

# Online Trajectory Planning and Force Control for Automation of Surgical Tasks

Takayuki Osa, *Member, IEEE*, Naohiko Sugita, and Mamoru Mitsuishi, *Member, IEEE*

**Abstract**—Automation of surgical tasks is expected to improve the quality of surgery. In this paper, we address two issues that must be resolved for automation of robotic surgery: online trajectory planning and force control under dynamic conditions. By leveraging demonstrations under various conditions, we can model the conditional distribution of the trajectories given the task condition. This scheme enables generalization of the trajectories of spatial motion and contact force to new conditions in real time. In addition, we propose a force tracking controller that robustly and stably tracks the planned profile of the contact force by learning the spatial motion and contact force simultaneously. The proposed scheme was tested with bimanual tasks emulating surgical tasks that require online trajectory planning and force tracking control, such as tying knots and cutting soft tissues. Experimental results show that the proposed scheme enables planning of the task trajectory under dynamic conditions in real time. Additionally, the performance of the force control schemes was verified in the experiments.

**Note to Practitioners** — This study addresses the problem of motion planning and control for automation of surgical tasks. In surgical tasks, it is necessary to manipulate objects under conditions where positions or shapes of objects often change during the task. Thus, trajectories for surgical tasks need to be planned and updated according to the change in the conditions in real time. In this study, we propose a framework for learning both spatial motion and force profile from human experts. The proposed system can plan and update task trajectories in real time and robustly control the contact force under dynamic conditions. On the other hand, generalization of trajectories is limited to the conditions which are close to the conditions where the demonstrations were performed. In future work, we will investigate reinforcement learning approaches in order to enable autonomous improvement of the performance.

**Index Terms**—Motion planning, force control, surgical robot.

## I. INTRODUCTION

ROBOTIC surgery has established its position within medicine in recent years, and its benefits have been clarified through clinical studies [1]. On the other hand, there is still potential that is not leveraged in current commercialized systems. One such possibility is to automate surgical tasks in robotic surgery, which would reduce surgeons' mental and physical fatigue and improve the quality of surgery. This study addresses two issues that need to be resolved in order to achieve automation of robotic surgery: online trajectory planning and force control.

T. Osa is with Intelligent Autonomous Systems Group at TU Darmstadt, Germany (e-mail: takayuki@robot-learning.de).

N. Sugita and M. Mitsuishi are with Department of Mechanical Engineering, The University of Tokyo, 113-8656, Japan (e-mail: sugi@nml.t.u-tokyo.ac.jp; mamoru@nml.t.u-tokyo.ac.jp).

Manuscript accepted, February 2017.

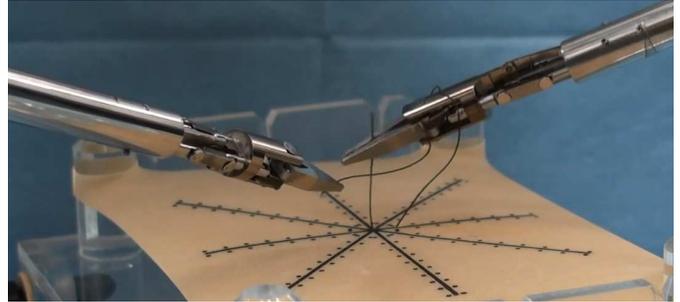


Fig. 1. Automation of knot tying in robotic surgery requires both online trajectory planning and force control.

The planning of task trajectories under dynamic conditions in real time is one of the major issues. When a robotic surgical system assists a surgeon through autonomous motions, it must collaborate with a human surgeon and work with movable and flexible objects such as threads and soft tissues during operations. For example, surgical knot tying involves making loops around a surgical instrument with a surgical thread (Fig. 1). If the surgical instrument to be looped is manipulated by a surgeon and moving during the task, the trajectory must be recalculated and updated online according to the motion of the instrument, thereby maintaining the topological feature of the trajectory. However, such online trajectory planning presents a difficult challenge, for which a solution has not yet been established.

The other issue is force control. Haptic information plays important roles in surgical operations. In some tasks where contacts between instruments and objects are necessary, the contact force must be explicitly controlled using the haptic information. For instance, when a surgeon cuts a membrane using a surgical knife, the contact force must be controlled to cut the target appropriately; or, when a surgeon makes a knot, a thread must be tightened with appropriate force to make a tight knot without breaking the thread. In these tasks, the desired profile of the contact force must be planned online and tracked robustly by the robotic system. Such planning and tracking of the contact force are essential to autonomously perform tasks that require explicit force control. When both online trajectory planning and force tracking control are necessary, the execution of the task becomes extremely challenging for many robotic systems.

In this paper, we present a framework for online trajectory planning and force control under dynamic conditions to achieve automation of surgical tasks. In our earlier work [2], [3], we presented a system that learns time- and space-dependent motions from trajectories demonstrated under vari-

ous conditions. Although such online trajectory planning was addressed in our prior work, we only showed how to learn spatial motions from the demonstration and the quantitative evaluation of the scheme was not exhaustive. The scheme proposed in this paper enables planning trajectories of both spatial motion and contact force in real time under dynamic conditions by leveraging demonstrations under various conditions. By learning the distribution of the spatial motions and contact force simultaneously, the developed system updates the profile of the contact force in real time under dynamic conditions and robustly tracks the planned contact force. The proposed scheme enables autonomous execution of tasks that require force control, such as cutting soft tissues and tying a knot.

The main contributions of this paper are (i) presentation of the framework to achieve online trajectory planning and force tracking control for surgical task automation; (ii) describing implementation details of the proposed scheme and (iii) quantitative evaluation of the developed scheme through experiments with tasks that emulate surgical tasks.

This paper is structured as follows. The next section describes the previous studies related to trajectory planning by learning from demonstrations and force control. Section III describes the details of the proposed trajectory planning scheme and force tracking control. Section IV provides the experiments and simulations to evaluate the performance of our online trajectory planning and force control. Section V discusses the results presented in this study. The conclusions can be found in Section VI.

## II. RELATED WORK

### A. Trajectory Planning for Automation of Surgical Tasks

Numerous studies have addressed the automation of surgical tasks [4]–[10]. Learning-from-demonstration (LfD) approaches have often been used in these previous studies. LfD is an approach that leverages human demonstration to achieve autonomous motion in robotic systems [11]. Robotic systems for surgery often employ a master-slave system in which an operator inputs motions from the master system for the slave manipulator to execute. In this typical configuration, the motion inputs from the human operator are easily obtained; therefore, LfD is a reasonable approach to achieve autonomous motion in robotic surgery. Mayer et al. developed a scheme to model demonstrated trajectories, using a recurrent neural network [5]. Berg et al. developed a system that learns the motion of tying a knot from multiple demonstrations and performs it faster than the given demonstration [4]. Recently, Murali et al. proposed an autonomous assistance system based on observation of human experts [10]. However, these schemes do not address the generalization of the demonstrated trajectories to new situations. Since the conditions in a surgical operation vary for each operation, the task trajectory must be planned according to the given condition.

Generalization of demonstrated trajectories is one of primary problems in LfD. Khansari et al. developed a scheme to model demonstrated motions as a time-invariant dynamic system using Gaussian mixture models (GMMs) [12], [13].

This scheme allows the motion to be generalized for a new state in the robotic system. In a reinforced learning framework, schemes to adapt the motion primitives for new situations were proposed in [14]. These studies focused on motion that can be defined by the starting state and the end state. However, some surgical operations will be difficult to describe using such frameworks designed primarily for point-to-point tasks. Making a loop around an instrument in order to tie a knot, for instance, is difficult to decompose into point-to-point motions, because its trajectory is seamless, and the topological shape of the entire trajectory must adapt to new situations. Schulman et al. developed a scheme to generalize demonstrated trajectories for new situations [9], [15], [16]. Their scheme is not limited to point-to-point motions, and the geometrical mapping from demonstrated situation to a new situation is computed by non-rigid registration. This scheme was applied to a suturing task in a simplified situation [9]. However, finding the best demonstration for the new condition is time-consuming; hence, their method is difficult to use for online trajectory planning under dynamic conditions. Recent work developed methods for learning task-parameterized policies such as [17] and [18]. Although these methods can be used to generalize the learned skills, they are not focused on online trajectory planning. The method in [18] estimate the model parameter using GMMs, and the estimated model parameters are utilized to generate trajectories. This method is indirect compared to our approach, which learns the direct mapping from the task condition to the trajectory. As a consequence, the method in [18] may require more computational time when compared to our approach. Parachos et al. proposed a framework of Probabilistic Movement Primitive (ProMP) to represent the distribution of demonstrated trajectories [19]. This method enables the generalization of the trajectory to new via-points or new goal points by conditioning the distribution of the trajectory. However, the study in [19] does not discuss the distribution of the situation where demonstrations are performed. Recent work in [20] proposed a trajectory optimization framework to generate collision-free motions by learning from demonstration. Although the method in [20] can generalize the trajectory to new scenes, it does not address online trajectory planning.

Although many studies have reported schemes for generalizing demonstrated trajectory for new via-points or goal points, specifying the via-points or goal points are not always trivial in motion planning for surgical tasks. For example, when the system needs to make a loop with a surgical thread around the instrument, it is challenging to plan via-points according to the position of the surgical instrument, maintaining the topological feature of the trajectory.

In this study, we extend an approach of learning from demonstrations under various situations in [2]. In the scheme in [2], the conditional distribution of the demonstrated trajectory given the situation is modeled using Gaussian Process Regression (GPR). This enables the modeling of time- and space-dependent task trajectories and their generalization to new situations. We extend this scheme to learning and planning for the desired force in real time during the task. We will show that the proposed scheme enables planning for both

spatial motion and force profile in real time under dynamic conditions.

### B. Planning and Tracking the Contact Force

Force tracking is a classical control problem in the field of robot manipulation [21]–[24]. Seraji et al. described schemes for force tracking control by impedance control and admittance control [25], [26]. Tracking the desired force requires consideration of the stiffness of the contact object and the ability to predict the motion for tracking the desired force. However, this problem remains non-trivial despite decades of research.

The learning and planning of the contact force trajectory have been addressed by some studies in the literature regarding learning from demonstration and reinforcement learning. Some studies reported robotic systems that learn the required impedance from demonstrations to react to the force exerted on the manipulator [27]–[29]. However, surgical tasks require explicit force control beyond the mere ability to react to the contact force caused by a disturbance. Reinforcement learning methods for tasks that require force control have been reported in [30], [31]. These previous studies proposed schemes for adapting motion parameters to new situations to achieve appropriate force and motion control. However, these methods require iterative learning to track the planned contact force in a new situation. Recently, Lee et al proposed a scheme for learning force control from demonstration in [32]. However, this method is based on the trajectory planning scheme in [9], and this scheme is not applicable to the task that require online trajectory planning due to its computational cost. Likewise, Rozo et al. proposed a method for learning and controlling both force profile and spatial motions [33]. The method in [33] is based on the trajectory learning in [18], and the methods in [33] may not be suitable for online trajectory planning for the same reason as the methods in [18].

In this work, we leverage the models learned from demonstrations to plan the trajectory of the contact force and track the planned contact force robustly. By learning the trajectory of the contact force and spatial position simultaneously, the proposed scheme plans the desired contact force and estimates the spatial motion to track the planned contact force. This trajectory learning scheme was then combined with the concept of the sliding mode control to achieve robust force tracking under dynamic conditions [34].

