5

Motion Planning by Integration of Multiple Policies for Complex Assembly Tasks

Natsuki Yamanobe[1], Hiromitsu Fujii[2], Tamio Arai[2] and Ryuichi Ueda[2]
[1]*National Institute of Advanced Industrial Science and Technology (AIST), JAPAN*
[2]*The University of Tokyo, JAPAN*

1. Introduction

Robotic assembly has been an active area of manipulation research for several decades. However, almost all assembly tasks, especially complex ones, still need to be performed manually in industrial manufacturing. The difficulty in planning appropriate motion is a major hurdle to robotic assembly.

In assembly tasks, manipulated objects come into contact with the environment. Thus, force control techniques are required for successfully achieving operations by regulating the reaction forces and dealing with uncertainties such as the position errors of robots or objects. Under force control, a robot's responsiveness to the reaction forces is determined by force control parameters. Therefore, planning assembly motions requires designing appropriate force control parameters. Many studies have investigated simple assembly tasks such as peg-in-hole, and some knowledge of appropriate force control parameters for the tasks has been obtained by detailed geometric analysis (Whitney, 1982). However, the types of parameters that would be effective for other assembly tasks are still unknown.

Here, it should be noted that the efficiency is always required in industrial application. Therefore, force control parameters that can achieve successful operations with a short time are highly desirable. However, it is difficult to estimate the cycle time, which is the time taken to complete an operation, analytically. Currently, designers have to tune the control parameters by trial and error according to their experiences and understanding of the target tasks. In addition, for complex assembly, such as insertion of complex-shaped objects, a robot's responsiveness to the reaction forces is needs to be changed according to the task state. Since tuning force control parameters with determining task conditions for switching parameters by trial and error imposes a very heavy burden on designers, complex assembly has been left for human workers.

Several approaches to designing appropriate force control parameters have been presented. They can be classified as follows: (a) analytical approaches, (b) experimental approaches, and (c) learning approaches based on human skill. In the analytical approaches, the necessary and sufficient conditions for force control parameters that will enable successful operations are derived by geometric analysis of the target tasks (e.g., Schimmels, 1997, Huang & Schimmels, 2003). However, the analytical approaches cannot be utilized for obtaining the parameters to achieve operations efficiently since the cycle time cannot be

estimated analytically. Further, it is difficult to derive these necessary or sufficient conditions by geometric analysis for complex shaped objects. In the experimental approaches, optimal control parameters are obtained by learning or by explorations based on the results of iterative trials (e.g., Simons, 1982, Gullapalli et al., 1994). In these approaches, the cycle time is measurable because operations are performed either actually or virtually. Thus, some design methods that consider the cycle time have been proposed (Hirai et al., 1996, Wei & Newman, 2002). However, Hirai et al. only dealt with simple planar parts mating operations, and the method presented by Wei and Newman was applicable only to a special parallel robot. In addition, these approaches cannot be applied to complex assembly since it is too time-consuming to explore both parameter values and task conditions for switching parameters. In the last approaches based on human skill, the relationship between the reaction forces and the appropriate motions are obtained from the results of human demonstration (e.g., Skubic & Volz, 2000, Suzuki et al., 2006). Although some studies on these approaches have addressed the complex assembly that needs some switching of parameters, they cannot always guarantee the accomplishment of tasks because of the differences in body structure between human demonstrators and robots. Above all, relying on human skill is not always the best solution to increasing the task efficiency. Therefore, there is no method for planning assembly motions that can consider the task efficiency and have the applicability to complex assembly.

From another point of view, a complex assembly motion consists of some basic assembly motions like insertion or parts matting motions. Basic assembly motions can be accomplished with fixed force control parameters; therefore, it is relatively simple to program them. In addition, there are many types of control policies and task knowledge that are applicable to planning complex assembly motions: programs previously coded for similar tasks; human demonstration data; and the expertise of designers regarding the task, the robot, and the work environment.

Therefore, we adopt a step by step approach in order to plan complex assembly motions required in industrial applications. First, a method for basic assembly motion has been presented in order to design appropriate force control parameters that can efficiently achieve operations (Yamanobe et al, 2004). Then, based on the results, a policy integration method has been proposed in order to generate complex assembly motions by utilizing multiple policies such as basic assembly motions (Yamanobe et al, 2008). In this paper, we present these methods and show the simulation results in order to demonstrate the effectiveness of them.

This paper will proceed in the following way: Section 2 explains the problem tackled in this paper. In Section 3, a parameter designing method for basic assembly motion is firstly shown. In Section 4, a method for planning robot motions by utilizing multiple policies is then presented. In Section 5, the proposed methods are applied to clutch assembly. Basic assembly motions that constitute the clutch assembly motion are first obtained based on the method explained in Section 3, and the simulation results of integrating them are shown. Finally, Section 6 concludes this paper.

2. Problem Definition

In assembly tasks, the next action is determined on the basis of observable information, such as the current position of the robot, the reaction forces, and the robot's responsiveness; and

information of the manipulated objects obtained in advance. Therefore, we assume that assembly tasks can be approximated by Markov decision processes (MDPs) (Sutton & Barto, 1998).

The problem considered in this paper is then formalized as follows:

- States $S = \{s_i | i = 1, \cdots, N_s\}$: A robot belongs to a state s in the discrete state space, S . A set of goal states, $S_{\text{goal}} \subset S$, is settled.
- Actions $\mathcal{A} = \{a_j | j = 1, \cdots, N_a\}$: The robot achieves the task by choosing an action, a, from a set of actions, $\mathcal{A}$, at every time step. A control policy for assembly tasks is defined as a sequence of force control parameters. Thus, the actions are represented as a set of force control parameters. While only one action is applied for basic assembly: $N_a = 1$, several actions need to be provided and swiched according to the states for achieving complex assembly: $N_a > 1$.
- State transition probabilities $\mathcal{P}^a_{ss'}$: State transition probability depends only on the previous state and the action taken. $\mathcal{P}^a_{ss'}$ denotes the probability that the robot reaches s' after it moves with a from s .
- Rewards $\mathcal{R}^a_{ss'} \in R$: $\mathcal{R}^a_{ss'}$ denotes the expected value of the immediate evaluation given to the state transition from s to s' by taking a . The robot aims to maximize the sum of rewards until it reaches a goal state. An appropriate motion is defined as the motion that can achieve a task efficiently. Hence, a negative value, namely, a penalty that is proportional to the time required for a taken action, is given as the immediate reward at each time step.

In addition, this paper presumes that the robot is under damping control, which is described as follows:

$$\boldsymbol{v}_{\text{ref}} = \boldsymbol{v}_0 + \boldsymbol{A}\boldsymbol{f}_{\text{out}}, \tag{1}$$

where $\boldsymbol{v}_{\text{ref}} \in R^6$ is the reference velocity applied to the robot; $\boldsymbol{v}_0 \in R^6$ is the nominal velocity; $\boldsymbol{A} \in R^{6\times 6}$ is the admittance matrix; and $\boldsymbol{f}_{\text{out}} \in R^6$ is the reaction force acting on the object from the environment. Both the nominal velocity $\boldsymbol{v}_0$ and the admittance matrix $\boldsymbol{A}$ are damping control parameters and determine robot motions. The admittance matrix determines how the reference velocity should be modified according to the reaction force, and the nominal velocity describes the motion of the robot in free space.

