal $E_k, k=1,2,\cdots,N$ using $\gamma_k^j = V_{Ek}(\alpha_k(c_k(a^j,\tilde{n}^j))), j=1,2,\cdots m$ is evaluated by the corresponding criterion function J_C. The best evaluated combination $\{\gamma_k^{j^*}\}$ among $\{\gamma_k^j\}, j=1,2,\cdots m$ is adopted as the concretization of composite action, and then evaluated by

$$\begin{aligned}&J_C(u_C(a,\tilde{n}))=J_C(u_C(\{\gamma_k^{j^*}\})),\\ &j^*=\arg\min_j J_C(u_C(\{\gamma_k^j\}))\end{aligned}$$

III. Generalized Bernstein Problem

Not only muscle coordination and joint coordination in motor control, but action composition and cooperative work for complex tasks are formulated as a redundancy resolution problem in a unified manner, which is referred to as Generalized Bernstein Problem (GBP) in this article. In this section the GBP is investigated.

A. Bernstein's problem in muscle-driven robot control

The Bernstein's problem does not necessarily have a definite formulation. Which motor units should be used and how frequently should the pulses be transferred to the units to realize a requested muscle contraction? Which joints should be used and how should they rotate to realize desired limb motions? Such problems are introduced[1] as examples. In the problems the objects to be optimized are selected and optimized. The selection and optimization ought to be done simultaneously. Supposing almost all joints will move, in the latter, however, the problem is simplified to the ones where all the given objects are optimized. Besides the selection of ions that are involved with the muscle stimulation is mentioned, and the necessity to deal with it as a complex system is suggested. But here we do not consider dealing with the motion control system for artificial-muscle-based robots as a complex system.

For legged walking in rough terrain the terrain condition with adequate precision should be known. There have been various model acquisition methods proposed[15] such as the realtime 3D terrain map building[6] that constructs a terrain map to determine sutable next footstep with credibility infomation using visual information obtained during walking without any a priori knowledge. Although it is difficult to quickly construct a high precision model in an evironment without a priori infomation, the muscle-based flexible joints with stiffness controlled can adequately adapt to the model uncertainty. Where should be the next footstep, how fast should the leg move with how large joint stiffness? There is a large number of redundant DOF in walk planning, especially in environment with sparse information, it is therefore a typical example of the Bernstein's problem.

1) Redundancy concerning muscle stiffness and contraction speed: In the previous section, described is that it is possible to control the joint stiffness by controlling muscle length and muscle tension, $i.e.$, muscle stiffness, in **MEC**. By inheriting the joint stiffness it is possible to control the stiffness for higher level actions. But it is not necessarily needed to control all the joint stiffness during every action, and in fact they are not always controlled. Restricting the control scheme to the simple on/off control the redundancy in **MEC** control is largely reduced. But without the muscle stiffness specification huge redundancy would remain. When there is no need to specify joint stiffness in **JR** there is no criterion to specify the muscle tensions. But even if there is no criterion, the muscle tension must be adequately specified to control the joint angle.

In the same manner the specification of the muscle contraction speed is not always necessary. Without the specification, however, neither joint rotation speed nor limb motion speed can be controlled.

The most ordinary way to solve the redundant DOF is introduction of some additional criterion. The real-time muscle cooperation is possible by muscle tension averaging or minimization of the maximum muscle tension[11], for example. Considering an additional criterion for every action to solve the redundancy is, however, very difficult, and somewhat preposterous. Should the action parameter of **PTP** include handtip/foottip stiffness? Or the stiffness of the whole leg, posture stiffness, as it were, should be considered? If the answers are yes, every higher level action must include "action stiffness" determined by each joint stiffness in its action parameter. In fact not only action stiffness, but also action speed needs to be considered. Although such stiffness/speed controllability should never be excluded, most desirable is that the stiffness/speed is automatically and appropriately determined by some DOF reduction mechanism without specifying each time. The Bernstein's problem is, as it were, the problem to find such a mechanism.