## III. PROPOSED METHOD

### A. Overview of System and Algorithm

1) *System*: We integrated the proposed online trajectory planning and force control schemes with a teleoperated robotic system developed for laparoscopic surgery [35], [36] (Fig. 2). The robotic surgical system consists of master and slave systems, in which the motions input at the master system by an operator are executed by the slave manipulator. In this type of system, the LfD approach is expected to be efficient since the trajectory executed by experts is easily obtained. A standard system for robotic surgery employs rate control, where the



Fig. 2. A robotic system for teleoperated surgery.

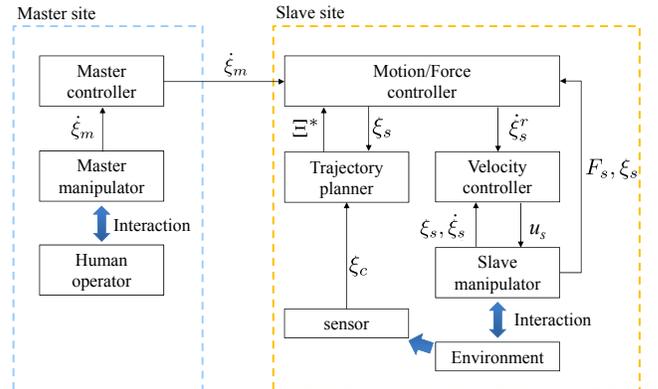


Fig. 3. Overview of proposed system.

motion of the slave manipulator is proportional to the motion input at the master system as

$$\dot{\xi}_s = K_{ms}\dot{\xi}_m, \quad (1)$$

where  $\dot{\xi}_s$ ,  $\dot{\xi}_m$ , and  $K_{ms}$  are the velocity of the slave manipulator, velocity of the master manipulator, and motion scaling gain between the master and slave systems, respectively. To achieve this control, a velocity/position control is often implemented in a robotic surgery system. Hence, we developed a motion planning and execution system based on the master-slave system with a velocity controller.

Fig. 3 shows the structure of the proposed system. A trajectory planner and a motion-force controller are implemented in the master-slave system. The trajectory planner estimates a trajectory  $\Xi_s^*$  required to perform the learned task according to a given condition  $\xi_c$ . The motion-force controller is implemented as an outer loop based on an underlying velocity controller as shown in Fig. 3. To track the planned trajectory, the motion-force controller determines the reference velocity for the velocity controller  $\dot{\xi}_s^r$  using the feedback of the sensory information, such as the measured contact force  $F_s$ . The torque input,  $u_s$ , to each joint in the slave manipulator is determined by the underlying velocity controller.

Force control is usually categorized into two types: impedance control and admittance control. Admittance control works well for soft environments and robots with a high gear ratio, whereas impedance control works well for hard environments and robots with a low gear ratio [37]. Surgical robots often have a high gear ratio and a velocity controller to achieve precise motions by rate control. In addition, the contact force is relatively small in laparoscopic surgery because the robotic manipulator is expected to cope with soft tissues. For

these reasons, we developed a force control strategy based on admittance control in this study. Hence, the force controller was implemented as an outer loop with an inner loop that implemented the velocity controller.

2) *Learning and Planning of Trajectory*: In the proposed trajectory planning scheme, we assume that demonstrations performed by a human expert under various task conditions  $\xi_c$  are available. The task condition was fixed for a single demonstration. Thereafter, the system models the conditional distribution of the demonstrated trajectories given the task condition at time  $t$  as  $P(\xi_s(t)|\xi_c)$ . Subsequently, if the new task condition is given as  $\xi_c^*$ , the optimal task trajectory  $\Xi_s^*$  under the given condition can be predicted as a conditional expectation as

$$\Xi_s^* = [\xi_s^*(0), \dots, \xi_s^*(N)], \quad (2)$$

where

$$\xi_s^*(t) = \mathbb{E}[\xi_s(t) | \xi_c^*] \quad (3)$$

and  $N$  is the total number of time steps of the trajectory. For modeling the distribution of the demonstrated trajectories, the system normalizes the demonstrated trajectories in the time domain and estimates the reference trajectory. In this step, dynamic time warping (DTW) was employed to eliminate the temporal variance of the demonstrated trajectories [38], and the Kalman smoother was used to estimate the reference trajectory [4], [39], [40]. Thereafter, the distribution of the demonstrated trajectories is modeled using GPR.

During the task execution, the task condition  $\xi_c$  is constantly measured and sent to the trajectory planner, and the system plans and updates the trajectory  $\Xi_s^*$  in real time according to the task condition  $\xi_c$ .

This trajectory learning scheme can be used for learning trajectories of various sensory information. In this work, we applied this scheme to learn both spatial motion and contact force simultaneously. The details of our trajectory planning scheme are described in Section III-B.

3) *Motion and Force Control*: In this study, we implemented the force tracking control scheme, which is a prerequisite for tasks that require force control such as cutting soft tissues and tying knots. These types of motions involve both spatial motions and contact force. The proposed system learns the trajectory of the contact force and the spatial position simultaneously from human demonstration by representing the state of the system as follows:

$$\xi_s = \begin{bmatrix} \xi_{space} \\ F_s \end{bmatrix} \quad (4)$$

where  $\xi_{space}$  and  $F_s$  represent the kinematic state of the system and state of the contact force, respectively. The motion-force controller determines the reference velocity  $\dot{\xi}_s^r$  to track the planned trajectory of the contact force  $F_s^*$  by leveraging the planned spatial trajectory  $\xi_{space}^*$  under the task condition  $\xi_c^*$ . This enables planning of the contact force trajectory under a given condition and robust tracking of the planned trajectory.

In robotic surgery, importance of spatial motion control and force control is different between tasks. In addition, the workspace is severely limited in robotic surgery, and it is

difficult to measure the contact force loaded on the end-effector in every direction. Therefore, we need to combine spatial motion control and force tracking control according to the availability of force information and the type of the given tasks. We implemented spatial motion control and force tracking control in the developed system, and both controls were used in the experiments to verify the proposed scheme as discussed in Section IV. We employed the spatial motion control, as presented in our prior work [2]. For self-containment, we describe the spatial motion control in Section III-C. The novel force tracking control is then described in Section III-D.

### B. Online Trajectory Planning by Learning from Demonstration

1) *Normalization in Time Domain and Estimation of Reference Trajectory*: Demonstrated trajectories usually contain temporal variations because the execution speed of the task varies for each demonstration. Therefore, these variations must be eliminated to model the distribution of the trajectory. For this purpose, we normalize the demonstrated trajectories in the time domain and estimate the reference trajectory by using the method described in [4], [39], [40]. As in [4], [39], [40], we regard the demonstrated trajectory as a noisy ‘‘observation’’ of the ‘‘reference’’ trajectory. We assume a linear system with process noise and measurement noise, each of which has a Gaussian distribution. We represent the state of the system in the reference trajectory  $z(t)$  as follows:

$$z(t) = \begin{bmatrix} \xi_s^{\text{ref}}(t) \\ u^{\text{ref}}(t) \end{bmatrix} \quad (5)$$

where  $\xi_s^{\text{ref}}(t)$  is the state of the robotic system at time  $t$ , and  $u^{\text{ref}}(t)$  is the input to the system in the reference trajectory. As in (4),  $\xi_s(t)$  can be represented by kinematic state (e.g. the Cartesian or configuration space) and sensory outputs (e.g. force/torque sensor). We represent the system as a stochastic model as

$$z(t+1) = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} z(t) + w(t), \quad (6)$$

where  $A$  is the state matrix, and  $B$  is the input matrix, and  $w(t)$  is the process noise of the system. In our system,  $\xi_s(t)$  is given by the state of the slave manipulator, and  $u(t)$  is given by the input from the master manipulator. Since we assume the velocity-velocity control as in (1),  $A$  is a identity matrix,  $B$  is given by  $K_{ms}$ , and  $u(t) = \dot{\xi}_m$ . We then assume that the  $i^{\text{th}}$  demonstrated trajectory  $y^i$  is drawn from this stochastic system, and  $y^i$  is given as

$$y^i(t) = \begin{bmatrix} \xi_s^i(t) \\ u^i(t) \end{bmatrix}, \quad (7)$$

where  $\xi_s^i(t)$  is the state of the system and  $u^i(t)$  is the input to the system at time  $t$  in the  $i^{\text{th}}$  demonstrated trajectory. When  $M$  demonstrated trajectories are given, the reference trajectory  $z$  can be associated with the demonstrated trajectories  $y^1, \dots, y^M$  as

$$\begin{bmatrix} y^1(\tau_1^1) \\ \vdots \\ y^M(\tau_1^M) \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} z(t) + v(t), \quad v(t) \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} R^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R^M \end{bmatrix} \right), \quad (8)$$

where  $\tau_t^i$  is the time mapping from the demonstrated trajectory  $y^i$  to the reference trajectory  $z$ , and  $R^i$  is the variance of the  $i^{\text{th}}$  trajectory from the reference trajectory. Therefore, we can estimate the reference trajectory  $z$  by using the Kalman smoother. In our implementation, we set  $R^i = I$  for  $i = 1, \dots, M$  so that the importance of each demonstrated trajectory is equal. To estimate the reference trajectory, we initialized the time alignment as  $\tau_t^i = \frac{t}{N}T^i$ , where  $N$  is the total number of time steps of the reference trajectory and  $T^i$  is the length of the  $i^{\text{th}}$  demonstrated trajectory.

After the time alignment is updated, the reference trajectory is also updated using the updated time alignment. By alternately repeating the updating of the reference trajectory and the time alignment, we can properly align the demonstrated trajectories in the time domain.

2) *Modeling the conditional distribution of the demonstrated trajectories given the task condition:* We employed GPR, a statistical approach, to model the distribution of the demonstrated trajectories as a function of the task condition. Although other regression methods such as Gaussian mixture regression [41], [42] and locally weighted regression [43] are available, we selected GPR because it handles nonlinear relationships using relatively little training data [44]. To model the deviation of the demonstrated trajectory  $y$  from the reference trajectory  $z$  by using GPR, we represent the dataset as follows:

$$X = \begin{bmatrix} (\xi_c^1)^T \\ \vdots \\ (\xi_c^M)^T \end{bmatrix}, Y_t = \begin{bmatrix} (\xi_s^1(\tau_t^1) - \xi_s^{\text{ref}}(t))^T \\ \vdots \\ (\xi_s^M(\tau_t^M) - \xi_s^{\text{ref}}(t))^T \end{bmatrix}, \quad (9)$$

$$X^* = [(\xi_c^*)^T], Y_t^* = [(\xi_s^*(t) - \xi_s^{\text{ref}}(t))^T]$$

where  $X$  is the set of task conditions,  $Y_t$  is the set of deviations of the demonstrated trajectories from the reference trajectory at the  $t^{\text{th}}$  time step,  $X^*$  is the test task condition, and  $Y_t^*$  is the deviation from the reference trajectory under the test condition  $X^*$  at the  $t^{\text{th}}$  time step. When the state of the system is  $D$  dimensional, the dimension of  $Y_t$  is  $M \times D$ . In GPR, the joint distribution of  $Y_t$  and  $Y_t^*$  is modeled as a Gaussian distribution with zero-mean as

$$p \begin{pmatrix} Y_{t,j} \\ Y_{t,j}^* \end{pmatrix} \sim \mathcal{N} \left( 0, \begin{bmatrix} G(X, X) + \sigma_n^2 I & G(X, X^*) \\ G(X^*, X) & G(X^*, X^*) + \sigma_n^2 I \end{bmatrix} \right) \quad (10)$$

where  $Y_{t,j}$  is the  $j^{\text{th}}$  column of  $Y_t$ ,  $Y_{t,j}^*$  is the  $j^{\text{th}}$  column of  $Y_t^*$ , and  $\sigma_n$  is a constant.  $G$  is the kernel matrix defined as follows:

$$\begin{aligned} G(X, X) &\in \mathbf{R}^{M \times M}, (G(X, X))_{i,j} = g(\xi_c^i, \xi_c^j), \\ G(X^*, X) &\in \mathbf{R}^{1 \times M}, (G(X^*, X))_{1,j} = g(\xi_c^*, \xi_c^j), \\ G(X, X^*) &\in \mathbf{R}^{M \times 1}, (G(X, X^*))_{i,1} = g(\xi_c^i, \xi_c^*), \\ G(X^*, X^*) &\in \mathbf{R}, G(X^*, X^*) = g(\xi_c^*, \xi_c^*), \end{aligned}$$

where  $\xi_c^i$  is the  $i^{\text{th}}$  demonstrated task condition, and  $g(x_i, x_j)$  represents the kernel function. In our implementation, we used the square exponential kernel function defined as

$$g(x_i, x_j) = \sigma_f \exp \left( -\frac{1}{2l^2} (x_i - x_j)^T (x_i - x_j) \right), \quad (11)$$

where  $\sigma_f$  and  $l$  are parameters that define the shape of the kernel function. The performance of the GPR depends on the selection of the hyperparameters  $[\sigma_f, \sigma_n, l]$ , which are obtained by maximizing the marginal likelihood defined as

$$\log p_{\text{marg}} = -\frac{1}{2} y_j^T G(X, X) y_j - \log \det |G(X, X)| - M \log 2\pi. \quad (12)$$