3. Method of Designing Force Control Parameters for Basic Assembly

In order to obtain effective policies for basic assembly motions, a method of designing force control parameters that can reduce the cycle time has been proposed (Yamanobe et al., 2004). An experimental approach is adopted so as to evaluate the cycle time; and the parameter design method through iterative operations is formulated as a nonlinear constrained optimization problem as follows:

$$\begin{aligned} &\text{minimize}: \quad V(\boldsymbol{p}) \\ &\text{subject to}: \quad \boldsymbol{p} \in C, \end{aligned} \tag{2}$$

where $V(\boldsymbol{p})$ is the objective function that is equal to the cycle time; $\boldsymbol{p}$ is a vector that consists of optimized parameters; and C is a set of optimized parameters that satisfy certain constraints, which are conditions that must be fulfilled to ensure successful motions. Here, the optimized parameters are damping control parameters, such as the admittance matrix $\boldsymbol{A}$ and the nominal velocity $\boldsymbol{v}_0$.

A difficulty in this optimization problem is that it is impossible to calculate the derivatives of the objective function with respect to the optimized parameters since the cycle time is obtained only through trials. Therefore, we used a direct search technique: a combination of the downhill simplex method and simulated annealing (Press et al., 1992).

This method can deal with various assembly motions accomplished with fixed force control parameters. In addition, specific conditions desired for a particular operation can be easily considered by adding to the constraints of the optimization. Some effective policies for basic assembly motions, such as insertion motion and search motion, were obtained based on this method; the detailed results are shown in Section 5.

4. Motion Planning by Integration of Multiple Policies

In order to plan complex assembly motions, we have proposed a method for integrating several basic assembly motions and task knowledge that are effective for task achievement (Yamanobe et al. 2006) (Fig. 1). In our method, we represent a control policy for robots with a state action map, which denotes a look-up table connecting a state of a robot and its surroundings to its actions. Owing to the simplicity of the map, we can handle various policies and knowledge that exists in different forms using only one format, i.e., a state action map. The effective policies are selected and represented in a map by designers, and a new policy for the target task is efficiently constructed based on them. Here, it is difficult to determine the conditions for effectively applying the policies to the task. In some states, the applied policies would conflict with others and fail to achieve the task. Our method develops a robot motion by modifying the applied policies for the states in which they result in a failure.

4.1 Related works

On existing policies exploitation, several studies have been conducted especially in reinforcement learning in order to quickly learn motions for new tasks. Thrun and Mitchell proposed lifelong learning (Thrun & Mitchell, 1995). In this approach, the invariant policy of individual tasks and environments is learned in advance and employed as a bias so as to accelerate the learning of motions for new tasks. Tanaka and Yamamura presented a similar idea and applied it to a simple navigation task on a grid world (Tanaka & Yamamura, 2003). The past learning experiences are stored as the mean and the deviation of the value functions obtained for each task which indicates the goodness of a state or an action. Minato and Asada showed a method for transforming a policy learned in the previous tasks into a new one by modifying it partially (Minato & Asada, 1998). Although these approaches can acquire a policy that is common to a class of tasks and improve their learning performance by applying it to a new task in the class, only one type of policy is utilized in these methods.

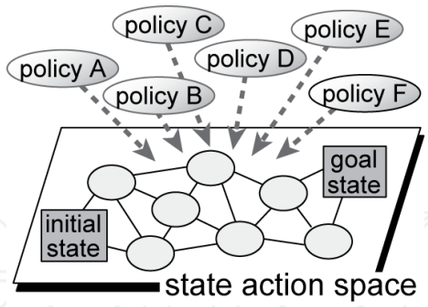


Fig. 1. Robot motion obtained by the integration of multiple policies

In the case of multiple-policy applications, Lin proposed a learning method to use various human demonstration data as informative training examples for complex navigation tasks (Lin, 1991). However, this method cannot deal with false teaching data. Sutton et al. defined a sequence of actions that is effective for task as an option; they then presented an approach to increase the learning speed by using options interchangeably with primitive actions in the reinforcement learning framework (Sutton et al., 1999). This approach can modify the unsuitable parts of options in the learning process and, therefore, integrate multiple options. This approach is similar to our methodology. However, the usable policy is limited to a sequence of actions. The advantage of our method is to be easily able to deal with various types of existing policies and knowledge.

4.2 Method for integrating multiple policies

As described above, the basic idea of our method is as follows: first, all applied policies are written in a state action map; after that, a new policy for the target task is constructed by partially modifying the applied policies.

Applied policies, such as policies for basic assembly motions, are selected by designers and represented in a state action map. The states in which each policy is represented are also determined by designers. Knowledge for the target task defines the state space and rewards, and sets a priority among the applied policies. When multiple policies are represented on a map, the map includes

states in which no policy is applied,
states in which multiple policies are written, and
states in which the actions following the applied policies fail to achieve the task.

We define the last-named states as “failing states.” In order to obtain a new policy that is feasible for the target task, the following processes are required.

- Policy definition according to the applied policies.
- Selection of failing states.
- Policy modification for the failing states.

Let us explain each procedure in the following sub-sections.

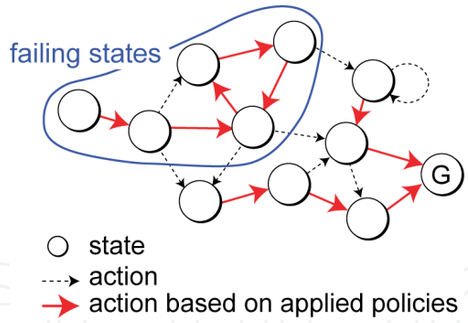


Fig. 2. Failing states

4.2.1 Policy Exploration Based on Applied Policies

$s_{\text{policy}} \in S$ represents the state in which policies are introduced. A set of actions available at s_{policy}, $\mathcal{A}_{\text{p}}(s_{\text{policy}}) \subset \mathcal{A}$, is defined as follows:

$$\mathcal{A}_{\text{p}}(s_{\text{policy}}) = \{a_{\text{p}} \in \mathcal{A}_{\text{p}}^{k}(s_{\text{policy}}) \mid k = 1, \cdots, N_{\text{p}}(s_{\text{policy}})\}, \tag{3}$$

where $\mathcal{A}_{\text{p}}^{k}(s_{\text{policy}})$ is a set of actions based on policy k at s_{policy} and $N_{\text{p}}(s_{\text{policy}})$ is the number of policies applied to s_{policy}. At the state in which no policy is applied, $s_{\overline{\text{policy}}}$, the robot can take all actions involved in $\mathcal{A}$. The new policy for the target task is efficiently decided on the basis of these actions limited by the applied policies. An optimal control policy can maximize the state value, $V(s)$, that is defined as the expected sum of the rewards from a state s to a goal state. The new policy is explored while estimating the state value function, V, based on dynamic programming (DP) (Bellman, 1957) or reinforcement learning (Sutton & Barto, 1998).