2) Learning difficulty caused by redundancy: In the ICAC, the action rating indices such as action scale, action learning difficulty, and so on, are defined by use of the action parameter and the control parameter[7]. The action scale is the index that shows how much must be done to accomplish the action, $i.e.$, how many control parameters are specified to execute the whole action. The index is hence measured by the number of control parameters finally used in the base motions. From the definition the action scale becomes larger as action compositions are repeated. With considering muscle stiffness, especially, the action scale of simple walking, for instance, is very large. The action learning difficulty is the index that shows how much decision-making is necessary compared to the given information to specify the action, and is basically measured by the ratio of the number of control parameters to that of action parameters. When all the **MEC**s have to be executed in higher level target actions such as walking without specification of joint stiffness, the learning difficulty is extremely large, that is, the decision-making to execute the action will be quite difficult.

B. Scope and assumptions

On the formulation of the Generalized Bernstein Problem, the scope of consideration is clarified, and some assumptions are given.

1) Intention-action mapping: Although the sensory-motor-coordination, or the construction of a perception to action mapping is essential in robot intelligence, separate consideration of "perception to intention" and "intention to action" is also important. In lower animals' reflective movements the perception-movement mapping might be quite dominant. But as the intelligence develops the intention in the intermediate stage plays more and more important roles. Large amounts of perceptual information are first summarized to form an intention, and then the intention is transformed to an action plan, which might be very complex in order to most likely achieve the intention. Especially in case of human this "summarization" at the intention part is remarkable. In this work the consideration will be focused on the intention-action mapping.

The intention often consists of several sub-intentions, hence they should be expressed in a vector form and the problem should be considered as multi-objective optimization in the near future.

2) Initially acquired actions and gradual action acquisition: In the ICAC the motion control of the level lower than the target action is supposed to be known at least with appropriate accuracy. We suppose this here in the case of the Bernstein's problem on the DOF of animals' motions, too. Without the supposition a human infant who has not acquired how to swing the leg would have to learn walking. It is apparently impractical. A horse can walk just after the birth, because it is born with the element actions necessary for walking acquired to some extent as the inborn ability, even if only suboptimal control parameters for very limited action parameter domain are available.

Living creatures including human being can improve their actions only by slightly modifying what they already know. Although it is possible to generate and program innovative actions based on mutation-like drastic change, the need to do so seems to be questionable. Therefore we do not try to realize complex actions from the beginning, but we assume gradual learning/optimization by modifying the acquired actions.

C. Setting and formulation

1) Problem setting: Suppose that the set of acquired actions $\boldsymbol{\mathcal{E}}$ in a situation $\boldsymbol{s}$ is given as (10). Then consider realizing the composite action $\boldsymbol{C}_a$ that is most suitable in the situation by optimally selecting adequate element actions from $\boldsymbol{\mathcal{E}}$. That is, the GBP is the problem to find: Which element actions $\boldsymbol{E}_n, n = 1, 2, \cdots, N$ should be selected, how they should be composed, and what control parameter $\boldsymbol{c}$ should be given to the element actions to specify the optimal action parameter $\boldsymbol{\alpha}_n$ in the given situation $\boldsymbol{s}$.

The former part "which element actions and how to compose" is the problem to determine an adequate composite action $\boldsymbol{C_a} = \sum_{k=1}^{N} \boldsymbol{E}_{\boldsymbol{\alpha}_{nk}}$ from $\boldsymbol{s}$, *i.e.*, to determine the composition operator Σ. Σ is defined by $\tilde{\boldsymbol{n}}$ in (11) and fusion parameter matrix $\boldsymbol{F}^{\Sigma}$. $\tilde{\boldsymbol{n}}$ specifies the number of element actions N and the element actions, and $\boldsymbol{F}^{\Sigma}$ consists of the parameters that specify the way of composition and other details of composition such as action start timing.

If N_0, the number of acquired element actions, is fixed, $\tilde{\boldsymbol{n}}$ can be expressed as an N_0-bit binary number where the bits corresponding to chosen element actions are 1, or as a single integer obtained by transforming the binary number. But $N_0 \gg N$ and it will gradually increase as actions develop. On the other hand N is no greater than 10 since the action composition at one time is not so complex. The expression given in (11) is therefore used in this work.