For details of selecting the hyperparameters of GPs, please refer to [44]. Using the models described above, the conditional distribution of  $Y_{t,j}^*$  is expressed as:

$$p(Y_{t,j}^* | Y_{t,j}, X, X^*) \sim \mathcal{N}(\mu_j^*, \Sigma_j^*) \quad (13)$$

where

$$\begin{aligned} \mu_j^* &= G(X^*, X)(G(X, X) + \sigma_n^2 I)^{-1} Y_{t,j} \\ \Sigma_j^* &= G(X^*, X^*) + \sigma_n^2 I \\ &\quad - G(X^*, X)(G(X, X) + \sigma_n^2 I)^{-1} G(X, X^*) \end{aligned} \quad (14)$$

Therefore, the trajectory under the given task condition is estimated as

$$\begin{aligned} \xi_{s,j}^*(t) &= \xi_s^{\text{ref}}(t) + \mathbb{E}[Y_{t,j}^* | Y_{t,j}, X, X^*] \\ &= \xi_s^{\text{ref}}(t) + (G(X^*, X)(G(X, X) + \sigma_n^2 I)^{-1} Y_{t,j})^T \end{aligned} \quad (15)$$

for  $j = 1, \dots, D$ , where  $\xi_{s,j}^*(t)$  is the state of the system in the  $j^{\text{th}}$  dimension at the  $t^{\text{th}}$  step in the trajectory predicted for the task condition  $X^*$ . The entire trajectory under the given task condition is estimated by computing (15) for  $t = 0, \dots, N$ . Since we employed the standard single output GP, we require  $N \times D$  models to obtain the whole trajectory.

Once the hyperparameters  $[\sigma_f, \sigma_n, l]$  and covariance matrix  $G(X, X)$  are obtained in the offline phase, the trajectory planning requires computing (15) for  $t = 0, \dots, N$ , which is a matrix computation with relatively little computational cost. Therefore, the proposed scheme enables planning of the trajectory under dynamic task conditions with a sufficiently short computation time.

### C. Spatial Motion Control for tracking the planned trajectory

1) *Control Scheme for Spatial Motion:* We present the spatial motion control scheme to stably track the spatial trajectory which is constantly updated during the task. This controller is necessary for tasks where the planned spatial motion needs to be achieved robustly.

The motion controller is implemented as an outer loop based on the underlying velocity controller, and it determines the reference velocity  $\xi_s^r$  in order to track the task trajectory in the proposed system (Fig. 3). When the task trajectory is updated during the task execution, the desired trajectory of the manipulator changes discontinuously. Given such changes, the planned trajectory cannot be input to the velocity controller directly as a reference value since this would cause unstable system behavior. Tracking the updated trajectory stably and robustly requires that we determine the appropriate reference velocity for the velocity controller. For this purpose, we used the concept of sliding mode control [34]. In this framework, the desired system behavior is expressed as a sliding surface as follows:

$$S(t) = \xi_s(t) - \xi_s^*(t) = 0 \quad (16)$$

where  $\xi_s(t)$  is the kinematic state of the system at time  $t$  and  $\xi_s^*(t)$  is the kinematic state at time  $t$  obtained from the planned trajectory. The desired velocity for tracking the planned trajectory is expressed as follows:

$$\dot{\xi}_s^r = \dot{\xi}_s^* - K \cdot \text{sat}(\xi_s(t) - \xi_s^*(t)) \quad (17)$$

where  $K = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_n \end{bmatrix}$ ,  $n$  is the dimension of  $\xi_s(t)$ , and  $k_i > 0$  for  $i = 1, \dots, n$ . The saturation function  $\text{sat}(S)$  is defined as

$$\text{sat}(S) = \begin{bmatrix} \text{sat}(s_1/c_1) \\ \dots \\ \text{sat}(s_n/c_n) \end{bmatrix}, \quad (18)$$

where  $S = [s_1, \dots, s_n]^T$ ,

$$\text{sat}(x_i) = \begin{cases} 1 & (x_i > 1) \\ x_i & (|x_i| < 1) \\ -1 & (x_i < -1) \end{cases} \quad (19)$$

and  $[c_1, \dots, c_n]$  are positive constants that determine the boundary layer on the sliding surface.

In (17), the first term represents the feed-forward control input to ensure the system slides on the sliding surface; the second term represents the feedback control input to ensure the system converges to the sliding surface.

2) *Stability of the Control Scheme*: In the framework of the sliding mode control [34], the system converges to the sliding surface within a finite time if the following condition is satisfied for a certain positive value  $[\eta_1, \dots, \eta_n]$ ,

$$\frac{d}{dt} s_i^2 = -\eta_i |s_i|. \quad (20)$$

From (16) and (17), we obtain the following equations.

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} s_i^2 &= s_i \cdot \dot{s}_i \\ &= -k_i s_i \cdot \text{sat}(s_i) \end{aligned} \quad (21)$$

When  $|s_i| > c_i$ , (21) becomes

$$\frac{1}{2} \frac{d}{dt} s_i^2 = -k_i |s_i|. \quad (22)$$

Therefore, the system reaches the neighborhood of the sliding surface within a finite time. Meanwhile, when  $|s_i| < c_i$ , (21) can be rewritten as

$$\frac{1}{2} \frac{d}{dt} s_i^2 = -k_i s_i \cdot s_i. \quad (23)$$

Thus, the system exponentially converges to the sliding surface in the neighborhood of the sliding surface. Therefore, it converges from any state to the desired trajectory within a finite time and can perform tracking in stable manner according to the control law in (17).

#### D. Force Tracking Control by learning both spatial motion and contact force

1) *Control Scheme for Force Tracking Control*: In this section, we describe the force tracking control scheme for tasks that require force control such as cutting soft tissue and tying a knot. The execution of such a task involves two main

challenges: planning the desired trajectory of the contact force, and tracking the planned contact-force trajectory.

These problems can be solved by extending the trajectory planning scheme described in the previous section. The system simultaneously learns the spatial position trajectory of the manipulator and the contact force by defining the system state as in (4). By using this redundant expression, the system learns the trajectory of the spatial position and the contact force simultaneously using the method described in the previous section. Given the task condition  $\xi_c^*$ , the trajectory is estimated as  $\xi_s^* = [\xi_{\text{space}}^*, F_s^*]^T$  where  $\xi_{\text{space}}^*$  is a spatial trajectory and  $F_s^*$  is the desired contact force. Thus, the desired value of the contact force can be planned by using the online trajectory planning scheme described in the previous section.

As we described in the previous section, we use the force tracking controller based on admittance control; the force controller is implemented as an outer loop based on an underlying velocity controller. In this study, we consider the contact force exerted at the tip of the end effector, which is given by  $F_s = [F_x, F_y, F_z]$ . To track the planned contact force, the force tracking controller determines the appropriate reference velocity for the velocity controller. Here, we define the error function  $e$  as follows:

$$e = F_s - F_s^* \quad (24)$$

where  $F_s$  is the current value of the contact force. To design the force tracking control scheme, we used the concept of the sliding mode control again. The control law to achieve the robust force tracking control in a linear system is expressed as follows:

$$u_f = \hat{u} - k \text{sat}(e/e_0) \quad (25)$$

where  $u_f$  is the control input,  $\hat{u}$  is the control input that achieves  $\dot{e} = 0$ ,  $\text{sat}(e/e_0)$  is the saturation function defined as in (19),  $k$  is a feedback gain, and  $e_0$  is a constant that determines the boundary layer on the sliding surface.

In our force control framework, the desired contact force  $F_s^*$  is planned under the task condition  $\xi_c^*$ , and the reference velocity for the velocity controller,  $\dot{\xi}_s^r$ , can be considered as a control input  $u_f$ . When the contact object is modeled as a spring and the state of the system is given by the translational position as  $\xi_{\text{space}} = [x, y, z]$ , the trajectory learned from the demonstration should satisfy the following equation:

$$\dot{F}_s^* = K^{\text{demo}} \dot{\xi}_{\text{space}}^* \quad (26)$$

where  $\xi_s^* = [\xi_{\text{space}}^*, F_s^*]^T$  is the planned trajectory under the given condition  $\xi_c^*$ , and  $K^{\text{demo}}$  is the stiffness of the contact object under the condition in which the demonstration was performed. Assuming that (26) holds,  $\hat{u}$  is replaced with  $\dot{\xi}_{\text{space}}^*$  in (25). Hence, the reference velocity for the velocity controller  $\dot{\xi}_s^r$  is determined as follows:

$$\dot{\xi}_s^r = \dot{\xi}_{\text{space}}^* - k \text{sat} \left( \frac{F_s - F_s^*}{e_0} \right) \quad (27)$$

As in (17), the first term represents the feedforward input, and the second term represents the feedback input. Although it is known that the transition between motions without contact to motions with contact can be unstable [45], the proposed

control scheme does not require switching control schemes for dealing with contacts.

2) *Stability of the Proposed Force Tracking Control*: The robustness of the proposed force tracking control law in (27) is shown below. We introduce  $K^{\text{test}}$  as the stiffness of the contact object under the given test condition. We then assume that the difference of the object stiffness between the test condition and the demonstration condition is bounded as

$$|(K^{\text{test}} - K^{\text{demo}})\xi_{\text{space}}^*| < \Delta, \quad (28)$$

where  $\Delta$  is a constant that indicates the upper bound of the difference between the test condition and the demonstration condition. From (27) and (28), the following relationship can be obtained.