4.2.2 Failing States Selection

If actions are limited by the applied policies, the robot might fail to perform the task at some states. The failing states, S_{fail}, are defined as the states from which the robot cannot reach a goal state only with the actions implemented based on the applied policies. Fig. 2 shows an example of failing states.

In failing states, state transitions are infinitely repeated. Since a penalty is given for each action, the state value at a failing state, $V(s_{\text{fail}})$, decreases. Hence, we select the failing states by using the decrease in the state values. First, a state $\tilde{s}_{\text{fail}}$ with a value $V(\tilde{s}_{\text{fail}})$ that is lower than $V_{\min}$ is found. $V_{\min}$ is the threshold value. Then, S_{fail} is defined as a set of $\tilde{s}_{\text{fail}}$ and the states that the robot can reach from $\tilde{s}_{\text{fail}}$ according to the actions limited by the applied policies.

4.2.3 Policy Modification

In order to correct the infinite state transitions in the range of the failing states, the applied policies need to be modified partially. In particular, the actions that are available in the failing states are changed from the actions limited by the policies, $\mathcal{A}_{\text{p}}(s_{\text{policy}})$, into the normal actions, $\mathcal{A}$, that are available for the robot. Then, the new policy is explored again

only for the failing states. By repeating these processes until no failing state is selected, we can efficiently obtain a new policy that is not optimal but feasible for the whole target task.

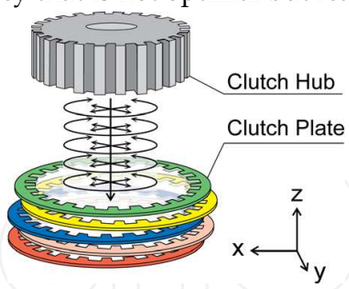


Fig. 3. Clutch assembly

5. Application of Policy Integration Method to Complex Assembly

The proposed method for the integration of multiple policies is applied to clutch assembly (Fig. 3) in order to demonstrate its validity in complex assembly.

Clutch assembly is a complicated assembly task, in which a splined clutch hub is inserted through a series of movable toothed clutch plates. Since the clutch plates can move in the horizontal plane and rotate about the vertical, the plates are nonconcentric and have random phase angles before the clutch hub is inserted. In order to efficiently execute the task, a search motion for matching the centerline and the phase angle of the hub to those of each plate is required in addition to a simple insertion motion. However, the task is achieved by only search motion in practical applications because it is difficult to perceive that the teeth on the hub become engaged with the proper grooves on the plate.

In this section, the appropriate motion for clutch assembly is developed by integrating the policies for insertion motion and search motion.

5.1 Simulator for clutch assembly

We utilize a simulator for integrating multiple policies as well as the optimization for basic assembly motions in order to avoid problems such as the occurrence of a crash when an operation results in a failure during policy exploration and the deterioration of objects and/or a robot on account of iterative operations. Although a modelling error might be a problem in a simulation, this problem can be overcome by developing a simulator based on preliminary experiments.

In this subsection, the simulator used in this paper is explained. The simulator consists of a physical model and a control system model. The physical model has been developed using LMS DADS that is mechanical analysis software. This model expresses the work environment in which operations are performed and is composed of the manipulated object and the assembled objects. For the simulator of clutch assembly, the physical model consists of a clutch hub as the manipulated object, clutch plates as the assembled objects, and the housing that holds the clutch plates.

The control system model has been developed using MATLAB Simulink. In this model, the mechanical compliance and the control system of the robot are expressed. A schematic view of the simulator is shown in Fig. 4. The position of the manipulated object, $\boldsymbol{x}_{\text{object}} \in R^6$, and the reaction force acting on the object, $\boldsymbol{f}_{\text{out}}$, constitute the output from the physical model

and are fed into the control system model. The reference velocity of the robot $\boldsymbol{v}_{\mathrm{ref}}$ is calculated from the damping control law (eq. 1) by the damping controller. The position controller of the robot is modeled as a second-order system. The robot is modeled as a rigid body, and its mechanical compliance is described as a spring and a damper between the end-effector of the robot and the manipulated object. Based on the position controller and the robot's mechanical compliance, the position of the robot, $\boldsymbol{x}_{\mathrm{robot}} \in R^6$, is written as follows:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{s}}[\ddot{\boldsymbol{x}}_{\mathrm{robot}} + 2\zeta\omega_n(\dot{\boldsymbol{x}}_{\mathrm{robot}} - \boldsymbol{v}_{\mathrm{ref}}) + \omega_n{}^2(\boldsymbol{x}_{\mathrm{robot}} - \boldsymbol{v}_{\mathrm{ref}}\Delta t)] &= \boldsymbol{D}_{\mathrm{e}}(\dot{\boldsymbol{x}}_{\mathrm{object}} - \dot{\boldsymbol{x}}_{\mathrm{robot}}) + \boldsymbol{K}_{\mathrm{e}}(\boldsymbol{x}_{\mathrm{object}} - \boldsymbol{x}_{\mathrm{robot}}) \\ &= -\boldsymbol{f}_{\mathrm{in}}, \end{aligned} \tag{4}$$

where $\boldsymbol{M}_{\mathrm{s}} \in R^{6\times6}$ is the inertia matrix; ζ and ω_n are the damping coefficient and the natural frequency of the second-order system, respectively; Δt is the sampling time in the controller; $\boldsymbol{D}_{\mathrm{e}} \in R^{6\times6}$ and $\boldsymbol{K}_{\mathrm{e}} \in R^{6\times6}$ are the damping matrix and the stiffness matrix between the robot and the manipulated object, respectively. $\boldsymbol{f}_{\mathrm{in}}$ is the force acting on the manipulated object from the robot through the spring and the damper and fed into the physical model for actuating the object.

In oder to obtain data for the simulator, preliminary experiments: measurement of the stiffness of the robot, $\boldsymbol{K}_{\mathrm{e}}$, and clutch assembly, were performed using a 6 DOF (degree of freedom) manipulator, FANUC M-16i. A clutch consisting of five clutch plates was used in the experiments of clutch assembly. Each clutch plate has 45 teeth and is 0.8 [mm] thick; the distance between adjacent plates is 3.75 [mm]. The plates are contained within a fixed subassembly, and they can move independently in the horizontal plane ±1 [mm] and rotate about the vertical. The clutch hub is 95 [mm] in diameter and 35 [mm] in height. The height of each of the teeth is 5 [mm]. It is possible to represent the actual tasks by adjusting the parameters in the simulator. Thus, the parameters of the control system model and the coefficient of kinetic friction in the physical model were determined by trial and error in order to obtain simulation results that are close to the experimental results.

5.2 Acquisition of policies for basic assembly motions

Using the method for optimizing force control parameter, which is presented in Section 3, appropriate policies for insertion motion and search motion are obtained.