2) Formulation and feature: With the assumptions and setting above the Generalized Bernstein Problem (GBP) is formulated as follows.

$$
\begin{aligned}
&\mathbf{GBP}:\\
&\min_{\Sigma} \min_{\boldsymbol{c} \in \boldsymbol{\mathcal{C}}} J_{\boldsymbol{C}}(\boldsymbol{u}_{\boldsymbol{C}})\\
&\text{subj. to} \quad \boldsymbol{u}_{\boldsymbol{C}}(\boldsymbol{c}, t) = \sum_{k=1}^{N} \boldsymbol{u}_{\boldsymbol{E}n_k}(\boldsymbol{\gamma}_{n_k}, t)\\
&\qquad\qquad \boldsymbol{\gamma}_{n_k} = \boldsymbol{V}_{\boldsymbol{E}n_k}(\boldsymbol{\alpha}_{n_k}(\boldsymbol{c})),\ k = 1, 2, \cdots, N\\
&\qquad\qquad \boldsymbol{\mathcal{C}} = \{\boldsymbol{c} \mid \boldsymbol{\alpha}_{n_k}(\boldsymbol{c}) \in \mathcal{D}_{\boldsymbol{\alpha}_{n_k}}, \forall k\}\\
&\qquad\qquad \boldsymbol{c} \in \boldsymbol{G}
\end{aligned}
$$

The major difference between **GBP** and the composite action optimization problem $\mathbf{P}_{\boldsymbol{C}}^{\Sigma}$ in subsection II-B is that Σ is given in $\mathbf{P}_{\boldsymbol{C}}^{\Sigma}$, in other words, there is no need to consider which element actions should be chosen and how should they be composed. On the other side they are all open in **GBP**. **GBP** optimally determines both the composition operator Σ including $\tilde{\boldsymbol{n}}$ that specifies the contents of $\boldsymbol{C_a}$, and the control parameter $\boldsymbol{c}$ that defines the composite action specifically by giving action parameters to the element actions, at the same time.

Because $\tilde{\boldsymbol{n}}$ is an integer-valued vector, **GBP** is a mixed integer programming (MIP). The problem is, however, different from the standard MIP in terms of the fact that the number of real-valued decision variables (action parameter, control parameter, fusion parameter) depends on the selected integer (which element actions to compose and which way of composition to adopt). The problem has, needless to say, infinite number of optimal solutions because of the redundant DOF. In most cases some of them are continuously distributed on several hypersurfaces.

D. Optimal realization of **JR** *for StMA based robots*

Consider an n_{DOF}-DOF joint equipped with sufficient M-muscles ($M \geq n_{\mathrm{DOF}}$). Then the target action is **JR** specified with action parameter $\boldsymbol{a}_{\mathbf{JR}}$ defined in (4). Which N muscles, *i.e.*, which elements of $\boldsymbol{\mathcal{E}}_{\mathrm{MEC}} = \{\mathbf{MEC}n, n = 1, 2, \cdots, M\}$ should be used? How much tension and contraction should be generated by each selected muscle? How fast should each muscle contract? The GBP for a multi-DOF joint rotation to answer such questions is formulated as follows:

$$
\begin{aligned}
&\mathbf{GBP_{JR}}:\\
&\min_{\Sigma_{\mathrm{JR}}} \min_{\boldsymbol{c}_{\mathrm{JR}} \in \boldsymbol{\mathcal{C}}_{\mathbf{JR}}} J_{\mathbf{JR}}(\boldsymbol{u}_{\mathbf{JR}})\\
&\text{subj. to} \quad \boldsymbol{u}_{\mathbf{JR}}(\boldsymbol{c}_{\mathrm{JR}}, t) = \sum_{k=1}^{N} \boldsymbol{u}_{\mathrm{MEC}n_k}(\boldsymbol{\gamma}_{\mathrm{MEC}n_k}, t)\\
&\qquad\qquad \boldsymbol{\gamma}_{\mathrm{MEC}n_k} = \boldsymbol{V}_{\mathrm{MEC}n_k}(\boldsymbol{\alpha}_{\mathrm{MEC}n_k}(\boldsymbol{c}_{\mathrm{JR}})),\\
&\qquad\qquad\quad k = 1, \cdots, N\\
&\qquad\qquad \boldsymbol{\mathcal{C}}_{\mathbf{JR}} = \left\{\boldsymbol{c}_{\mathbf{JR}} \mid \boldsymbol{\alpha}_{\mathrm{MEC}n_k}(\boldsymbol{c}_{\mathrm{JR}}) \in \mathcal{D}_{\boldsymbol{\alpha}_{\mathrm{MEC}n_k}}, \forall k\right\}\\
&\qquad\qquad \boldsymbol{c}_{\mathbf{JR}} \in \boldsymbol{G}_{\mathbf{JR}} = \mathcal{D}_{\boldsymbol{\alpha}_{\mathrm{MEC}n_1}} \times \cdots \times \mathcal{D}_{\boldsymbol{\alpha}_{\mathrm{MEC}n_N}}
\end{aligned}
$$

where $\mathcal{D}_{\boldsymbol{\alpha}_{\mathrm{MEC}n_k}}$ expresses the minimum/maximum value restrictions on $\xi_{n_k}, T_{n_k}, \dot{\xi}_{n_k}$ determined by the StMA characteristics.