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} e_i^2 &= e_i \cdot \dot{e}_i \\ &= e_i \cdot (\dot{F}_{s,i} - \dot{F}_{s,i}^*) \\ &= e_i \cdot \left( K^{\text{test}} \left( \xi_{\text{space},i}^* - k \text{sat}(e_i/e_0) \right) - K^{\text{demo}} \xi_{\text{space},i}^* \right) \\ &\leq |e_i| \cdot \Delta - K^{\text{test}} k e_i \cdot \text{sat}(e_i/e_0) \end{aligned} \quad (29)$$

where  $e_i$  denotes the  $i^{\text{th}}$  component of  $e = F_s - F_s^*$ . When  $|e_i| > e_0$ , (29) is rewritten as:

$$\frac{1}{2} \frac{d}{dt} e_i^2 \leq |e_i| \cdot \Delta - K^{\text{test}} k |e_i| \quad (30)$$

Hence, if we set  $k$  as  $k > \frac{\Delta + \eta}{K^{\text{test}}}$ , the condition (20) is satisfied. When  $|e_i| \leq e_0$ , (29) is rewritten as:

$$\frac{1}{2} \frac{d}{dt} e_i^2 \leq |e_i| \cdot \Delta - K^{\text{test}} k e_i^2 \quad (31)$$

Therefore, if we set  $k$  as  $k > \frac{\Delta + \eta}{K^{\text{test}}}$ , the norm of the error function  $|e_i|$  monotonically decreases to  $\frac{\Delta}{K^{\text{test}} k}$ . Thus, the proposed force tracking scheme enables robust tracking of the planned force trajectory with a disturbance by setting  $k$  sufficiently large.

Obviously, (28) is rewritten as

$$|K^{\text{test}} - K^{\text{demo}}| < \frac{\Delta}{|\xi_{\text{space}}^*|}. \quad (32)$$

Hence, if the stiffness of the contact object in the test scene satisfies the condition of (32), the system robustly tracks the planned contact force.

#### IV. EXPERIMENTS

We evaluated the developed system through experiments with tasks that emulate tying a knot and cutting soft tissues. First, we describe the experiment to evaluate the online trajectory planning scheme. Next, the performance of the proposed force tracking scheme is discussed.

##### A. Evaluation of Trajectory Planning Scheme

1) *Evaluation with a unimanual looping task*: The first experiment was performed to evaluate the performance of the proposed trajectory planning scheme. In this experiment, the system learned a unimanual DOUBLE LOOP task, which involves making a loop around the instrument on the left with a thread held by an instrument on the right. This task was

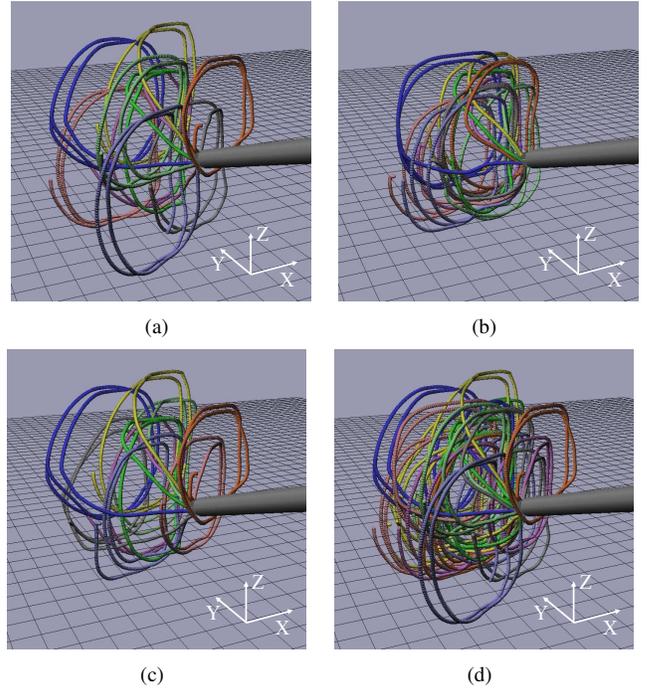


Fig. 4. Datasets for learning DOUBLE LOOP task: (a) dataset1, (b) dataset2, (c) dataset3, and (d) dataset4. Specific conditions for each dataset are shown in Appendix.

designed to demonstrate trajectory generation for a time- and space-dependent task that required adaptation of the topological shape of the trajectory to the environmental conditions.

In this experiment, in this unimanual task, the state of the system is given by the position of the right instrument as

$$\xi_s = [x_r, y_r, z_r] \quad (33)$$

where  $[x_r, y_r, z_r]$  is the positions of the tip of the right instruments. The task condition was defined as the position of the left instrument, as given below :

$$\xi_c = [x_l, y_l, z_l] \quad (34)$$

where  $[x_l, y_l, z_l]$  is the positions of the tip of the left instruments. The position of the left instrument was fixed for a single demonstration. The transformation between the coordinates of the right and left instruments was unknown. In this experiment, we set  $N = 100$  in (15).

We used the four sets of demonstrations shown in Fig. 4 to test whether the quality of the dataset affected the results. Nine demonstrations each were performed for dataset1, dataset2, and dataset3. Among these datasets, dataset1 had the largest variance, while dataset2 had the smallest variance (Fig. 4). Dataset4 includes all of the demonstrations used in dataset1, dataset2, and dataset3. Specific conditions for each dataset are shown in Table I - IV in the Appendix. The initial position of the right instrument was the same throughout this experiment.

To test the performance of trajectory planning, we varied the distance between the left and right instruments. The distance was changed from 10 to 40 mm along the Z and Y axes. To quantify the smoothness of the planned trajectories, we computed the norm of jerk of the planned trajectory as,  $\frac{1}{T} \sum_{t=0}^T \|\ddot{\xi}_s(t)\|$ .

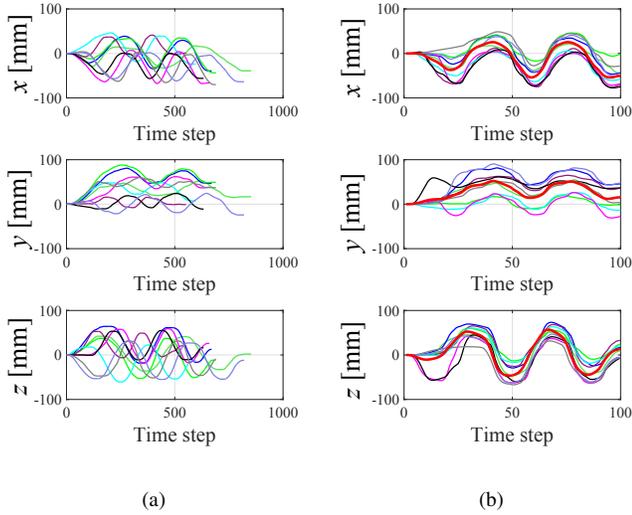


Fig. 5. Results of dataset1 (a) before and (b) after the DTW process. The red thick lines in (b) represent the reference trajectory  $z$ .

The result of the DTW is shown in Fig. 5. The thick red line in Fig. 5(b) represents the reference trajectory estimated by the Kalman smoother. Although the demonstrated trajectories contain temporal variations (Fig. 5(a)), the motion was synchronized with the reference trajectory after the DTW process (Fig. 5(b)). The distribution of the demonstrated trajectories was modeled by GPR from the data shown in Fig. 5(b).

The trajectories planned in the simulation are visualized in Fig. 6. The trajectories show that they were successfully planned under the given conditions when the dataset1 was used as demonstration. On the other hand, when the dataset2 was used, the trajectories became unnecessarily wavy if the condition was far from the demonstrated condition as  $\xi_c = [0, 0, 40]$  shown in Fig. 6 (h). Such unnecessary motions are not preferable because they may lead to collisions with surrounding objects during an actual surgery.

The jerk of the planned trajectories is shown in Figs. 7. As the distance between the left and right instrument increased, the jerk of the planned trajectory increased. This indicates that the quality of the estimated trajectory deteriorates when the difference between the test condition and the demonstration condition is large. As for the effects of variance of the training dataset, the jerks of the planned trajectories were smaller the greater the variance.

These are natural results because our framework considers trajectory planning to be a regression problem. These results indicate that the demonstration should be performed under sufficiently various conditions.

The average computation time for planning an entire trajectory when using a 64-bit machine with an Intel Core i7-4600U CPU 2.1 GHz is shown in Fig. 8. The computation time was nearly proportional to the number of demonstrated trajectories given as inputs. Therefore, a trade-off exists between the computation time and quality of the planned trajectory. The demonstrated trajectory is often sub-optimal and contains unnecessary motions. If we have more demonstrations, unnecessary motions in the demonstrated trajectories are often

anceled out, and we can estimate better trajectories. However, increasing the number of demonstrations leads to increase of computational time for planning trajectories.

We performed trajectory planning under a dynamic condition to evaluate the proposed trajectory planning scheme. In this experiment, the developed system performed the DOUBLE LOOP task autonomously according to changing task conditions, and the spatial motion controller described in Section III-C was employed. Dataset1 in Fig. 4 was used for learning the DOUBLE LOOP task in this experiment. During the experiment, the left instrument was moved to disrupt the task execution while the system planned and updated the task trajectory under the dynamic conditions. The left instrument started moving approximately 4 seconds after the motion of the right instrument started.

Figure 9 shows the experimental procedure of this experiment. The system performed the task successfully despite the motion of the left instrument. The planned and executed trajectories in this experiment are visualized in Fig. 10. In Fig. 10(a), the displacement of the left instrument was  $[\Delta x, \Delta y, \Delta z] = [+12\text{mm}, -28\text{mm}, +36\text{mm}]$ . In Fig. 10(b), the displacement of the left instrument was  $[\Delta x, \Delta y, \Delta z] = [-23\text{mm}, +20\text{mm}, +44\text{mm}]$ . The trajectories were updated approximately every 50 ms. During the task, the system updated the trajectory 205 times in less than 11 seconds. The figure shows that the trajectories were updated according to the motion of the left instrument, and the right instrument tracked the planned trajectories in a stable manner. Fig. 11 shows the plots of the same trajectories as in Fig. 10 over time. From this figure, one can see that the trajectory converged to updated trajectories smoothly and quickly. This quick convergence to the updated trajectory shows the benefit of our tracking control scheme.

2) *Application Example: Knot Tying Task* : To demonstrate the capability of the proposed trajectory planning, we implemented autonomous knot tying in the developed system. The developed system learned two bimanual motions: making a loop with a thread, and then grabbing and pulling the thread. In the bimanual motion for looping, the relative position of the two arms was given as the task condition:

$$\xi_c = [x_r - x_l, y_r - y_l, z_r - z_l] \quad (35)$$

We performed the demonstration of the bimanual motion for making a loop nine times from the different initial positions. In the bimanual motion for grabbing and pulling the thread, the position of the stage was tracked by the stereo camera and given as the task condition:

$$\xi_c = [x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}] \quad (36)$$

We placed a color marker on the stage that was visually tracked by the KLT tracker implemented in OpenCV [46]. The measurement from the visual tracking was smoothed by using the Kalman filter. The motion for grabbing and pulling the thread was demonstrated nine times with the different stage position.