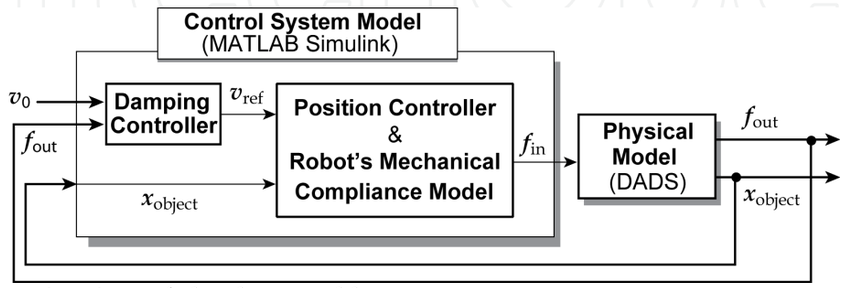


Fig. 4. Schematic view of clutch assembly

5.2.1 Policy acquisition for insertion motion

A policy for insertion motion, i.e. appropriate force control parameters, is obtained on the basis of cylindrical peg-in-hole tasks. A simulator for peg-in-hole tasks was first developed only by changing the physical model in the simulator for clutch assembly. Then, the optimization of force control parameters was performed by considering the following constraints.

Stability conditions

We consider the stability of the control system in the case where the manipulated object is in contact with the assembled object. When the manipulated object is constrained, eq. 4 can be discretely expressed as follows:

$$\boldsymbol{M}_{\mathrm{s}}\left[\left(\frac{\boldsymbol{x}_{\mathrm{robot}(i+1)}-2\boldsymbol{x}_{\mathrm{robot}(i)}+\boldsymbol{x}_{\mathrm{robot}(i-1)}}{\Delta t^{2}}\right)+2\zeta\omega_{n}\left(\frac{\boldsymbol{x}_{\mathrm{robot}(i)}-\boldsymbol{x}_{\mathrm{robot}(i-1)}}{\Delta t}-\boldsymbol{v}_{\mathrm{ref}(i)}\right)+\omega_{n}{}^{2}\left(\boldsymbol{x}_{\mathrm{robot}(i)}-\boldsymbol{x}_{\mathrm{ref}(i)}\right)\right]$$
$$=\boldsymbol{K}_{\mathrm{e}}\left(\boldsymbol{x}_{\mathrm{object}}-\boldsymbol{x}_{\mathrm{robot}(i)}\right)-\boldsymbol{D}_{\mathrm{e}}\left(\frac{\boldsymbol{x}_{\mathrm{robot}(i)}-\boldsymbol{x}_{\mathrm{robot}(i-1)}}{\Delta t}\right), \tag{5}$$

where, $\boldsymbol{x}_{\mathrm{object}}$ is constant; $\dot{\boldsymbol{x}}_{\mathrm{object}}=0$; and $\boldsymbol{x}_{\mathrm{robot}(i)}$ is the position of the robot at $t=i\Delta t$. Using eq. 5 and considering the delay of the reaction force information from the force sensor, we can discretely express the damping control law (eq. 1) as follows:

$$\begin{aligned}\boldsymbol{v}_{\mathrm{ref}(i)}&=\frac{\boldsymbol{x}_{\mathrm{object}}-\boldsymbol{x}_{\mathrm{robot}(i)}}{\Delta t}\\&=\boldsymbol{v}_{0}+\boldsymbol{A}\left[\boldsymbol{K}_{\mathrm{e}}\left(\boldsymbol{x}_{\mathrm{object}}-\boldsymbol{x}_{\mathrm{robot}(i-2)}\right)-\boldsymbol{D}_{\mathrm{e}}\left(\frac{\boldsymbol{x}_{\mathrm{robot}(i-2)}-\boldsymbol{x}_{\mathrm{robot}(i-3)}}{\Delta t}\right)\right],\end{aligned} \tag{6}$$

Let $\boldsymbol{X}_{\mathrm{robot}(i)}=\left(\boldsymbol{x}_{\mathrm{robot}(i)},\boldsymbol{x}_{\mathrm{robot}(i-1)},\boldsymbol{x}_{\mathrm{robot}(i-2)},\boldsymbol{x}_{\mathrm{robot}(i-3)}\right)^{T}\in R^{24}$; eq. 5 and eq. 6 can then be rewritten as follows:

$$\begin{aligned}\boldsymbol{X}_{\mathrm{robot}(i+1)}&=\begin{bmatrix}\boldsymbol{W}_{11}&\boldsymbol{W}_{12}&\boldsymbol{W}_{13}&\boldsymbol{W}_{14}\\\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{I}&\boldsymbol{O}\end{bmatrix}\boldsymbol{X}_{\mathrm{robot}(i)}+\boldsymbol{C}\\&:=\boldsymbol{W}\boldsymbol{X}_{\mathrm{robot}(i)}+\boldsymbol{C},\end{aligned} \tag{7}$$

where

$$\begin{aligned}\boldsymbol{W}_{11}&=\boldsymbol{M}_{\mathrm{s}}^{-1}\left[2\boldsymbol{M}_{\mathrm{s}}-\Delta t\left(2\zeta\omega_{n}\boldsymbol{M}_{\mathrm{s}}+\boldsymbol{D}_{\mathrm{e}}+\Delta t\boldsymbol{K}_{\mathrm{e}}\right)\right],\\\boldsymbol{W}_{12}&=\boldsymbol{M}_{\mathrm{s}}^{-1}\left[\Delta t\left(2\zeta\omega_{n}\boldsymbol{M}_{\mathrm{s}}+\boldsymbol{D}_{\mathrm{e}}\right)-\boldsymbol{M}_{\mathrm{s}}\right],\\\boldsymbol{W}_{13}&=-\Delta t\omega_{n}\boldsymbol{A}\left(\Delta t\omega_{n}+2\zeta\right)\left(\Delta t\boldsymbol{K}_{\mathrm{e}}+\boldsymbol{D}_{\mathrm{e}}\right),\\\boldsymbol{W}_{14}&=-\Delta t\omega_{n}\boldsymbol{A}\boldsymbol{D}_{\mathrm{e}}\left(\Delta t\omega_{n}+2\zeta\right),\\\boldsymbol{C}&=\left[\Delta t^{2}\boldsymbol{M}_{\mathrm{s}}^{-1}\left[\omega_{n}\boldsymbol{M}_{\mathrm{s}}\left(\Delta t\omega_{n}+2\zeta\right)\left(\boldsymbol{v}_{0}+\boldsymbol{A}\boldsymbol{K}_{\mathrm{e}}\boldsymbol{x}_{\mathrm{object}}\right)+\boldsymbol{K}_{\mathrm{e}}\boldsymbol{x}_{\mathrm{object}}\right],\boldsymbol{O},\boldsymbol{O},\boldsymbol{O}\right]^{T},\end{aligned}$$

and $\boldsymbol{I}\in R^{6\times 6}$ is the identity matrix.

The series $\boldsymbol{X}_{\mathrm{robot}(i)}$ must converge to a certain value in order to ensure the stability of the control system. Therefore, the stability condition can be theoretically described as $|\lambda_{j}|<1$, where λ_{j} is each of the eigenvalues of $\boldsymbol{W}$. Here, the state of the control system gets close to

the instability as the maximum value of $|\lambda_j|$, $\lambda_{\max}$, becomes large. Then, $\lambda_{\max}$ can be used as a value that evaluates the instability of the system. Considering the modeling error of the simulator, we define the stability condition in the optimization as:

$$\lambda_{\max} < 0.99 \tag{8}$$

Condition for nominal velocity

To achieve insertion, the z-element of the nominal velocity $\boldsymbol{v}_{0z}$ must be negative.