The basic **JR** is a parallel composition of **MEC**s for muscles in use, then the fusion parameter is axiomatic and hence the composition operator is simplified as $\Sigma_{\mathrm{JR}} = \tilde{\boldsymbol{n}}_{\mathrm{JR}}$. And when all the muscles are in use, *i.e.*, $N = M$, we have $\tilde{\boldsymbol{n}}_{\mathbf{JR}} = [\,M \quad 1 \quad 2 \quad \cdots \quad M\,]$. For such a simple case the problem reduces to the one without optimization of Σ_{JR}.

Since the action parameter is defined as in (4) the most straightforward criterion function is defined as

$$
\begin{aligned}
J_{\mathbf{JR}}(\boldsymbol{u}_{\mathbf{JR}}) &= \left\| \boldsymbol{a}_{\mathbf{JR}} - \begin{bmatrix} \boldsymbol{x}_{\mathrm{J}}(t) \\ \dot{\boldsymbol{\omega}}(t) \end{bmatrix} \right\|_{\boldsymbol{Q}}\\
&= q_\theta \|\boldsymbol{\theta}^{\mathrm{r}} - \boldsymbol{\theta}(t)\| + q_S \|\boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}} - \boldsymbol{S}_{\mathrm{J}}(t)\| + q_\omega \|\boldsymbol{\omega}^{\mathrm{r}} - \boldsymbol{\omega}(t)\|
\end{aligned}
$$

in order to evaluate the achievement degree of (4) where $\boldsymbol{Q} = \mathrm{Diag}(q_\theta, q_\theta, q_\theta, q_S, q_S, q_S, q_\omega, q_\omega, q_\omega)$. If there is no redundant muscle and all the evaluation weights are positive, *i.e.*, $q_\theta > 0, q_S > 0, q_\omega > 0$, then the problem can uniquely determine the optimal solution, which can be obtained according to the joint control scheme given in subsection II-A. Otherwise the control redundancy is large even without redundant muscles. If $q_S = 0$ which means $\boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}}$ is not given, for instance, then $\boldsymbol{f}_S(\boldsymbol{\theta}^{\mathrm{r}}, \boldsymbol{T}) = \boldsymbol{S}_{\mathrm{J}}^{\mathrm{r}}$ in (6) is not available. Therefore $\boldsymbol{\xi}$ and $\boldsymbol{T}$ cannot be uniquely determined.

E. Toward smart solution

As mentioned in subsection III-A artificial criterions should not be lightly introduced just to uniquely determine the solution. A smart solution is to follow the natural manner used by living creatures. For living creatures "easiest" seems the natural criterion. And here we suppose that "easy" means "less effort" and "as usual", *i.e.*, less decision effort. The muscle stiffness determined according to the *easy* criterion becomes the basic stiffness and inherited to the succeeding composite actions.

While "less effort" is simply translated to "less energy consumption", "as usual" is practiced in this work by making the most of experiences. For a new composite action, the muscle stiffness of most frequently experienced value is adopted as a trial reference value, that is, the initial search point. The target action with such easy determination of stiffness might usually give an unsatisfactory result. The easy decision making and its result, however, will be effectively utilized for action improvement at the next trial chance. When a local search based optimization is done based on action practices from the initial search point that is determined based on the experience, the search point converges to various optimal solutions distributed on the hypersurface depending on the experiences. This will result in the beginning of individuality derived from the experience.

Initially, the redundancy reduction should not be sought, rather admissible solutions obtained by trial and error should be adopted for trial use, and the reduction mechanism should work as needed to improve proficiency by action practices, that is, local search in order to improve the performance and obtain the suboptimal solutions.