The procedure of the autonomous knot-tying is shown in Fig. 12. In this experiment, spatial motion control described in Section III-C was employed. The developed system

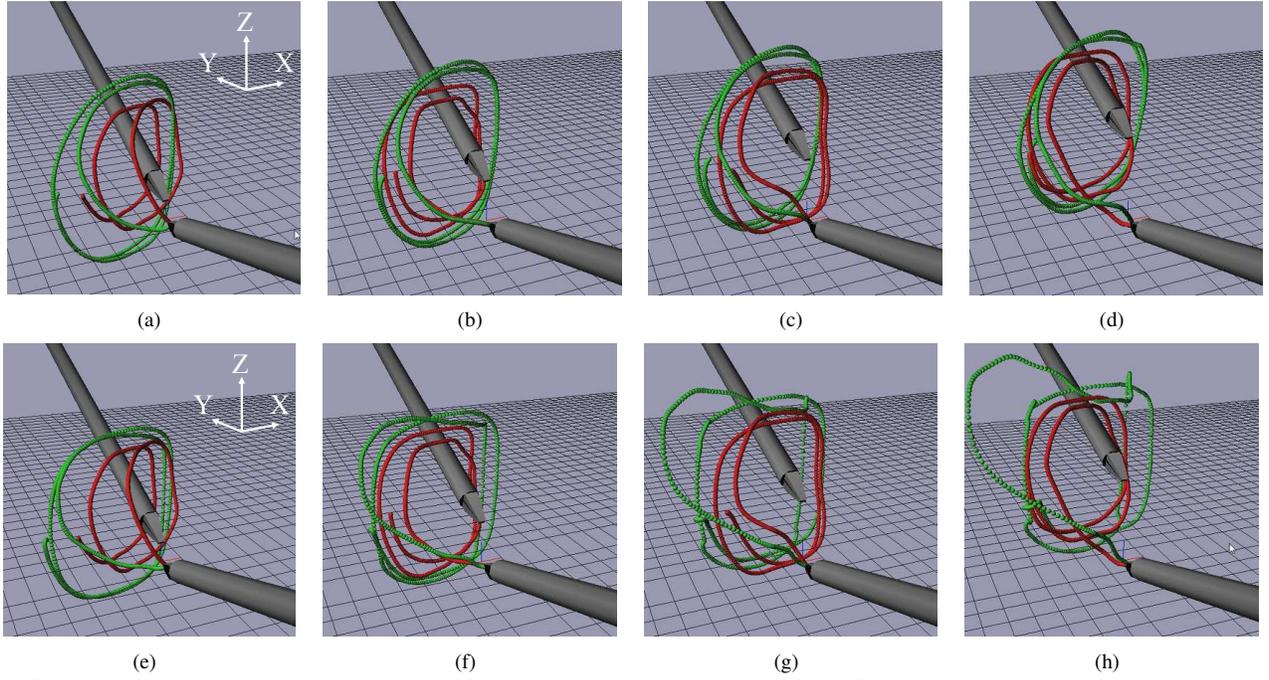


Fig. 6. Trajectory planned under test conditions. (a)–(d):Trajectories planned using dataset1. (e)–(h):Trajectories planned using dataset2. (a)and(e),  $\xi_c = [0, 0, 10]$ ; (b)and(f),  $\xi_c = [0, 0, 20]$ ; (c)and(g),  $\xi_c = [0, 0, 30]$ ; (d)and(h),  $\xi_c = [0, 0, 40]$ . Green dots represent the trajectory planned using the proposed scheme. Red dots represent the trajectory demonstrated by a human operator under the same condition.

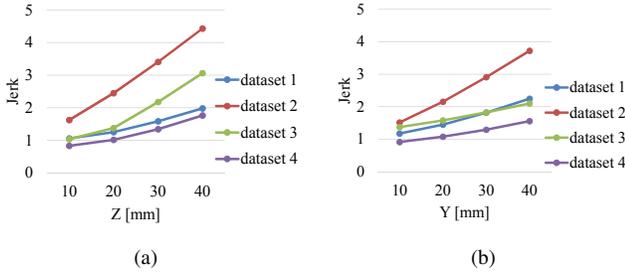


Fig. 7. Jerk of planned trajectory as a function of distance between the left and right instruments: (a) Z axis and (b) Y axis.

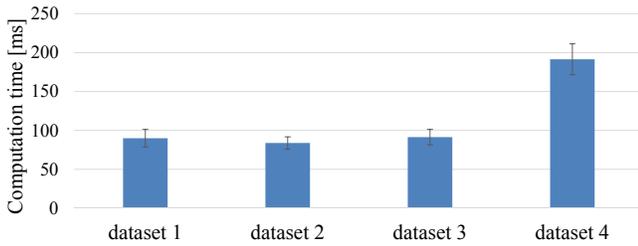


Fig. 8. Computation times for planning an entire trajectory to perform DOUBLE LOOP task, using dataset1, dataset2, dataset3, and dataset4.

autonomously performed knotting by executing the learned looping motion and grasping motion in sequence. In addition, by tracking the position of the object, the system planned and updated the trajectory in real time, and grabbed the thread even when the position of the thread was disturbed during the autonomous motions.

The trajectories planned and executed during the autonomous knotting is visualized in Fig. 13. As shown, the trajectories were planned and updated during the motion for grabbing the thread. The trajectory was updated 372 times in

about 20 seconds. This result indicates the capability of the proposed scheme that enables planning the task trajectory in a sufficiently short time under dynamic conditions.

## B. Evaluation of Online Trajectory Planning and Force Tracking Control

### 1) Performance of Proposed Force Tracking Control:

Here we evaluated the system in which the online trajectory planning and robust force tracking control schemes were integrated. The system learned the TOUCH-LINE task, which involves touching a line depicted on the contact object with a certain contact force (Fig. 14). This task was designed to emulate the motion of cutting an organ by using a surgical knife. The task requires the precision to touch along a straight line on the object and robustness to track the planned contact force trajectory under disturbances. This task is time- and space-dependent and difficult to describe as a single point-to-point task.

An operator demonstrated the TOUCH-LINE task nine times by controlling the slave manipulator from the master system. The force exerted on the slave manipulator was reflected on the master system, and the operator performed the task based on the haptic feedback from the master system. The force tracking control was implemented in the Z axis indicated in Fig. 14(a). The position of the contact object was varied for each demonstration.

In this experiment, the state of the system is defined as follows:

$$\xi_s = [x_r, y_r, z_r, F_{z,r}] \quad (37)$$

where  $[x_r, y_r, z_r]$  is the position of the right instruments, and  $F_{z,r}$  is the force exerted on the tip of the right instrument in the direction of the  $z$  axis. In this experiment, the force

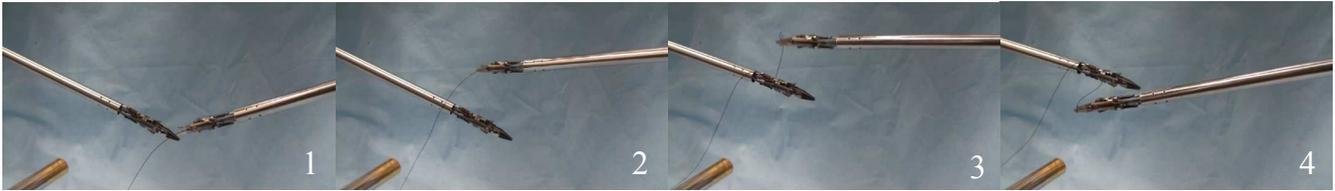


Fig. 9. Procedure of the DOUBLE LOOP task under dynamic conditions. The left instrument was moved upward to disrupt the task execution. The planned and executed trajectories are visualized in Fig. 10(a).

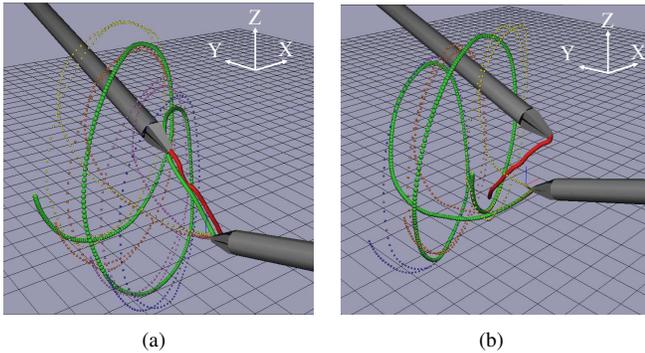


Fig. 10. Visualization of the planned and executed trajectories. The red and green dots represent motions executed by the left and right instruments, respectively. The blue, purple, orange, and yellow dots represent trajectories planned in this order during the experiment. The left instrument moved differently between (a) and (b).

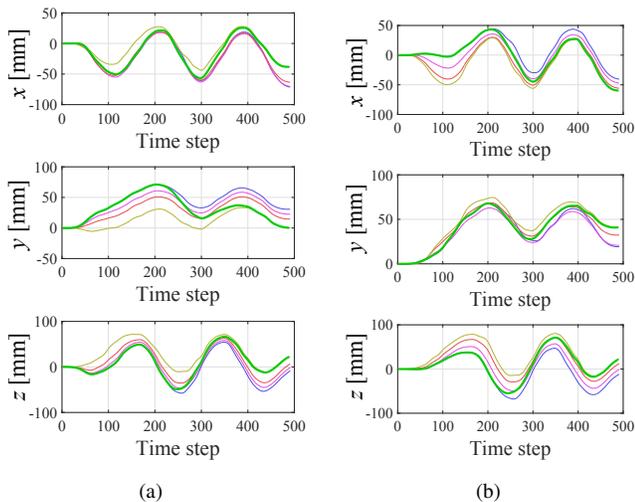


Fig. 11. Plots of trajectories over time. Green lines represent motions executed by the left and right instruments, respectively. Blue, purple, orange, and yellow lines represent trajectories planned in this order during the experiment. (a) and (b) correspond to Fig. 10(a) and (b).

tracking control described in Section III-D was employed in the direction of the  $z$  axis, and the spatial motion control described in Section III-C was employed in the direction of the  $x$  and  $y$  axis. The task condition is given as

$$\xi_c = [x_r(0), y_r(0), z_r(0)], \quad (38)$$

where  $[x_r(0), y_r(0), z_r(0)]$  represents the initial position of the tip of the right instrument. The demonstration was performed nine times using different initial positions. The demonstrated trajectories are visualized in Fig. 15. Specific conditions for the demonstration is found in Table V in the Appendix. The trajectories were generalized to new initial positions of the end

effector using the proposed trajectory planning method. The performance of the force tracking control was then examined.

To examine the robustness of the proposed force tracking scheme, we examined the system under three conditions: (a) the same as the demonstration; (b) a condition with a disturbance of the contact object stiffness (Fig. 14(b)); and (c) a condition with a disturbance of the contact object posture (Fig. 14(c)). We used a rubber sheet as the contact object, under which was placed a sponge in the demonstration. For the condition with disturbance of the contact object stiffness, we placed a steel plate between the rubber sheet and sponge (Fig. 14(b)). For the condition with disturbance of the contact object posture, we placed another sponge under the stage to tilt the contact surface (Fig. 14(c)). We performed the experiment with two systems: the system that learns and controls both spatial motion and contact force with the proposed scheme, and the system that learns only spatial motion with the proposed trajectory planning scheme.

To measure the contact force at the tip of the instrument, strain gauges were attached as shown in Fig. 16. In practice, it is not trivial to measure the contact force at the tip of a surgical instrument [47]. We used an experimental setup to allow for measuring the contact force in a stable manner.

Fig. 17 shows the results under the same condition as that of the demonstration, the condition with a disturbance of the contact object stiffness, and the condition with a disturbance of the contact object posture, respectively. In these experiments, the initial position of the end effector was set to  $\xi_c = [0, 0, 30]^T$ ; hence, the initial position of the end effector was different from the demonstration. Because the demonstrated trajectories were statistically modeled, the planned trajectories of the contact force were smooth despite the noisiness of the measured contact force in the demonstrations.

The system without force tracking control followed the planned spatial trajectory regardless of the disturbance. Under the same condition as the demonstration, the planned trajectory of the contact force was roughly achieved by the system that only learned the spatial motion (Fig. 17(a)). This indicates that it is reasonable to assume that (26) is satisfied. However, under conditions with a disturbance, the contact force largely deviated from the planned trajectory in the experiment with the system that learned only spatial motion (Fig. 17(b) and Fig. 17(c)).