Limitation of the reaction force

The rating of the force sensor bounds the allowable reaction force. We define the limit of the reaction force as the rating: 294 [N] and 29.4 [Nm].

If the given parameters cannot satisfy these above constraints, the simulation is stopped. The operation is regarded as a failure, and a very large value is assigned to the objective function. Namely, we use a penalty function to consider these constraints.

Here it should be noted that, if the optimization of the control parameters were performed simply, the obtained parameters would be too specialized in a specific initial condition. Thus, simulations are performed with possible errors in the initial position of the peg in order to deal with the various errors. Let the maximal value of the possible position error of the peg be 1 [mm] and that of rotation error be 1 [deg]. Six kinds of position errors are considered here: along the x-axis, along the y-axis (positive, negative), rotation error around the x-axis (positive, negative), and rotation error around the y-axis. Simulation of the peg-in-hole is performed with the above errors in the initial position of the peg; then, the mean value of the six kinds of cycle time obtained from the simulation with each error is defined as the objective function.

The admittance matrix $\boldsymbol{A}$ was defined as a diagonal matrix; the robot was position-controlled only around the insertion axis, i.e. z-axis, and the x-axis and the y-axis were treated equally because of cylindrical peg-in-hole tasks. The optimization was performed for the optimized control parameters, $\boldsymbol{p} = (a_{x-y}, a_z, a_{rx-ry}, v_{0z})^T$, and the damping control parameters are thus expressed as follows:

$$\begin{aligned} \boldsymbol{A} &= \mathrm{diag}(a_{x-y}, a_{x-y}, a_z, a_{rx-ry}, a_{rx-ry}, 0), \\ \boldsymbol{v}_0 &= (0, 0, v_{0z}, 0, 0, 0)^T. \end{aligned} \tag{9}$$

In Fig. 5, the results for the optimization are presented. The horizontal axis in Fig. 5 represents the number of the simplex deformation in the optimization, which denotes the progress of the parameter exploration. The values of each element of the vector $\boldsymbol{p}$ at the best and worst points of the simplex are plotted. Similarly, the objective values at the best and worst points of the simplex are also plotted.

As shown in the bottom-left figure in Fig. 5, the objective value decreased as the optimization proceeded. Around the deformation count 80, the objective values at the worst point of the simplex were huge. That is because the parameters at the points broke the condition for the reaction force. The value of a_{rx-ry} shown in the middle-left figure in Fig. 5 became large for dealing with the orientation errors quickly. As shown in the top-right and the middle-right figures in Fig. 5, the magnitude of a_z and v_{0z} changed interacting each other. The peg is inserted more quickly, as v_{0z} increases. However, the larger the value of the initial velocity is, the larger the magnitude of the reaction force becomes. Thus, the value of a_z changed in order to keep the reaction force from violating its constraint. As shown in

these results, we obtained the appropriate force control parameters that can achieve insertion motions with short cycle time and handle various possible errors.

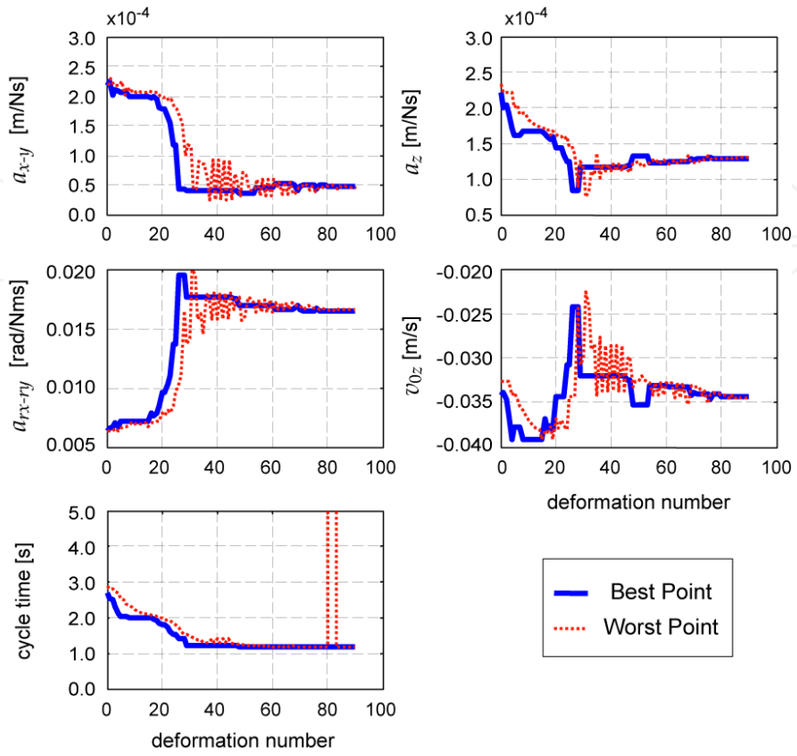


Fig. 5. Results of optimization for insertion motion

The simulation whose results are shown in Fig. 5 took about 189[h] using a Windows PC with Pentium 4 CPU running at 2.8[GHz].

5.2.2 Policy acquisition for search motion

A policy for search motion is acquired on the basis of the clutch assembly performed in the preliminary experiments.

In the search motion, a cyclic motion in the horizontal plane is performed while pressing the assembled object in order to engage the manipulated object with it. Each clutch plate can move in the x-y plane and rotate about the z-axis in the clutch assembly. The cyclic motion along the x-axis and the y-axis as well as around the z-axis was adopted and was achieved by reversing the nominal velocity $\boldsymbol{v}_0$ when the hub goes beyond the search area $\mathbf{R} = (\pm R_x, \pm R_y, \pm R_{rz})^T = (\pm 1\,[\mathrm{mm}], \pm 1\,[\mathrm{mm}], \pm 4\,[\mathrm{deg}])^T$. The elements of the nominal velocity, which are related to the cyclic motion, are v_{0x}, v_{0y}, and v_{0rz}; they were determined as follows:

$$(v_{0x}, v_{0y}, v_{0rz})^T = k_c \boldsymbol{v}_{bc}, \tag{10}$$

where k_c is the coefficient of the cyclic motion velocity, and $\boldsymbol{v}_{bc} \in R^3$ is the base velocity of the cyclic motion defined so as to cover the entire search area $\boldsymbol{R}$. In order to achieve this

motion when the clutch hub is constrained by the clutch plates, the target force $\boldsymbol{f}_{\mathrm{t}} \in R^6$, which is the force applied by the manipulated object on the environment in the steady state, should be defined appropriately. Here, the target force $\boldsymbol{f}_{\mathrm{t}}$ is expressed from eq. 1 as follows:

$$\boldsymbol{v}_0 + \boldsymbol{A}\boldsymbol{f}_{\mathrm{t}} = 0 \,. \tag{11}$$