IV. Concluding Remarks

The Bernstein's problem to cope with the redundant degrees-of-freedom of musculoskeletal systems in motion control is extended to more general redundancy resolution problems such as the action coordination for a complex task and multi-robot cooperative work. In this article they are unified and formulated as Generalized Bernstein Problem. Through the optimal realization of various robot actions from fundamental motions to complex behaviors, the investigation of Generalized Bernstein Problem will lead to pioneer a study methodology for a wide range of fields from exercise physiology to operations research in a single framework, as it were seeking total intelligence research, although it will be a long story.

Needless to say a human realizes complex actions not only by action composition but also by modifying an acquired action depending upon the purpose and/or the situation, which can be then regarded as a new different action that will be included as a new entry of element actions to compose more complex actions. This process is not a decomposition but a differentiation, which rather is the main route of action development for living creatures. The differentiation is the issue of action labeling. The next step is to construct the systematic solution for the Generalized Bernstein Problem formulated in this article utilizing the differentiation for generating actions of a wider variety, for example, in addition to the action composition based on the ICAC for generating the actions of larger scale and complexity.

References

[1] M.L.Latash, M.T.Turvey: Dexterity And Its Development, Lawrence Erlbaum Associates, Publishers, 1996

[2] G. Taga: Freezing and freeing degrees of freedom in a model neuro-musculo-skeletal system for development of locomotion, Proc. XVIth Int. Soc Biomechanics Congress, p.47, 1997

[3] M. Suzuki: Suboptimal Robot Motion Control for Unexperienced Situations through Intelligent Composite Motion Control, Proc. CD of 1997 European Control Conference, Brussel, 1997

[4] R. Q. van der Linde: Design, analysis, and control of a low power joint for walking robots, by phasic activation of McKibben muscles, IEEE T. Robotics and Automation, Vol. 15, No. 4, pp.599-604, 1999

[5] M. Suzuki, K.-U. Scholl, R. Dillmann: A Method for Learning Complex and Dexterous Behaviors through Knowledge Array Network, Proc. of 2001 IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp.532-538, Maui, 2001

[6] B. Gassmann $et., al.$: Real-time 3d map building for local navigation of a walking robot in unstructured terrain, Proc. IEEE Int. Conf. Intelligent Robots and Systems, Las Vegas, pp.2185-2190, 2003

[7] M. Suzuki, S. Taneshima: Behavior Evaluating Indices for Robot Behavior Learning and Evolution, Proc. IASTED Int. Conf. Robotics and Applications, pp.7-12, Salzburg, 2003

[8] M. Suzuki, A. Ichikawa: Toward springy robot walk using Strand-muscle actuators, Proc. 7th Int. Conf. Climbing&Walking Robots, pp.467-474, Madrid, 2004

[9] S. Arimoto, M. Sekimoto, and R. Ozawa: A challenge to Bernstein's degrees-of-freedom problem in both cases of human and robotic multi-joint movements, IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences, Vol. E88-A, no. 10, pp.2484-2495, 2005

[10] M. Suzuki: Evolutionary acquisition of complex behaviors through Intelligent Composite Motion Control, Proc. 6th IEEE Int. Symp. Computat. Intelligence in Robotics and Automation, Espoo, 2005

[11] M. Suzuki $et.al.$: Redundant muscle coordination of a multi-DOF robot joint by online optimization, Proc. 2007 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics, paper No.106, Zurich, 2007

[12] M. Suzuki: Planning of diverse complex cooperative robot actions using Multi-stage Genetic Algorithm, Proc. 8th IEEE Int. Symp. Computat. Intelligence in Robotics and Automation, pp.8-14, Daejeon, 2009

[13] N. Tomita et.al., Development of a Quadruped Soft Robot with Fully IPMC Body, Proc. SICE Annual Conference 2011, pp.1687/1690, Tokyo, 2011

[14] K. Takashima et.al., Development of Curved Type Pneumatic Artificial Rubber Muscle Using Shape-memory Polymer, Proc. SICE Annual Conference 2011, pp.1691/1695, Tokyo, 2011

[15] K. Walas, D. Belter: Supporting Locomotive Functions of A Six Legged Walking Robot, Int. J. Appl. Math. Comput. Sci., Vol. 21, No. 2, pp.363-377, 2011

[16] M. Suzuki: Base Motions of Muscle-driven Robots for Compositional Action Development, Proc. SICE2012, Akita, 2012