In contrast, the proposed system with the force tracking control robustly tracked the planned trajectory of the contact force even under conditions with disturbances. Under conditions with disturbance, the contact force deviated from the planned contact force at the beginning of contact; however, the contact force converged to the planned trajectory in a stable

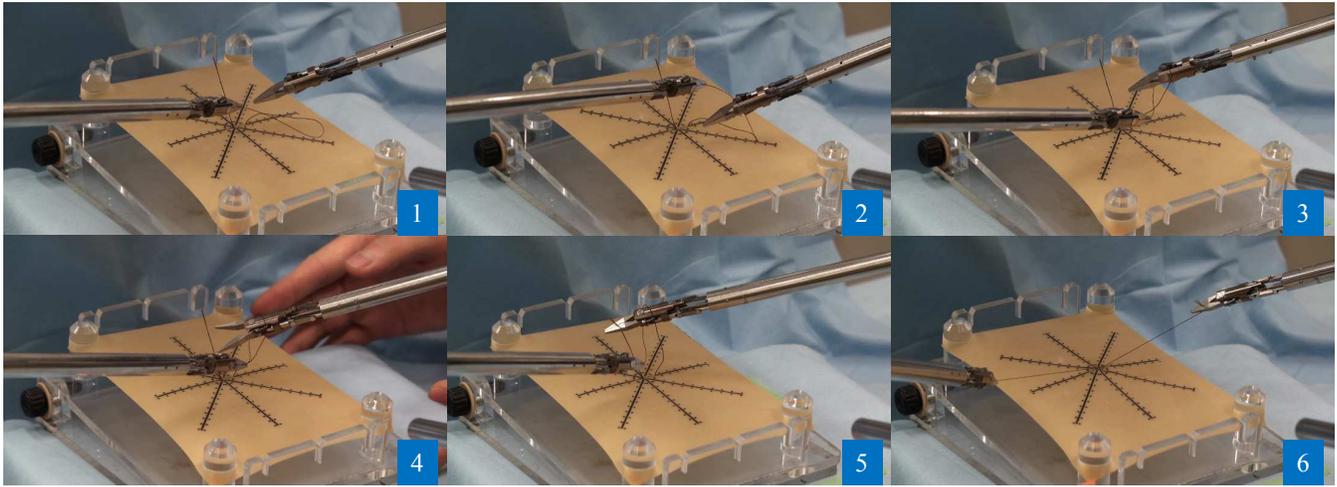


Fig. 12. Experimental procedure for autonomous knot tying. The position of the stage was moved manually to disrupt the task execution.

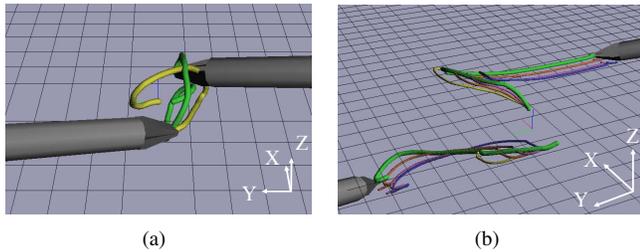


Fig. 13. Visualization of trajectory planned and executed during the autonomous knotting: (a) trajectories for making a loop and (b) trajectories for grabbing and pulling the thread. In (b), the green dots represent the executed trajectory. The Blue, orange, and yellow dots represents the trajectories planned in this order.

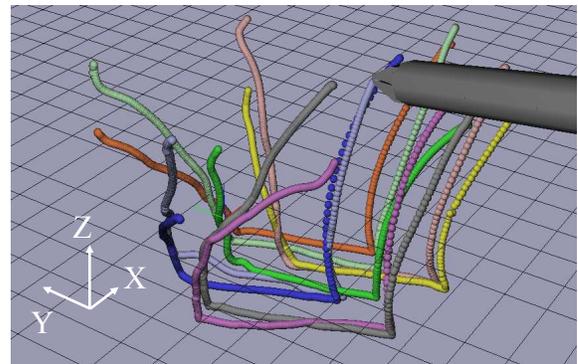
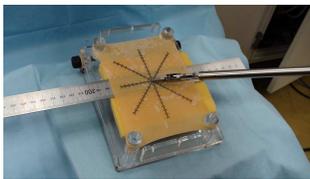


Fig. 15. Visualization of demonstrated trajectories for TOUCH-LINE task. Specific conditions are shown in Appendix.



(a)



(b)



(c)

Fig. 14. Setup for TOUCH-LINE task. The task involves pushing along the line indicated in (a); (b) shows the setup for experiment with a disturbance in the contact object stiffness; (c) shows the setup for experiment with a disturbance in the contact object posture.

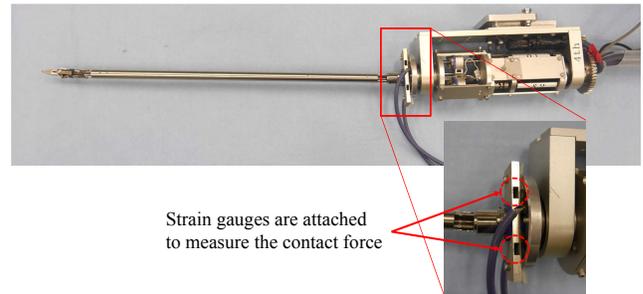


Fig. 16. Robotic instrument with force sensors. Strain gauges were attached to the base of the rod.

manner (Fig. 17(b) and Fig. 17(c)). Following convergence, the system robustly tracked the planned contact force even when the planned contact force began to decrease. These results show that the proposed force tracking control scheme enables robust tracking and planning of the contact force even under conditions with disturbances. In this experiment, the trajectory does not change the directions where the position control is used because the disturbance from the contact object does not affect the trajectory in the direction where the position control is used.

2) *Force Tracking Control with Online Trajectory Planning:* We performed online trajectory planning with force tracking control under dynamic conditions. In this experiment, we disturbed the position of the contact object during automatic execution of the TOUCH-LINE task.

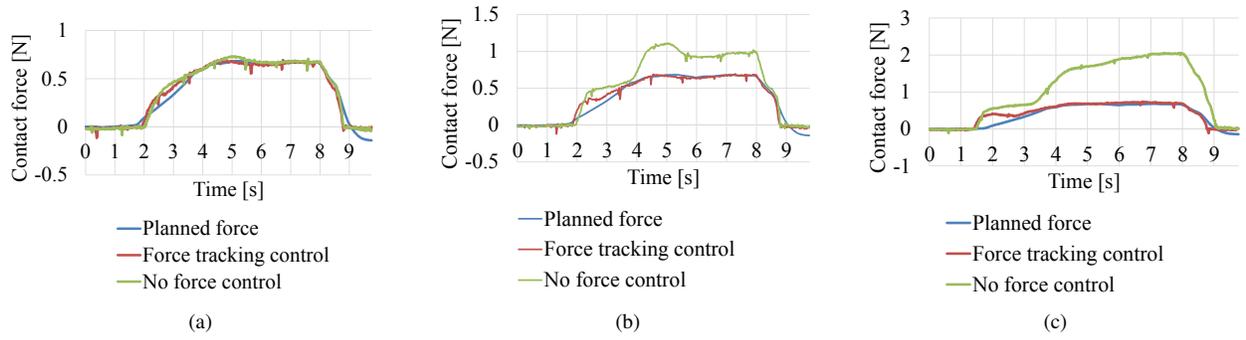


Fig. 17. Contact force in force tracking experiment under three conditions: (a)result under the same condition as the demonstration, (b)results under the condition with disturbance of the contact object stiffness, and (c)results under the disturbance of the contact object posture.

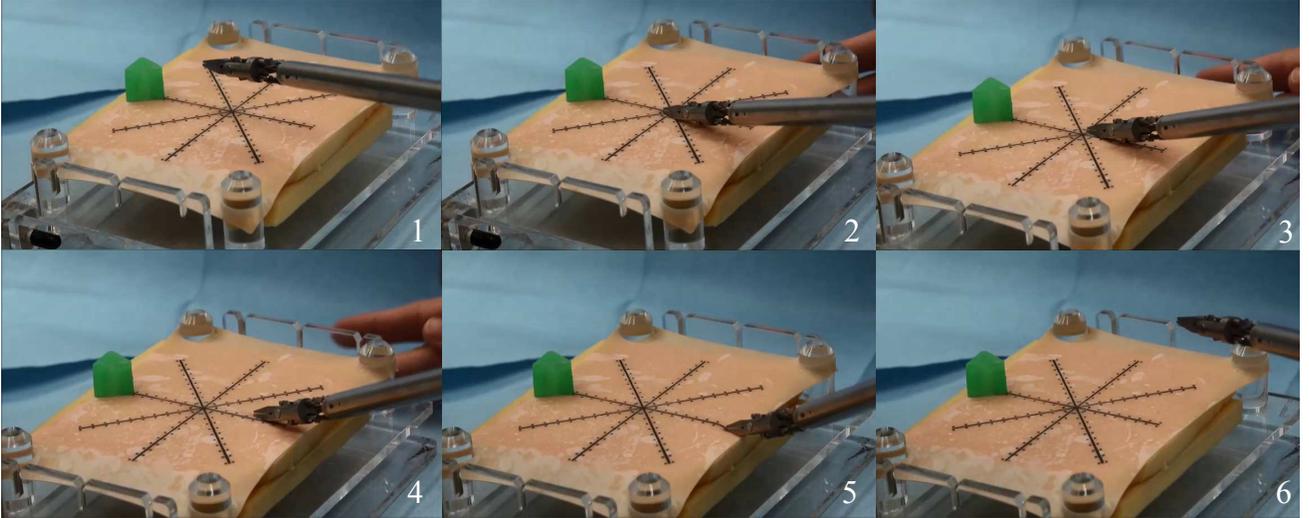


Fig. 18. Experimental procedure for the TOUCH LINE task to evaluate online trajectory planning with the proposed force tracking control.

In this experiment, the position of the contact object was tracked by a stereo camera with a KLT tracker implemented in OpenCV. The measurement from the visual tracking was smoothed by using Kalman filter. The state of the system was given as the same as the previous experiment:

$$\xi_s = [x_r, y_r, z_r, F_{z,r}]. \quad (39)$$

We used the task condition given as

$$\xi_c = [x_{cam}, y_{cam}, z_{cam}], \quad (40)$$

where  $[x_{cam}, y_{cam}, z_{cam}]$  represents the position of the contact object in the stereo camera coordinates. The transformation between the stereo camera coordinates and the robot coordinates was unknown. The demonstration was performed nine times with different positions of the contact object. The initial position of the end effector was fixed in this experiment.

The experimental procedure is shown in Fig. 18. The manipulator tracked the line on the contact object even though the contact object was moved during the task execution. The trajectory was updated 188 times in less than 11 seconds. This indicates that the task trajectory was appropriately generalized to the new conditions in real time.

The planned and executed trajectories during the experiment are visualized in Fig. 19. The trajectories were planned and updated approximately every 50 ms according to the movement of the contact object. The system updated the trajectory

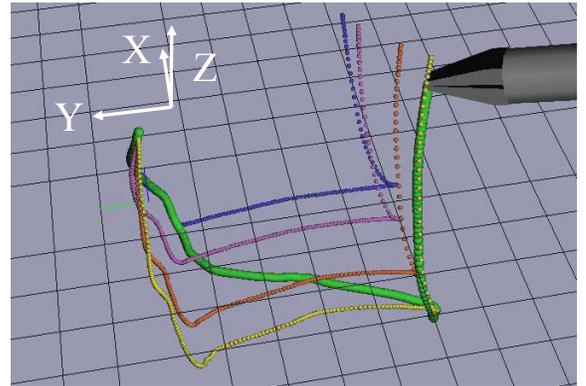


Fig. 19. Visualization of the planned trajectories and the executed trajectory during the experiment of online trajectory planning with force tracking control. The green dots represent trajectory executed by right instrument. The blue, orange, and yellow dots represent trajectories planned in this order during the experiment.