Let $\boldsymbol{v}_{\mathrm{bc}} = (2.25 \times 10^{-3}\ [\mathrm{m/s}], 7.54 \times 10^{-4}\ [\mathrm{m/s}], 0.92\ [\mathrm{rad/s}])^T$ and $\boldsymbol{f}_{\mathrm{tc}} = (f_{\mathrm{t}x}, f_{\mathrm{t}y}, f_{\mathrm{t}rz})^T = (49\,[\mathrm{N}], 49\,[\mathrm{N}], 7.8\,[\mathrm{Nm}])^T$, which is the elements of the target force related to the cyclic motion, based on the experiences. The admittance matrix $\boldsymbol{A}$ was defined as a diagonal matrix. The manipulator was position-controlled for the directions that were not relevant to the cyclic motion and the pressing: around the x-axis and the y-axis. Therefore, the vector of the optimized control parameters $\boldsymbol{p}$ was defined as $\boldsymbol{p} = (k_{\mathrm{c}}, a_z, v_{0z})^T$, and the damping control parameters are then expressed as follows:

$$\begin{aligned} \boldsymbol{A} &= \mathrm{diag}(a_x, a_y, a_z, 0, 0, a_{rz}), \\ \boldsymbol{v}_0 &= (v_{0x}, v_{0y}, v_{0z}, 0, 0, v_{0rz})^T, \end{aligned} \tag{12}$$

where v_{0x}, v_{0y}, v_{0rz} is calculated from eq. 10 with k_{c} and $\boldsymbol{v}_{\mathrm{bc}}$, and a_x, a_y, a_{rz} is determined using eq. 11 with $\boldsymbol{f}_{\mathrm{tc}}$.

The optimization for search motion was executed with the same constraints considered in that for the insertion motion: stability condition, condition for nominal velocity, and limitation of the reaction force. In addition, in order to deal with various arrangements of the clutch plates, we divided the clutch assembly into two phases: insertion to the first plate from free space and insertion to the other clutch plates. Simulations were only performed for the insertion through the first and the fifth plate with the possible errors of the clutch plate, which are presented in Table 1. The mean value of the cycle time obtained from the simulation through each plate and with each error was defined as the objective function as well as the optimization for the insertion motion.

The results of this optimization are presented in Fig. 6. The horizontal axis represents the number of the simplex deformation in the optimization. The vertical axis represents the values of each element of the vector $\boldsymbol{p}$ and the objective values at the best and worst points of the simplex.

Position error (along x-axis)	Phase angle error
+0.4 [mm]	+1 [deg]
+0.4 [mm]	-1 [deg]
-0.4 [mm]	+1 [deg]
-0.4 [mm]	-1 [deg]

Table 1. Position/phase angle error of the clutch plate in the parameter optimization

As shown in the bottom-right figure in Fig. 6, the objective value decreased as the optimization proceeded. It shows that the parameters that can achieve the task with short cycle time were obtained. The value of k_{c} plotted in the top-left figure grew large as the optimization proceeded, and the value of a_z and the absolute value of v_{0z} got smaller as

interacting each other. The decrease in a_z causes the stiff force control along the insertion axis. Due to the increase in k_c, which led to the increase of the cyclic motion velocity, the stiff force control was desired in order to insert the clutch hub effectively when the teeth on the hub engaged with the grooves on the plates. In addition, since the stiff force control tends to cause large reaction force, v_{0z} changed interactively in order to keep the reaction force from violating its constraints.

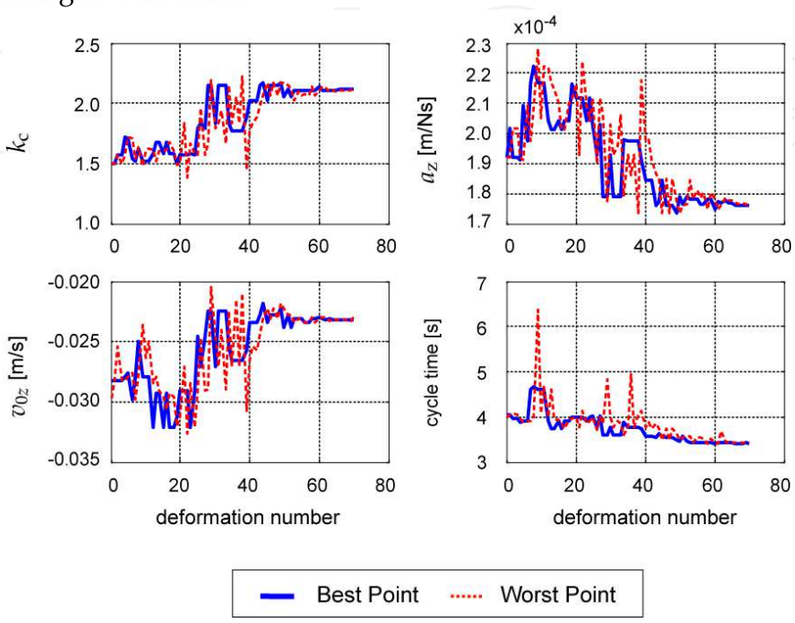


Fig. 6. Results of optimization for search motion

The cyclic motion and the pushing force need to be determined appropriately in the clutch assembly. For example, when the velocity of the cyclic motion is too high with soft force control along the insertion axis, the teeth on the clutch hub will fail to engage with the grooves on the clutch plate. When the clutch hub pushes the clutch plate with too large force, the pressed plate tends to move along with the hub. As shown in the above results, the obtained force control parameters through the optimization had a good balance between the velocity and the force and can deal with various plate arrangements.
The simulation whose results are shown in Fig. 6 took about 39[h] using a Windows PC with Pentium 4 CPU running at 2.8[GHz].

5.3 Integration of policies for insertion and search motions

An appropriate policy for the clutch assembly is constructed by integrating the policies for insertion and search motions that are obtained in the previous subsection. The limitation of the reaction forces is utilized as task knowledge and defines the states in which the reaction forces exceed their limit as terminal states of the task.

5.3.1 State space

A state space of assembly tasks is constructed by the current position of the robot, the reaction forces, and the robot's responsiveness. However, the number of states becomes enormous if all kinds of states are addressed. In clutch assembly, the reaction force along the

insertion axis, $f_{outz} \in R^6$, is the most effective state for recognizing that the teeth on the clutch hub becomes engaged with the proper grooves on the clutch plate. Therefore, the state space was confined as follows:

$$s = s\left(f_{outz}, d\right) \tag{13}$$

The reaction force f_{outz} was segmented into 62 states between the lower limit and the upper limit of it. The robot's responsiveness d was divided into two: $d_{\text{insertion}}$ which is the responsiveness for the insertion motion and d_{search} which is that for the search motion.

5.3.2 Actions

The policies of insertion and search motions were applied to the whole state space. A set of implemented actions was defined as follows:

$$\mathcal{A}\left(s_{\text{policy}}\right) = \left\{\boldsymbol{a}_{\text{insertion}}, \boldsymbol{a}_{\text{search}}\right\}, \tag{14}$$

where $\boldsymbol{a}_{\text{insertion}}$ and $\boldsymbol{a}_{\text{search}}$ are the damping control parameters that can effectively perform the insertion and search motions, respectively. When an action is selected, the damping control parameters expressed by the action are applied to the damping controller of the simulator.