188 times in approximately 11 seconds. The trajectory of the contact force and the position of the manipulator are shown in Fig. 20. Although the contact force is disturbed by the motion of the contact object, the planned contact force was robustly achieved by the manipulator. This result indicates that the proposed system is applicable to the online trajectory planning and force tracking control under dynamic task conditions.

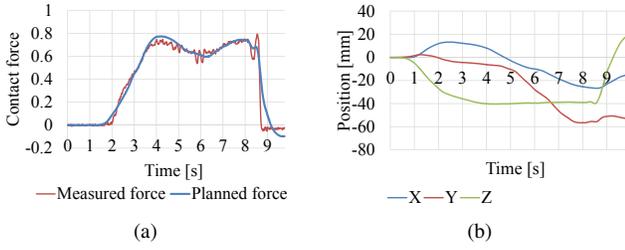


Fig. 20. Data recorded in the experiment when the contact object was moving: (a) Contact force and (b) the position of the slave manipulator.



Fig. 21. Experimental setup for the tightening-a-knot task.

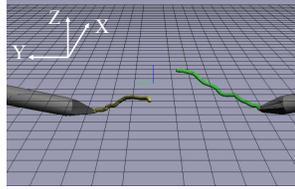


Fig. 22. Example of the trajectory demonstrated for the tightening-a-knot task.

3) *Application Example: Tightening a Knot:* To demonstrate the performance of the force control scheme, we implemented an autonomous motion for tightening a knot. We performed a motion for tightening a knot nine times from the different initial positions of two instruments (Fig. 21). In the demonstration, the position of the thread held by the instrument was fixed. In this experiment, the system learned the force in one direction as shown in Fig. 21. The state of the system was given as:

$$\xi_s = [x_r, y_r, z_r, x_l, y_l, z_l, F_{y,r}] \quad (41)$$

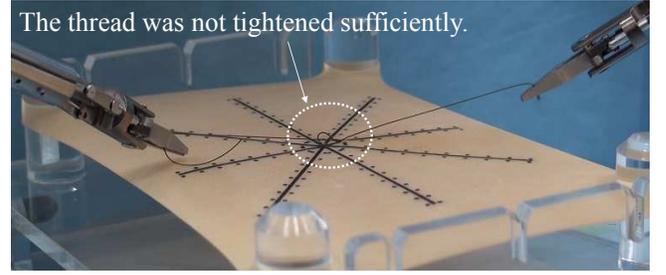
where  $F_{y,r}$  is the force exerted on the tip of the right instrument in the direction of the  $y$  axis. The initial positions of the tips of the right and left instruments are used as the task condition:

$$\xi_c = [x_r(0), y_r(0), z_r(0), x_l(0), y_l(0), z_l(0)]. \quad (42)$$

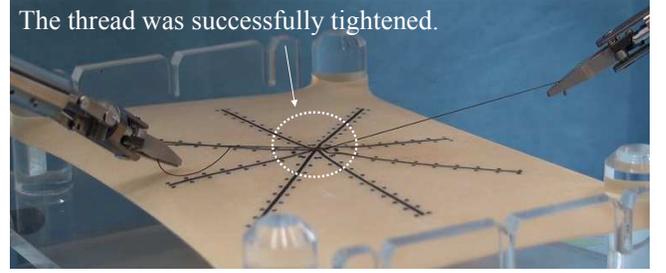
By using this representation, the system generalized the task trajectories to different initial positions of the right and left instruments. One of the demonstrations is visualized in Fig. 22. In this experiment, the force tracking control was employed in the direction of the  $y$  axis, and the spatial motion control was employed in the direction of the  $x$  and  $z$  axis. The conditions of the demonstrations are shown in the Appendix. In these experiments, the initial position of the end effector was set to  $\xi_c = [0, 0, 0, 0, 0, 0]^T$ .

To examine the robustness of the proposed scheme, we varied the length of the thread and performed autonomous tightening for several scenarios: (1) The thread was the same length as that of the demonstration; (2) The thread was 10mm shorter than that of the demonstration; and (3) The thread was 10mm longer than that of the demonstration.

The result of the experiment is shown in Figs. 23 and 24. As shown, the system using the proposed force tracking control apparently outperforms the system without force control. Despite the difference of the thread length, the proposed system tightened the knot appropriately. Figure 24 shows the force profile recorded in this experiment. The length of the thread



(a)



(b)

Fig. 23. Results of autonomous tightening under the condition where the thread was held to be longer than that of the demonstration: (a) Result without force control, and (b) result with the proposed force control.

significantly affected the force profile during the task. If force control is not implemented, the measured force was nearly zero at the end of the task when the thread is longer than that of the demonstration, and when the thread is shorter, the measured force largely exceeded the planned value at the end of the task.

In contrast, the force was appropriately controlled to be the desired value at the end of the task regardless of the length of the thread if the proposed force control method was implemented. This result indicates that the proposed scheme achieves robust tracking of the planned force profile even when experiencing disturbances.

## V. DISCUSSION

Our scheme enables online planning and robust tracking of spatial motion and contact force based on demonstrations. In addition, our force control scheme achieves robust tracking of the planned contact force under dynamic conditions. Although we updated the trajectories every 50-60 ms in the experiments, it might not be necessary to update the trajectory as often for simple tasks. However, we think that the faster update of the trajectory leads to more adaptive behaviors of the system. Our approach solves the trajectory planning problem as a regression problem. Although we use GPR to model the distribution of the demonstrated trajectories, our framework is not limited to specific regression methods. Sparse GPR and GMR can be also used in our framework, thereby reducing the computational costs [48], [49].

With regard to the use of GPR, the work by Calinon et al. described a method using GPR to retrieve the trajectory parameters from task conditions [18]. In order to generalize the motion to a new task condition, the method in [18] requires two steps: 1) estimate the parameters of GMMs with GPR,

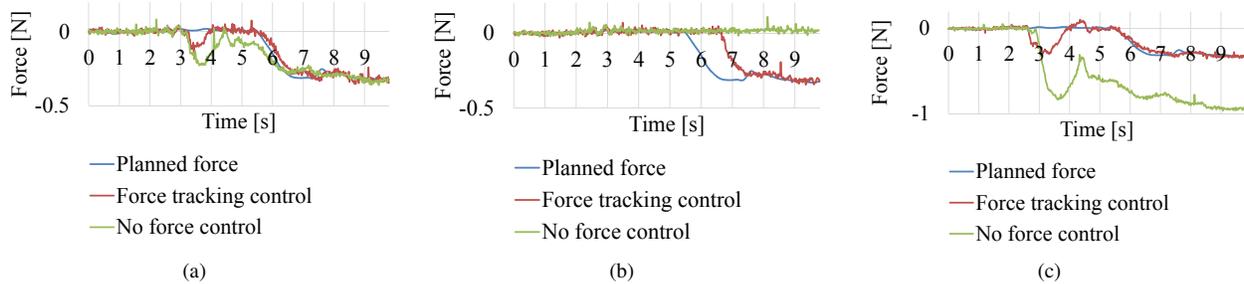


Fig. 24. Results of autonomous tightening: (a)Result under the condition where the thread was held to be the same length as that of the demonstration, (b)result under the condition where the thread was held to be shorter than that of the demonstration, and (c)result under the condition where the thread was held to be longer than that of the demonstration.

and 2) generate a trajectory with the GMMs. On the contrary, we use GPR to directly retrieve the state of the system from the task conditions. Since our approach directly estimates a trajectory from a given task condition, the computational cost will be less, which is suitable for online trajectory planning.

In the DOUBLE LOOP and TOUCH-LINE tasks, the relationship between task conditions and trajectories is nearly linear. For this reason, the trajectory distribution was successfully modeled from only nine demonstrated trajectories in the experiments. When the relationship between task conditions and trajectories is nonlinear, more demonstrations will be required. In addition, task conditions in our experiments comprised three or six dimensions. When the dimension of the task condition increases, the system will require more demonstrations, which in turn will lead to greater computational costs. For example, if the orientation of the object needs to be additionally taken into account, demonstrations with various object orientations are necessary.

In prior work in [12], a spatio-temporal perturbation is referred to as a perturbation that affects the length of a task, which results in changes in both a spatial trajectory and temporal dynamics. Pushing away a target object during a task is presented as an example of such a spatio-temporal perturbation in [12]. The experiment presented in Section IV-A shows the performance of our system under spatio-temporal perturbations. Moving the target instrument away from the controlled instrument during the DOUBLE LOOP task is a spatio-temporal perturbation that affects both the spatial and temporal dynamics of the movement. As shown in the experiment in Section IV-A, when a spatio-temporal perturbation was introduced, our system adapted the behavior without unnatural behaviors. In our framework, spatio-temporal perturbation is addressed using online trajectory planning and tracking control. By updating a spatial trajectory on the fly, our system plans natural spatial trajectory according to the change of the condition. Simultaneously, our tracking control adapts the velocity and tracks the planned trajectory in a stable manner. This adaptation of the velocity can be interpreted as the adaptation of the temporal dynamics to the updated trajectory. Therefore, our system stably adapts its behavior even when a spatio-temporal perturbation is introduced.

In the step where we align the trajectory in the time domain using DTW, the trajectory distribution is assumed to be a Gaussian distribution in (8). Therefore, if the distribution of the demonstrated trajectories does not follow this assumption,

it may be necessary to modify the method for aligning the trajectories in the time domain. When we compute the variance in (8), we assumed that every demonstration has equal relevance to the reference trajectory. However, one can use the variance that encapsulates the information about the task condition. One simple way to implement this condition-related variance is to set the variance as  $R^i \propto \exp(-\|\xi_c^i - \xi_c^{\text{test}}\|)$  for the given context  $\xi_c^{\text{test}}$  in (8). This extension needs to be investigated in future work.

In Section IV-B, the performance of the force tracking control is discussed. The results show that the proposed scheme can track the planned trajectory of the contact force robustly even under disturbances. The approaches described in [30], [31] enable explicit force control, but they require iterative learning to adapt the controllers for new situations. In contrast to these approaches, our approach enables robust control of the contact force without iterative learning. As shown in the tightening-a-knot task, small changes of the condition significantly affect the profile of the contact force, especially when working with a flexible and non-stretchable object such as a thread. Thus, the robustness shown in the experiments are very important to perform such tasks that require the force tracking control.

This study shows that the force tracking control with online trajectory planning is potentially feasible by using the proposed scheme. Additionally, it can be used for automation of surgical tasks such as tying a knot and cutting soft tissues. Although experiments in this paper emulated surgical tasks, the tested situation shown in this paper is still very simple compared with real surgery. Experiments in a more realistic situation should be performed in future work. One of the main challenges for applying our framework to actual surgery is sensing. Although our framework requires sensing the task condition in real time, it is challenging to track a surgical thread and measure the contact force at the tip of an instrument during actual robotic surgery.

Regarding the application to actual clinical use, full automation of surgical task is not realistic at this time because supervision by human operators is essential to ensure the safety. Several recent studies developed shared control frameworks for teleoperation [50]–[53]. The method described in this paper is potentially applicable to such frameworks. Although we applied our method to automation of robotic surgery, our method is also applicable to various applications. The results shown here will contribute to various fields, which are not

strictly limited to medical applications.

In our framework, the performance of the automated motions is dependent on the quality of the demonstrated motions. In future work, we will investigate reinforcement learning approaches, which would enable autonomous improvement of the performance of automated motions [54], [55].

## VI. CONCLUSION

We presented a framework of online trajectory planning and force tracking control for automation of surgical tasks. The proposed scheme enables the planning of trajectories for spatial motion and contact force in real time under dynamic conditions by leveraging demonstrations under various conditions. By learning the spatial motion and contact force simultaneously, the planned profile of the contact force can be tracked robustly and stably. The proposed scheme enables performing surgical tasks that require online trajectory planning and force tracking control, such as cutting soft tissue and tying knots. The robustness of the proposed force tracking control scheme was analytically described.

The performance of the trajectory planning and force control schemes was evaluated through experiments that emulate tying a knot and cutting soft tissues. In the experiments, the developed system planned and updated both spatial trajectory and force profiles under dynamic conditions. Additionally, the planned contact force was tracked robustly against changes in the stiffness and position of the contact object. Experimental results indicate the feasibility of surgical task automation under dynamic conditions using our framework.

## APPENDIX

Tables I–IV present the specific conditions of the demonstrations for DOUBLE LOOP task in Section IV-A. Table V presents the specific conditions of the demonstrations for TOUCH LINE task in Section IV-B1. Table VI presents the specific conditions of the demonstrations for the tightening-a-knot task in Section IV-B1.

## ACKNOWLEDGMENT

This work was supported by JSPS Grant-in-Aid for Scientific Research(S) Number 23226006 and Grant-in-Aid for JSPS Fellows Number 25-7106.

## REFERENCES

- [1] J. Nix, A. Smith, R. Kurpad, M. E. Nielsen, E. M. Wallen, and R. S. Pruthi, "Prospective randomized controlled trial of robotic versus open radical cystectomy for bladder cancer: Perioperative and pathologic results," *European Urology*, vol. 57(2), pp. 196–201, 2010.
- [2] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning in dynamic environments for surgical task automation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2014.
- [3] T. Osa, K. Harada, N. Sugita, and M. Mitsuishi, "Trajectory planning under different initial conditions for surgical task automation by learning from demonstration," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6507–6513.
- [4] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Y. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 2074–2081.

TABLE I  
CONDITIONS FOR DATASET 1

No.	$x_1$	$y_1$	$z_1$
1	0	0	0
2	20	20	20
3	-20	20	20
4	20	-20	20
5	-20	-20	20
6	20	20	-20
7	20	-20	-20
8	-20	20	-20
9	-20	-20	-20

TABLE II  
CONDITIONS FOR DATASET 2

No.	$x_1$	$y_1$	$z_1$
1	0	0	0
2	10	10	10
3	-10	10	10
4	10	-10	10
5	-10	-10	10
6	10	10	-10
7	10	-10	-10
8	-10	10	-10
9	-10	-10	-10

TABLE III  
CONDITIONS FOR DATASET 3

No.	$x_1$	$y_1$	$z_1$
1	0	0	0
2	20	20	20
3	-20	20	20
4	20	-20	20
5	-20	-20	20
6	20	20	0
7	20	-20	0
8	-20	20	0
9	-20	-20	0

TABLE IV  
CONDITIONS FOR DATASET 4

No.	$x_1$	$y_1$	$z_1$
1	0	0	0
2	20	20	20
3	-20	20	20
4	20	-20	20
5	-20	-20	20
6	20	20	0
7	20	-20	0
8	-20	20	0
9	-20	-20	0
10	20	20	-20
11	20	-20	-20
12	-20	20	-20
13	-20	-20	-20
14	10	10	10
15	-10	10	10
16	10	-10	10
17	-10	-10	10
18	10	10	-10
19	10	-10	-10
20	-10	10	-10
21	-10	-10	-10

TABLE V  
CONDITION FOR THE TRAINING DATASET FOR THE TOUCH LINE TASK IN SECTION IV-B1.

No.	$x_r(0)$	$y_r(0)$	$z_r(0)$
1	0	0	20
2	20	20	20
3	20	-20	20
4	-20	20	20
5	-20	-20	20
6	20	20	40
7	20	-20	40
8	-20	20	40
9	-20	-20	40

- [5] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 543–548.
- [6] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, "Adaptive control for human-robot skill transfer: Trajectory planning based on fluid dynamics," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1800–1807.
- [7] H. Mayer, I. Nagy, D. Burschka, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, "Automation of manual tasks for minimally invasive surgery," in *Proceedings of International Conference on Autonomic and Autonomous Systems (ICAS)*, 2008, pp. 260–265.
- [8] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin, "Stitching planning in laparoscopic surgery: Towards robot-assisted suturing," *International Journal of Robotics Research*, vol. 28, no. 10, pp. 1303–1321, 2009.
- [9] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*,

TABLE VI

CONDITION FOR THE TRAINING DATASET FOR THE TIGHTENING-A-KNOT TASK IN SECTION IV-B3.

No.	$x_1(0)$	$y_1(0)$	$z_1(0)$	$x_r(0)$	$y_r(0)$	$z_r(0)$
1	0.5	-0.5	0.5	0.5	0.5	-0.5
2	-14.2	2.0	-5.0	10.4	-8.3	4.5
3	8.8	2.1	6.5	-20.1	-7.8	-8.1
4	15.3	1.9	-12.0	-24.3	-7.3	6.0
5	-16.3	1.5	7.3	10.7	-8.2	-11.2
6	9.2	8.9	-3.9	5.2	-16.1	-4.8
7	7.9	3.8	17.5	7.5	-7.9	14.5
8	-4.2	1.1	-2.9	11.2	-9.5	8.276
9	-4.5	6.6	13.2	-5.4	-7.5	-3.7

2013, pp. 4111–4117.

- [10] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1202–1209.
- [11] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009.
- [12] S. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [13] S.-M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, pp. 433–454, 2012.
- [14] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, no. 4, pp. 361–379, 2012.
- [15] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Proceedings of the 16th International Symposium on Robotics Research (ISRR)*, 2013.
- [16] A. X. Lee, S. H. Huang, D. Hadfield-Menell, E. Tzeng, and P. Abbeel, "Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 4402–4407.
- [17] B. C. da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5239–5244.
- [18] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 610–616.
- [19] A. Parachos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2616–2624.
- [20] T. Osa, G. E. A. M., R. Stolk, R. Lioutikov, J. Peters, and G. Neumann, "Guiding trajectory optimization by demonstrated distributions," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 819–826, 2017.
- [21] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [22] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*, June 1984, pp. 304–313.
- [23] D. E. Whitney, "Historical perspective and state of the art in robot force control," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 1985, pp. 262–268.
- [24] D. Lawrence, "Impedance control stability properties in common implementations," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 1988, pp. 1185–1190.
- [25] H. Seraji, "Adaptive admittance control: an approach to explicit force control in compliant motion," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 1994, pp. 2705–2712.
- [26] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *The International Journal of Robotics Research*, vol. 16, no. 1, pp. 97–117, 1997.
- [27] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009, pp. 399–404.
- [28] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4326–4332.
- [29] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, vol. 7, no. 3, pp. 367–380, July 2014.
- [30] J. Kober, B. Mohler, and J. Peters, "Learning perceptual coupling for motor primitives," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 834–839.
- [31] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3368–3375.
- [32] A. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [33] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell, "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1024–1030.
- [34] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.
- [35] J. Arata, H. Takahashi, S. Yasunaka, K. Onda, K. Tanaka, N. Sugita, K. Tanoue, K. Konishi, S. Ieiri, Y. Fujino, Y. Ueda, H. Fujimoto, M. Mitsuishi, and M. Hashizume, "Impact of network time-delay and force feedback on tele-surgery," *International Journal for Computer Assisted Radiology and Surgery*, vol. 3, pp. 371–378, 2008.
- [36] M. Mitsuishi, M. Hashizume, P. Navicharern, Y. Fujino, K. Onda, S. Yasunaka, N. Sugita, J. Arata, H. Fujimoto, K. Tanimoto, K. Tanoue, S. Ieiri, K. Konishi, and Y. Ueda, "A telesurgery experiment between japan and thailand," in *Proceedings of International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2009.
- [37] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 554–561.
- [38] H. Sakoe and S. Chiba, "Dynamic programming algorithm for spoken word recognition," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, A. Waibel and K.-F. Lee, Eds., San Francisco, CA, USA, 1978, pp. 159–165.
- [39] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*. ACM, 2008, pp. 144–151.
- [40] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, Nov. 2010.
- [41] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.
- [42] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics Automation Magazine*, vol. 17, pp. 44–54, 2010.
- [43] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.
- [44] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [45] R. L. Williams, D. W. Repperger, J. M. Henry, and M. A. Murphy, "Naturally-transitioning rate-to-force control in free and constrained motion," *Journal of Dynamic Systems Measurement and Control*, vol. 121(3), pp. 425–432, 1999.
- [46] [Online]. Available: <http://opencv.org/>
- [47] A. L. Trejos, R. V. Patel, and M. D. Naish, "Force sensing and its application in minimally invasive surgery and therapy: A survey," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 224, pp. 1435–1454, 2010.
- [48] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2005, pp. 1257–1264.
- [49] L. Weruaga and J. Via, "Sparse multivariate gaussian mixture regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1098–1108, 2015.
- [50] K. Shamaei, Y. Che, A. Murali, S. Sen, S. Patil, K. Goldberg, and A. M. Okamura, "A paced shared-control teleoperated architecture for

supervised automation of multilateral surgical tasks,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1434–1439.

- [51] N. Padoy and G. Hager, “Human-machine collaborative surgery using learned models,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5285–5292.
- [52] K. A. Nichols and A. M. Okamura, “A framework for multilateral manipulation in surgical tasks,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 68–77, Jan 2016.
- [53] F. B. Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. Giordano, “A learning-based shared control architecture for interactive task execution,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [54] M. P. Deisenroth, G. Neumann, and J. Peters, “A survey on policy search for robotics,” *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.
- [55] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, 2013.



**Takayuki Osa** received his Master’s degree and doctoral degree in Mechanical Engineering from the University of Tokyo in 2010 and 2015, respectively. He worked for Terumo Corporation, Japan, from 2010 to 2012. Currently, he is working as a postdoctoral fellow at Intelligent Autonomous Systems institute, Technische Universität Darmstadt, Germany. His research interests include machine learning for object manipulation, teleoperation, and intelligent surgical systems.



**Naohiko Sugita** received a Master’s degree in Mechanical Engineering from the University of Tokyo in 1996. He worked for NEC, Japan, from 1996 to 2003. He became a research associate in 2003 and an associate professor in 2007 at the University of Tokyo. He received a doctoral degree in Mechanical Engineering in 2005 from the University of Tokyo. His research interests include machining of biomaterials, robot-assisted surgical systems, and intelligent manufacturing systems.



**Mamoru Mitsuishi** (M93) received his Master’s and D.E. degrees in Mechanical Engineering from the University of Tokyo, Tokyo, Japan, in 1983 and 1986, respectively. In 1986, he was a lecturer at the University of Tokyo, where he also held the post of an associate professor in 1989; he has been a professor there since 1999. From 1987 to 1988, he was a visiting researcher at the Fraunhofer Institute for Production Technique and Automation, Germany. His research interests include computer integrated surgical systems and manufacturing systems.