In damping control, the target force, which is the force applied by the manipulated object on the environment in the steady state, is defined by the applied damping control parameters. The target force along the insertion axis of the insertion motion is about twice of that of the search motion, and the admittance along the insertion axis of the insertion motion is smaller than that of the search motion. Namely, the robot strongly presses the assembled object when the insertion motion is applied, compared with when the search motion is done.

5.3.3 Method for exploring new policy

In assembly tasks, it is difficult to obtain the model of the target task, i.e. to calculate the state transition probability $\mathcal{P}_{ss'}^{a}$ beforehand because of the uncertainties like the position errors of the robot and the friction between manipulated objects. Therefore, we adopted Q-learning (Sutton & Barto, 1998) in order to explore a new policy. Q-learning is one of the reinforcement learning techniques and can construct a policy without the model of the target task. The goal states were defined as the states in which the clutch assembly is successfully achieved. The error states were also determined as the states in which the reaction forces exceed their limit. If the robot reaches the error states, the simulation is stopped, and the task is regarded as a failure. In order to reduce calculation time for obtaining a new policy, the three clutch plate model was applied.

Rewards:

The robot selects and executes an action at each sampling time of its control system, and the sampling time Δt is 0.004 [s]. Thus, a reward −0.004 was given at each step. In addition, a penalty −4 was given when the task results in a failure.

Learning parameters:

	Position error (along x-axis)	Phase angle error
Type1	Alternately: ±0.5 [mm]	Alternately: +4, 0 [deg]
Type2	Alternately: ±0.5 [mm]	All plates: +4 [deg]
Type3	Alternately: ±0.5 [mm]	Alternately: ±2 [deg]
Type4	All plates: +0.5 [mm]	Alternately: +4, 0 [deg]
Type5	All plates: +0.5 [mm]	All plates: +4 [deg]

Table 2. Initial position/phase angle of the clutch plates

Type1	Type2	Type3	Type4	Type5	mean
1.18 [sec]	1.18 [sec]	1.16 [sec]	0.98 [sec]	0.75 [sec]	1.05 [sec]

Table 3. Cycle time of the clutch assembly based only on the search motion

The ε-greedy method was used for making experiences and in which actions are conducted randomly at the rate ε. The parameter of ε-greedy method and the learning parameters of Q-learning, α, were adopted as $\varepsilon = 0.1$ and $\alpha = 0.1$, respectively.

Uncertainties of the task:

A new policy for clutch assembly needs to handle various plate arrangements. Thus, simulations were performed against the five kinds of arrangements of the clutch plates described in Table 2. As a reference, Table 3 shows the cycle time obtained by the simulations with each plate arrangement in Table 2 based only on the policy of search motion.

5.3.4 Simulation results of integrating the policies for insertion and search motions

A new policy for the clutch assembly was developed by integrating two policies: for insertion motion and search motion, based on the conditions that are mentioned above. The simulation results are presented in Fig. 7. The horizontal axis represents the learning step. The vertical axis shows the average cycle time of ten trials. Here, the average is obtained without including the results of tasks failed. As shown in Fig. 7, the average cycle time was slightly shortened as the learning proceeded. In addition, the cycle time was extremely reduced compared with that obtained based only on the search motion presented in Table 3. It shows that integrating the insertion motion into the search motion is effective to the clutch assembly.

In Fig. 8, a result of the clutch assembly using the state action map obtained after 600 trials is shown. The last graph shows the action taken at each sampling time during the task. As shown in Fig. 8, the clutch hub was inserted through each clutch plate with a cyclic search motion and the insertion motion. The values of z and f_{outz} represent the fitting of the hub in each plate. When the teeth of the clutch hub is engaged with that of each clutch plate, the reaction force f_{outz} becomes small since only the frictional force is acting on the hub. From the result of the selected action, the policy of insertion motion was continuously selected while f_{outz} was almost zero. In fact, the effective policy was taken with perceiving the engagement of the objects based on the obtained state action map. Compared to the result based only on the search motion (Fig. 9), the hub is quickly inserted by selecting the policy of insertion motion. **Error! Reference source not found.**Fig. 9 presents the cycle time of the clutch assembly against the plate arrangements in Table 2 using the obtained state action. It

shows that the obtained new policy can also deal with various arrangements of the clutch plates.

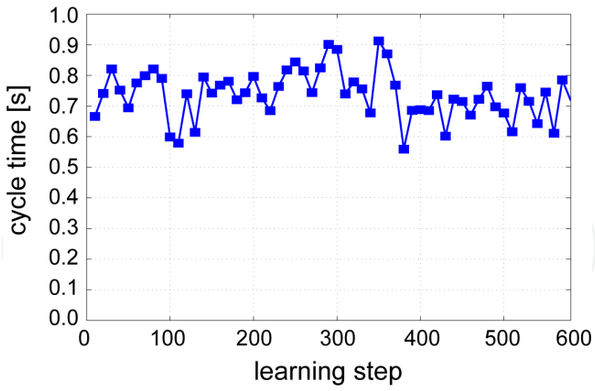


Fig. 7. Change of the cycle time through the learning

On the other hand, the insertion motion and the search motion were selected at random while the reaction force f_{outz} was acting on the clutch hub. In the insertion motion, the clutch hub strongly presses each clutch plate compared with the case of the search motion. If the insertion motion is chosen when the position and the phase angle of the hub are not matched to those of the objective plate, the task might result in a failure by the excessive reaction forces. Actually the reaction force in Fig. 8 is larger than that in Fig. 9. The learning was not converged after 600 trials, and the robot failed to perform the task because of the excessive reaction forces once in three times on average. The insertion motion and the search motion are similar motion in a certain part. In addition, the clutch assembly can be accomplished only with the search motion. It is hard to differentiate these motions while constructing a new policy. In future work, the reward setting should be improved, and a learning method also needs to be developed for obtaining a new policy efficiently based on similar policies.

6. Conclusion

In this paper, we proposed a step by step approach to planning robot motions for complex assembly tasks. It can be assumed that a complex assembly motion consists of some basic assembly motions, which can be accomplished with fixed force control parameters. Therefore, we firstly proposed an optimization method for basic assembly motions to obtain appropriate force control parameters that can efficiently achieve operations. Based on the results, we then proposed a policy integration method in order to generate complex assembly motions by utilizing optimized basic assembly motions.

These methods were applied to clutch assembly in this paper. Using the parameter optimization method, effective policies for insertion and search motions were obtained; and then, a new control policy for clutch assembly was developed by integrating these basic assembly motions. Based on the new policy obtained by the policy integration method, the effective motion was chosen with perceiving the engagement of the clutch hub and the clutch plates. The result shows that the proposed approach makes it possible to generate complex assembly motions that need some switching of force control parameters according to task states.

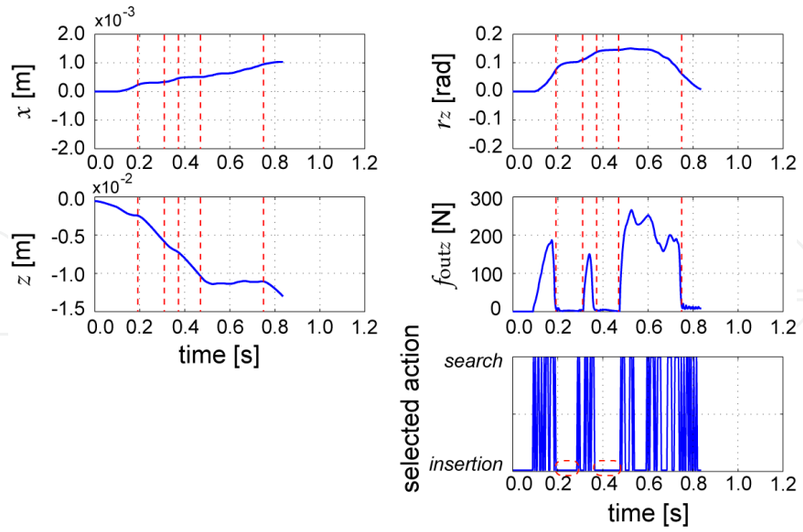


Fig. 8. Result of the clutch assembly based on the state action map obtained after 600 iterations

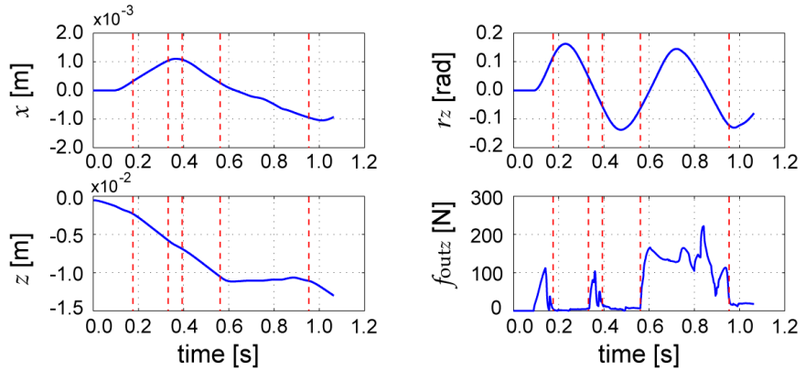


Fig. 9. Result of the clutch assembly based only on the search motion with the same clutch plate arrangement with that used in Fig. 8.

It is one of the future works to develop a learning method in order to efficiently generate a new policy based on similar policies. Although the approach was applied only to the clutch assembly in this paper, it will be applied to the other complex assembly tasks for more evaluation.

7. References

Bellman, R. (1957). *Dynamic Programming*, Princeton University Press

Gullapalli, V.; Franklin, J.A. & Benbrahim, H. (1994). Acquiring robot skills via reinforcement learning, *IEEE Control Systems*, Vol. 14, No. 1, pp. 13–24

Hirai, S.; Inatsugi, T. & Iwata, K. (1996). Learning of admittance matrix elements for manipulative operations, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 763–768

Huang, S. & Schimmels, J.M. (2003). Sufficient conditions used in admittance selection for force-guided assembly of polygonal parts, *IEEE Transaction on Robotics and Automation*, Vol. 19, No. 4, pp. 737–742

Lin, L-J (1991). Programming Robots Using Reinforcement Learning and Teaching, *Proceeding on the Ninth National Conference on Artificial Intelligence*, pp. 781–786

Minato, T. & Asada, M. (1998). Environmental Change Adaptation for Mobile Robot Navigation, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1859–1864

Press W.H.; Teukolsky, S.A.; Vetterling, W.T. & Flannery, B.P. (1992). *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.*, pp. 394–455, Cambridge University Press, ISBN-10: 0521750334

Schimmels, J.M. (1997). A linear space of admittance control laws that guarantees force-assembly, *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 5, pp. 656–667

Simons, J.; Van Brussel, H.; De Schutter, J. & Verhaert, J. (1982). A self-learning automation with variable resolution for high precision assembly by industrial robots, *IEEE Transaction on Automatic Control*, Vol. AC–27, No. 5, pp. 1109–113

Skubic, M. & Volz, R.A. (2000). Acquiring Robust, Force-Based Assembly Skills from Human Demonstration, *IEEE Transaction on Robotics and Automation*, Vol. 16, No. 6, pp. 772–781

Sutton, R.S. & Barto, A.G. (1998). *Reinforcement Learning: An Introduction*, The MIT Press, ISBN-10: 0262193981

Sutton R.S.; Precup, D. & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, *Artificial Intelligence*, Vol. 112, pp. 181–211

Suzuki, T.; Inagaki, S. & Yamada, N. (2006). Human Skill Modelling Based on Stochastic Switched Dynamics, *Proceeding of IFAC Conference on Analysis and Design of Hybrid Systems*, pp. 64–70

Tanaka, F. & Yamamura, M. (2003). Multitask Reinforcement Learning on the Distribution of MDPs, *Proceeding of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 1108–1113

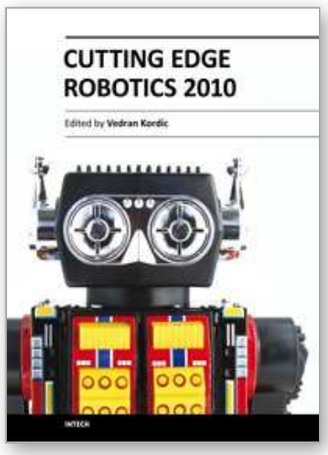
Thrun, S. & Mitchell, T.M. (1995). Lifelong Robot Learning, *Robotics and Autonomous Systems*, Vol. 15, pp. 25–46

Wei, J. & Newman, W.S. (2002). Improving robotic assembly performance through autonomous exploration, *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 3303–3308

Whitney, D.E. (1982). Quasi-static assembly of compliantly supported rigid parts, *Transaction of ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 104, pp. 65–77

Yamanobe, N.; Maeda, Y. & Arai, T. (2004). Designing of Damping Control Parameters for Peg-in-Hole Considering Cycle Time, *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 1129-1134

Yamanobe, N.; Fujii, H.; Arai, T. & Ueda, R. (2008). Motion Generation for Clutch Assembly by Integration of Multiple Existing Policies, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3218–3223



Cutting Edge Robotics 2010

Edited by Vedran Kordic

ISBN 978-953-307-062-9

Hard cover, 440 pages

Publisher InTech

Published online 01, September, 2010

Published in print edition September, 2010

Robotics research, especially mobile robotics is a young field. Its roots include many engineering and scientific disciplines from mechanical, electrical and electronics engineering to computer, cognitive and social sciences. Each of this parent fields is exciting in its own way and has its share in different books. This book is a result of inspirations and contributions from many researchers worldwide. It presents a collection of a wide range of research results in robotics scientific community. We hope you will enjoy reading the book as much as we have enjoyed bringing it together for you.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Natsuki Yamanobe, Hiromitsu Fujii, Tamio Arai and Ryuichi Ueda (2010). Motion Planning by Integration of Multiple Policies for Complex Assembly Tasks, Cutting Edge Robotics 2010, Vedran Kordic (Ed.), ISBN: 978-953-307-062-9, InTech, Available from: http://www.intechopen.com/books/cutting-edge-robotics-2010/motion-planning-by-integration-of-multiple-policies-for-complex-assembly-tasks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen