A Cooperative Control Framework For Haptic Guidance Of Bimanual Surgical Tasks Based On Learning From Demonstration

Maura Power, Hedyeh Rafii-Tari, Christos Bergeles, Valentina Vitiello, Guang-Zhong Yang *Fellow, IEEE*

***Abstract*—Whilst current minimally invasive surgical robots offer many advantages to the surgeon, most of them are still controlled using the traditional master-slave approach, without fully exploiting the complementary strengths of both the human user and the robot. This paper proposes a framework that provides a cooperative control approach to human-robot interaction. Typical teleoperation is enhanced by incorporating haptic guidance-based feedback for surgical tasks, which are demonstrated to and learned by the robot. Safety in the surgical scene is maintained during reproduction of the learned tasks by including the surgeon in the guided execution of the learned task at all times. Continuous Hidden Markov Models are used for task learning, real-time learned task recognition and generating setpoint trajectories for haptic guidance. Two different surgical training tasks were demonstrated and encoded by the system, and the framework was evaluated using the Raven II surgical robot research platform. The results indicate an improvement in user task performance with the haptic guidance in comparison to unguided teleoperation.**

I. Introduction

Over the last few years, the number of surgeries performed with the aid of surgical robotics has increased dramatically. Despite the considerable success of robotic surgical systems, the question of their overall superiority with respect to non-robotic techniques is still debated, [1]. Advantages include motion scaling, tremor reduction and highly dexterous instruments. However, these advantages come with trade-offs such as large footprints in the operating theatre, high cost and a lack of haptic feedback. Furthermore, there is a need to incorporate more intelligence into these robotic systems through shared control by combining the high-level decision making of the surgeon with the super-human dexterity provided by the robot.

Robotic Learning from Demonstration (LfD) is a promising technique which incorporates theory from robotics, statistics, computer science, machine learning, psychology, control and human-robot interaction. There are three main stages in a typical LfD framework; task demonstration (usually provided by a human user), task encoding (generally carried out using machine learning technique) and, finally, reproduction of the learned task (executed by the robot). Thus far, there exists a range of machine learning techniques, from trajectory-level to symbolic-level learning, batch learning to on-line learning for this purpose. Refer to [2] for a comprehensive review of LfD. LfD is already used in a wide range of applications including skills analysis, learned task execution and skills transfer. Furthermore, LfD does not require any conventional programming skills or knowledge of robotic kinematics from the task demonstrator. This is particulary suited to repetitive tasks during minimally invasive surgery (MIS).

In terms of general robotic control, LfD is an attractive alternative to teaching a robot a complex task. The Continuous Hidden Markov Model (CHMM), which is a trajectory-level machine learning technique, has been used to autonomously complete a learned sub-task upon recognised completion of the previous sub-task, [3]. Another trajectory-level learning technique based on Gaussian Mixture Models (GMMs) has been used in conjunction with Dynamic Time Warping (DTW) to encode and automate various surgical tasks, [4]. A combination of trajectory- and symbolic-level encoding has been explored, [5]. This symbolic-level approach is based on a concept adapted from psychology known as 'scaffolding' and is realised by providing a high-level description of the task prior to the demonstration of a surgical knot tying procedure. A combination of iterative learning control and an LQR controller is used to increase the speed of execution up to 10 times that of the original demonstrated executions of a knot tying task [6].

It can be argued, however, that the prospect of surgeons handing over full control to autonomous surgical task execution in the near future is unlikely. Cooperative, or shared, control between robot and surgeon offers a complementary compromise between master-slave operation (complete user control) and full autonomy (complete robot control). Cooperative Control (CC) is a paradigm whereby the workload of a task is distributed between a human and a robot. One such example applied to a medical scenario utilises a hybrid control loop combining the force input of the human operator and the force sensor readings of the robot to adjust the cutting rate during a bone resection operation for safer outcomes, [7]. Active Constraints (also known as 'virtual fixtures') is another method of CC which renders forces based, typically, on preoperative scans of the area of interest. A live-stream of point cloud data from an RGB-D camera is used to construct a safe-zone around a beating heart, [8].

In the work proposed in the paper, a CC framework is implemented by combining LfD with the concept of safe-zones borrowed from active constraints. Learned task sequence recognition is implemented for automated engagement of

Maura Power, Hedyeh Rafii-Tari, Christos Bergeles, Valentina Vitiello and Guang-Zhong Yang are with the Hamlyn Centre for Robotic Surgery, Imperial College London, SW7 2AZ, London, UK (email:m.power13@imperial.ac.uk)

haptic guidance. The proposed system brings together a number of benefits during task execution under guidance; 1) *consistency* - the order of motions is encoded so that guided execution will be constrained within the boundaries of sequential steps of the learned task, 2) *efficiency* and *economy of motion* - short movements without superfluous motions, and 3) *safety* - the surgeon remains in the loop, and is therefore in ultimate control of the robot, at all times. The novel contributions of this work include an algorithm for online equidistant trajectory resampling, a generalised protocol for task encoding using CHMMs, a windowed approach to CHMM evaluation for real-time sequence recognition, and, finally, the first application of haptic guidance in a CC system to the Raven II robot.

The paper is organised as follows: Section II describes the methodology behind the LfD and CC framework, from task demonstration to encoding to task execution with haptic guidance, Section III outlines the experimental setup for the two tasks, Section IV presents the results of the user studies and, finally, Section V concludes the findings of this work and where its potential future work lies.

II. Methods and Materials

The proposed framework has two phases, with CHMMs playing a key role in both. 1) The learning phase - the LfD stage of the framework employs CHMMs to encode the surgical tasks using multiple demonstrations. This phase is used only when a new task needs to be encoded. 2) The task execution phase - this encapsulates the CC element of the framework. This can be further sub-divided into two stages; the task-searching stage and the haptic guidance stage. CHMMs are used here for evaluation of the live trajectories versus learned models. The learned models are then used to guide the user based on force-feedback.

In order to set up a common dialogue between the surgical task demonstrator, the LfD framework and the CC framework used during haptic guidance, it is first necessary to define how a task is structured in the context of this system. Two key definitions are used to describe the surgical tasks, with two levels of abstractions. At the lowest level is the *primitive movement* (PM) of the task. The primitive movements are elementary motions, which form the most basic units of a surgical task. A PM is encoded as a single CHMM. At the highest level of abstraction is the surgical *task*, which comprises a short series of PMs. A task is stored as a sequence of CHMMs.

In the following sections the LfD and CC elements of the framework are described using an example of a task which represents a needle-passing sub-task. A peg-transfer task is also introduced later as a second example of an encoded task. In Section III, the choice of these two tasks is discussed in more detail. The framework was initially developed and tested using a surgical robot simulator, before integrating the framework into the a real robotic platform. All results presented in this section are taken from the robot simulator.

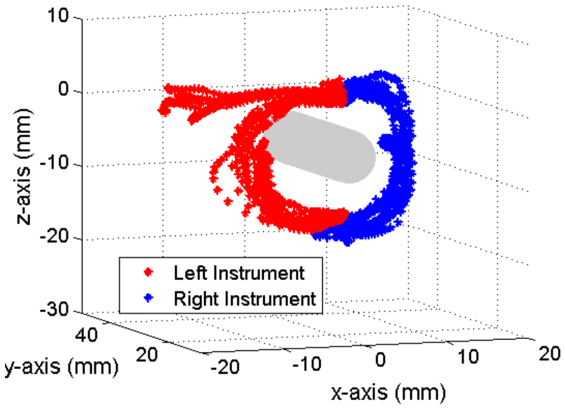


Fig. 1: Raw demonstration data of needle-passing sub-task

A. Task Demonstration

In this paper we identified two goals when designing the LfD framework; 1) generality of task encoding, and 2) automated encoding. To achieve these two goals, a simple protocol was established for the task demonstration stage. When providing example task executions the demonstrator should ensure smooth movements with minimal jerk or hesitation. The movements should also consist of simple line or arc movements with a brief pause in between to aid task segmentation. Segmentation of the trajectories is performed based on velocity and path-length thresholds. A segment is defined as any continuous sequence of end-effector data points, which satisfy the following two criteria (to ignore very small or very short movements); the velocity remains above a low speed threshold and the path length travelled is above a low distance threshold.

More elaborate movements are decomposed into shorter motions during the demonstration phase. These will be stitched together during encoding. The main purpose of defining how a task is structured and how it should be demonstrated is to accelerate learning and negate the need for manual feature extraction. This approach does not require the framework to have an intrinsic knowledge of specific surgical tasks, and therefore is widely applicable to any task demonstrated in the appropriate manner.

The above demonstration rules can cause the demonstrator to perform the task at a reduced speed. This will also ensure more well-defined PMs of the task for the encoding step. Moreover, the speed of the demonstrations has no implication on the speed of the guided task execution. The framework is designed to provide positional guidance only and, hence, it was a design choice for our framework to omit time-based guidance. The robot guides the user through the desired trajectory but the surgeon has full control over the pace of task execution.

Several demonstrations of a needle-passing sub-task provided using the robot simulator are plotted in Fig. 1. This task consists of five distinct movement. The five PMs, derived using the velocity segmentation procedure described above,

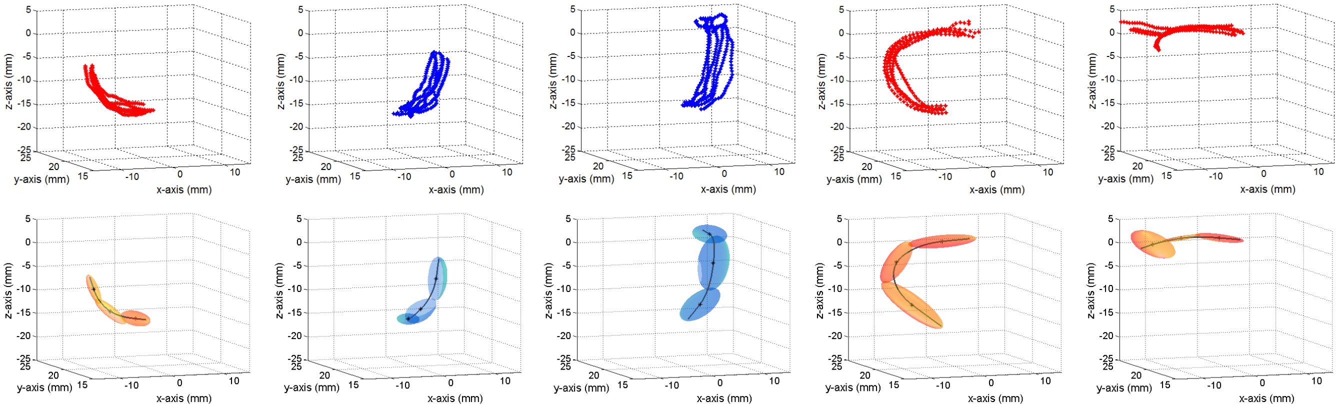


Fig. 2: Needle-passing task demonstrations (top row), encoded models and regressed ideal trajectories (bottom row) - 3 left (red) and 2 right (blue) PMs

of this task are shown separately in the top row of Fig. 2.

B. Continuous Hidden Markov Models

CHMMs are a doubly-stochastic means of encoding time-dependant or sequential data, such as trajectory information. A CHMM is represented by;

$$\lambda = (\pi, A, \mu, \Sigma) \tag{1}$$

In this framework CHMMs are used to encode end-effector Cartesian position information.

1) Encoding with CHMMs: Encoding multiple tasks demonstrations is carried out using the standard Baum-Welsh (BW) algorithm for model training. Given a set of observations (demonstrations), $\mathbf{O}$, the BW algorithm maximises (2)

$$P(\mathbf{O}|\lambda) \tag{2}$$

to train the CHMM, λ.

In applications where the encoded trajectory is not segmented, it is necessary to choose the optimal number of states for the CHMM, λ. In this framework, however, because all of the encoded PMs consist of short lines or arcs, the number of states for each primitive movements is set to three states. Choosing a set number of states allows fairer comparison between encoded PMs in the task execution and sequence recognition phase of the framework. The velocity-segmented trajectories are resampled to be distributed equally in space prior to encoding, and the motivation behind this step is explained in the following section. Finally, in order to to regress a trajectory from the CHMMs, a pseudo-time dimension must also be included prior to task encoding. The CHMMs, therefore, consist of four dimensions. The pseudo-time element is required only for the Cartesian regression step to provide a sequential dimension, and is not used when evaluating the real-time trajectories. This sequential fourth dimension, $\mathbf{t}$, for a trajectory of length L is calculated according to (3).

$$\mathbf{t} = (t_0, t_1 ... t_l ... t_{L-1}) \quad where \;\; t_l = \frac{1.0}{L-1} \tag{3}$$

The encoded PMs of the demonstrated needle-passing task from Fig. 1 are shown (sequentially from left to right) in the bottom row of Fig 2.

2) Evaluation with CHMMs: Evaluation, in the context of CHMMs, is the application of the Forward (or Forward-Backward) algorithm to determine the degree of match between a CHMM and a trajectory observation. The output of the Forward algorithm is a probability value. To avoid data underflow, the log likelihood of the probability value is usually considered.

The log likelihood of observing a trajectory, $\mathbf{T}$, for a given CHMM, λ is;

$$log(P(\mathbf{T}|\lambda)) \tag{4}$$

A novel method was developed for a real-time evaluation of trajectories based on a windowing approach. Short 'windows' of the live datastream are continuously compared with the encoded PMs of the learned tasks. The Forward algorithm is of $O(N^2T)$ complexity (where N is the number of states and T is the number of observations), but since the number of states is small and the number of observation datatpoints is also small for the windowing method, this number of calculations is relatively low and can be carried out at every data acquisition loop. This approach offers an alternative method of evaluation to comparing full trajectory segments with the encoded PM CHMMs. The application of both full and windowed live trajectory information for learned task sequence recognition is described in part E of this section.

C. Online Equidistant Resampling of Trajectories

An issue, which was encountered during the encoding and evaluation phase, was unequal distribution of data points as a function of demonstration velocity. This occurs as a result of data collection at a constant frequency, selected to be 20Hz. Portions of the trajectories where the end-effector moved quickly resulted in sparse data point distribution, and a dense clustering of data points resulted in segments of the trajectory where the end-effector velocity was relatively slow. This was incompatible with the desire to encode only sequential-based (and not explicitly time-based) information. Therefore, an additional preprocessing step is included to remove unwanted time-related artefacts from the trajectories by resampling the constant frequency data to be equidistant in 3D space.

The constant frequency trajectory is evaluated one line segment $\mathbf{L}_t$ at a time, where t is the time index for a given trajectory data point. A line segment is described by the current and previous Cartesian positions of the end-effector;

$$\mathbf{L}_t = (\mathbf{T}_{t-1},\ \ \mathbf{T}_t) \tag{5}$$

The aim of this algorithm is to resample a trajectory, $\mathbf{T}_{constant\ frequency}$, to a trajectory which is consists of data points that are spaced evenly along the direction of motion, $\mathbf{T}_{equidistant}$. The equidistance spacing is set as R. The psuedo-code the procedure is outlined in Algorithm 1.

```
Data: Set current resampled data point, $\mathbf{T}_n$, as first
      trajectory data point of original, non-equidistant
      trajectory, $\mathbf{T}_{constant\ frequency}$
while Current resampled, $\mathbf{T}_n$ data point is on line
segment $\mathbf{L}_t$ do
    Find the point of intersection $P$ between current
    line segment $\mathbf{L}_t$ and sphere centred on current data
    point $\mathbf{T}_n$ with radius $R$;
    if Intersection occurs then
        Add point of intersection to resampled
        trajectory, $\mathbf{T}_{equidistant}$. Reset current data point
        $\mathbf{T}_n$ as point of intersection $P$.
    else
        Stop resampling on current line segment.
    end
end
```

Algorithm 1: Online resampling of three-dimensional trajectory to equidistant points in space

This algorithm is applied separately to the real-time end-effector positions of both the right and the left instruments when the velocity is above a threshold (i.e. when the instrument is moving). The intersection between a line segment and a sphere is calculated using the standard computer graphics technique know as 'ray-sphere intersection'.

D. Task Encoding

An arbitrary number of tasks can be stored in memory. A 1D matrix stores each of the primitive movements for each of the K tasks. Each row of the task matrix contains a single task's associated PMs, and a PM is associated either with the left or right instrument.

$$\boldsymbol{\lambda}_{Task,k} = (\lambda_{PM,k1}, \lambda_{PM,k2}, ..., \lambda_{PM,kM}) \tag{6}$$

where

$$\boldsymbol{\lambda}_{Task,k} \in \boldsymbol{\lambda}_{Task,K} \qquad 1 \leq k \leq K \tag{7}$$

E. Sequence Recognition

A combination of evaluating full velocity-segmented movements and continuous evaluation of a short, buffered windows of live trajectory information is used to recognise when the user initiates the beginning of a learned task. The purpose of the sequence recognition is to smoothly engage haptic guidance when the user starts and the robot identifies a task that has been encoded by the framework.

In order to demonstrate the sequence recognition element of this framework, the second trained task model is introduced to illustrate the ability to distinguish when and which task has is being executed. The CHMMs of the needle-passing and peg-transfer tasks are plotted in Fig. 3. A trajectory of a reproduction of the peg-transfer task is also overlaid onto both models. The needle-passing task, as previously shown, consists of five PMs. The peg-transfer task is made up of four PMs; the first two from the left instrument, the second two from the right instrument. The aim is to identify, in real-time, that the peg-transfer task is being performed.

From a list of K stored tasks, using the same notation as the previous section, the most likely task is found by identifying:

1) a full, velocity segmented trajectory which has a sufficiently high degree of match using the standard Forward algorithm with the first PM of a learned task

$$k_{most\ likely} = \max_K(P(\mathbf{T}_{full\ segment} \mid (\boldsymbol{\lambda}_{PM,k1}))) \tag{8}$$

provided

$$P_{k,most\ likely} > P_{full\ segment\ threshold} \tag{9}$$

Fig. 4 shows that the first, full velocity segmented movement is most similar to the first PM of the encoded peg-transfer task, and is also above the probability threshold. Resampled, equidistant trajectories with a set number of datapoints are used for evaluation so that a fixed threshold can be chosen.

2) a consecutive, partial velocity-segmented trajectory which has a sufficiently high degree of match using the windowed approach of comparing short segments of live trajectory data using the Forward algorithm

$$k_{most\ likely} = \max_K(\frac{1}{Q}\sum_{t=1}^{Q} P(\mathbf{T}_{windowed,i} \mid \boldsymbol{\lambda}_{PM,k2})) \tag{10}$$

where Q is the number of data points collected until a threshold distance is travelled, and provided

$$P_{k,most\ likely} > P_{windowed\ segment\ threshold} \tag{11}$$

The window size used in the framework is two data points. Fig. 5 shows that the next movement matches the second PM of the peg-transfer task. At this stage haptic guidance can be engaged for the peg-transfer.

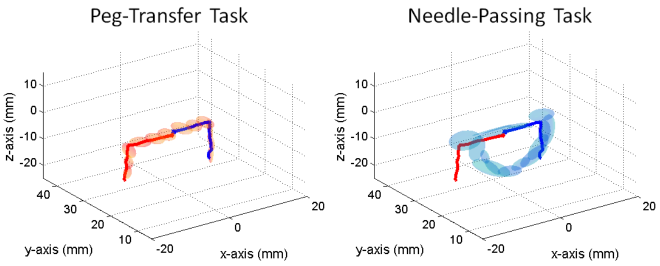


Fig. 3: Reproduction of a peg-transfer task trajectory overlaid onto encoded peg-transfer and needle-passing models

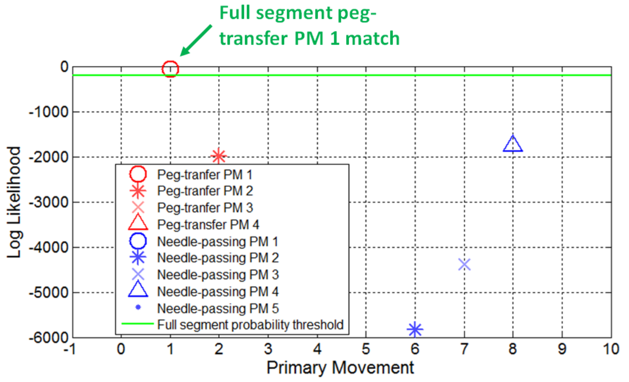


Fig. 4: Identification of first PM of peg-transfer task from velocity-segmented left instrument trajectory

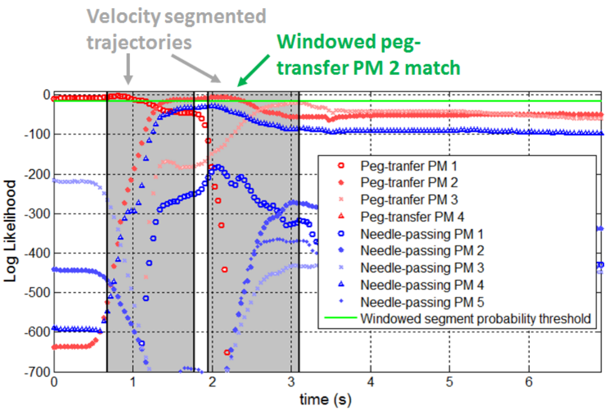


Fig. 5: Identification of second PM of the peg transfer task from windowed left instrument trajectory

F. Task Execution

The framework has three primary states; 1) *search* - compares velocity-segmented trajectories with the first PM of each trained task. 2) *identify task* - the first PM of a task is identified; the consecutive movement must match the next PM of this task in order to engage haptic guidance. 3) *haptic guidance* - if the next partial segment matches the expected PM (determined using the windowing technique) then haptic guidance is engaged midway through this PM and for the remainder of the task. Upon completion of the task, haptic guidance is disengaged. The cooperative control loop incorporates the learned model as the setpoint, the real-time trajectories readings of the robot as the loop input and the output of the control loop is used to provide haptic guidance to the user via the haptic devices. The haptic devices serve as both an input and output device, thus providing two-way communication between the user and robotic CC framework.

The ideal setpoint trajectory $\mathbf{T}_{sp}$ of each PM is derived using Gaussian Mixture Regression (GMR) using its encoded CHMM. The regressed trajectories corresponding to the PMs of the needle-passing task are plotted in grey on the bottom row of Fig. 2. The current end-effector setpoint $\mathbf{T}_{sp_{closest}}$ (a single Cartesian coordinate for each instrument) is found by calculating the closest single data point on the setpoint (sp) trajectory to the current end-effector position $\mathbf{P}_{ee}$;

$$\mathbf{T}_{sp_{closest}} = \mathbf{T}_{sp,n_{min}} \tag{12}$$

where

$$n_{min} = \min_{n} |\mathbf{T}_{sp,n} - \mathbf{P}_{ee}| \quad 1 \leq n \leq N \tag{13}$$

and

$$|\mathbf{T}_n - \mathbf{P}_{ee}| = \sqrt{(x_{ee} - x_{sp,n})^2 + (y_{ee} - y_{sp,n})^2 + (z_{ee} - z_{sp,n})^2} \tag{14}$$

A proportional control law is used to guide the user along the desired setpoint trajectory. The proportional gain K_p, which was kept constant for all user studies, was chosen empirically to be high enough to give the user a good sense of the trajectory they should follow, but not strong enough to completely prevent them from straying outside the desired trajectory, if necessary.

III. Experimental Setup

To evaluate the framework, eight subjects were recruited to carry out two surgical sub-tasks with and without haptic guidance. The experimental setup and tasks used are described in the following sections.

A. Robot Simulator and Raven II Robot Platform

A surgical robot simulator was used for the initial *in silico* development of the learning and cooperative control framework. This simulator was created in-house at the Hamlyn Centre. The Raven II robot (Applied Dexterity, Seattle, WA) is an open-source surgical robot research platform which was developed at University of Washington. The robot consists of a two 7 degree of freedom arms. Each arms includes detachable instrument which ends in a gripper. Both the surgical robot simulator and the Raven II robot are interfaced with using two Geomagic Touch (Geomagic, Morrisville, NC) haptic devices.

B. Surgical Tasks

Two tasks were chosen for evaluation of the framework using the Raven II robot. The tasks were sub-tasks of a peg-transfer task and a needle-passing task, which are the same as those encoded in the simulator. Both of these tasks are based on two surgical skills procedures from the Fundamental of Laparoscopic Surgery (FLS) program [9]. The FLS program is a well-established method of assessing the technical performance and hand-eye coordination of trainee surgeons.

The peg transfer task is carried out using a standard FLS peg transfer board. The needle passing task is a modified version of the FLS intracorporeal suturing task; instead of requiring the user to pierce the penrose drain, the user was asked to pass a suture needle through a hole in a piece of card for minimal resistance. Both tasks, trained with the simulator and the Raven II, are depicted in Fig. 6.

These two tasks were chosen as they both involve bimanual interactions between the tools and an additional object. In this respect they are similar, but the peg transfer task is an easier

The kinematics of the simulator will be made available online

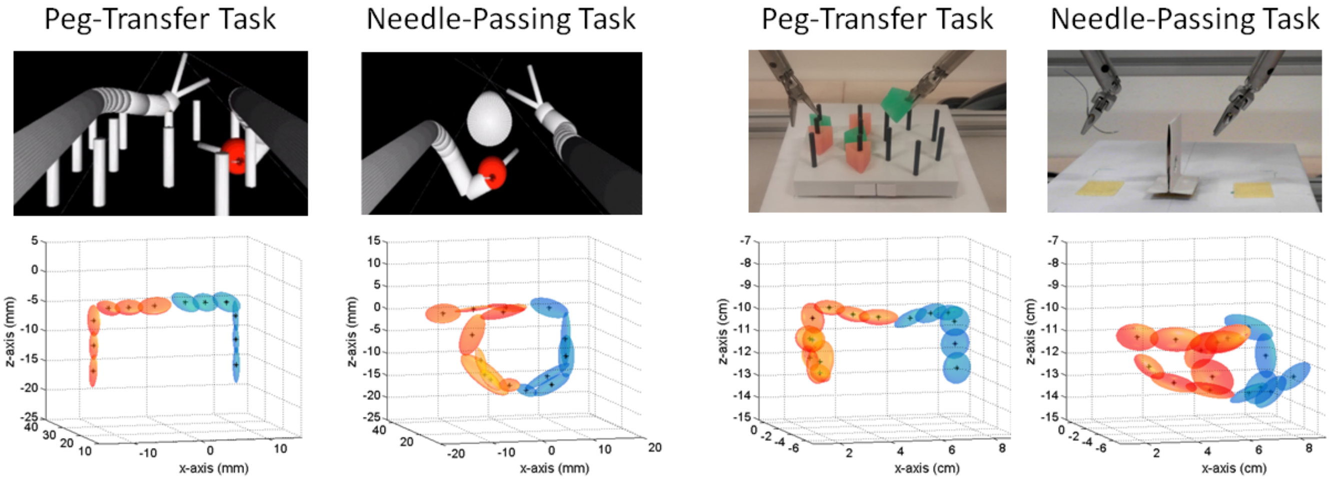


Fig. 6: Equivalent encoded peg-transfer and needle-passing tasks for the robot simulator and the Raven II robot

and more well-defined task due to the clearly visible pegs on the board. The experiments are conducted by observing the task via a live, non-stereo camera (as is typically used with the FLS program) feed on a monitor. For the needle-passing task, without stereo vision, the location of the hole is much more difficult to gauge and compromised depth perception makes the task somewhat challenging. The usefulness of the haptic guidance can then be evaluated on the grounds that it should help to keep the user within the constraints of the learned task model.

C. User Study Tasks

Prior to task executions, the tools are returned to a predefined start position. This ensures consistent initial conditions and a fair comparison between all evaluation studies. Once the left and right instrument end-effectors are in their initial positions, task execution begins. For the guided tasks, haptic guidance is always initiated by the user using the learned sequence recognition procedure described in the previous sections.

1) Needle-Passing User Study: For the needle-passing user study, each user was initially shown a figure describing the steps of the guided task, Fig. 7. Each user was allowed two to three practice sessions with and without haptic guidance. The purpose of the practice session was two-fold; (a) to become familiar with the Raven II teleoperation, and (b) to become accustomed to the haptic guidance and automatic engagement of the haptic guidance. After the practice sessions, the guided and unguided task performances were recorded for analysis.

The steps of the needle-passing task are 1) bring the left instrument (holding the suture needle) towards the hole in the obstacle, 2) bring the right instrument to grab the needle through the hole, 3) bring the right instrument above the obstacle, 4) bring the left instrument above the obstacle to grab the needle and 5) pull the suture needle to the left. The start of the task is defined by crossing a vertical plane slightly to the right of the initial position. The end of the task is determined by the left instrument crossing a boundary on the left of the workspace.

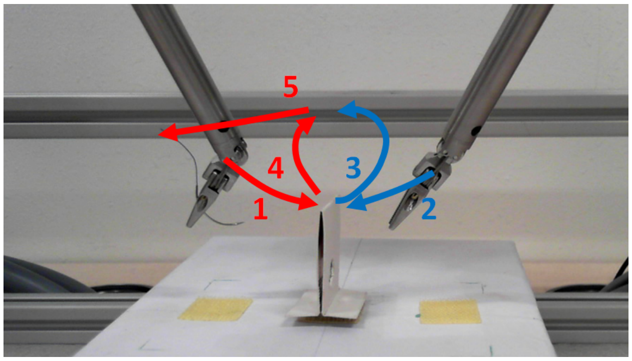


Fig. 7: Needle-passing task

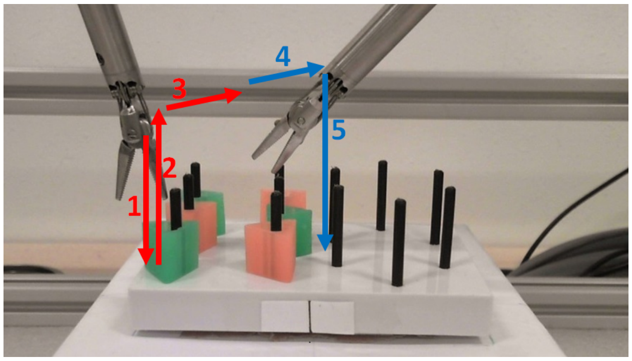


Fig. 8: Peg-transfer task

2) Peg-Transfer User Study: The peg-transfer user study was carried out in the same manner as the needle-passing user study, and a diagram of the steps are shown in Fig. 8. In this case, however, users were allowed just one or two practice sessions with and without guidance due to the the fact that they had already performed the needle-passing task.

The steps of this task are 1) bring the left instrument down to grab the triangle, 2) pull the triangle off the peg, 3) bring

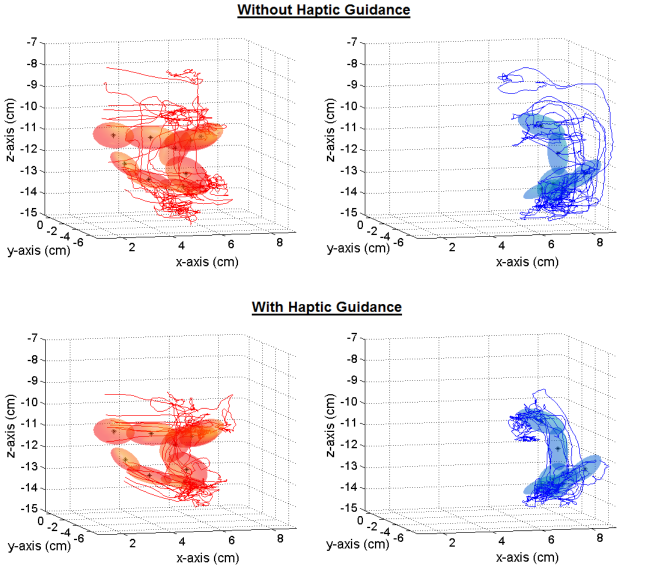


Fig. 9: Trajectories of needle-passing users studies with and without haptic guidance

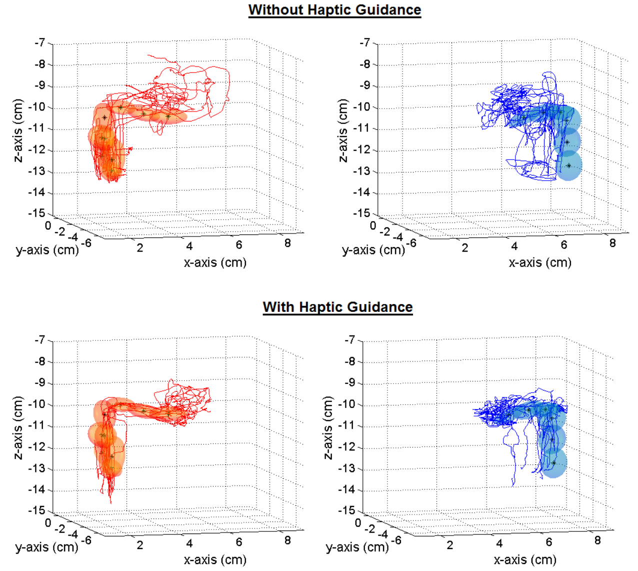


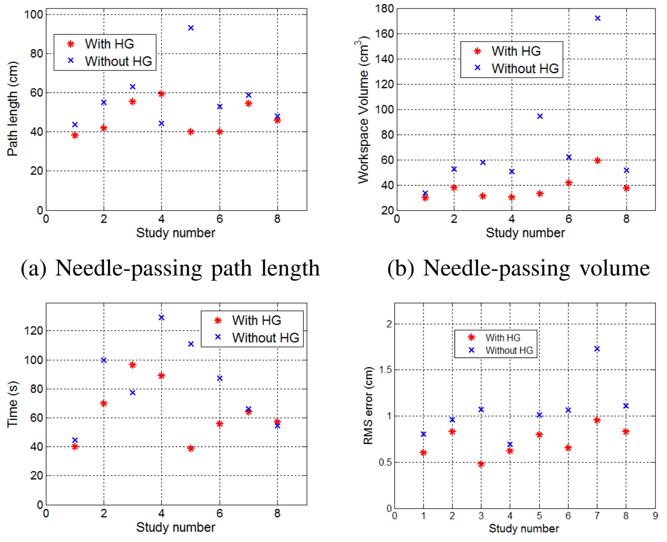
Fig. 10: Trajectories of peg-transfer users studies with and without haptic guidance

the left instrument to meet the right instrument, 4) pass the triangle from the left to the right instrument and move the right end-effector above the right peg and 5) bring the triangle down onto the right-hand side peg. Unlike the needle-passing task, there is a lot less room for variation in this task. The start of the peg-transfer is defined as the beginning of the downwards movement from the initial position. The task is considered complete when the triangle is successfully placed on the right-hand side peg.

IV. Results

Results of trajectories with and without haptic guidance for the needle-passing and peg-transfer tasks are shown in Fig. 9 and Fig. 10, respectively. The raw trajectory data already shows how the haptic guidance helps to keep the users within the desired region.

Four metrics were calculated from the raw trajectory data to assess the performance of task executions, including path length, workspace volume (a cuboid calculated based on the minimum and maximum values of the trajectory Cartesian values), time taken to complete the task and the root-mean squared (RMS) error. The RMS error is calculated based on the minimum distance between each trajectory point from the user's task execution and the regressed setpoint trajectories.



(a) Needle-passing path length

(b) Needle-passing volume

(c) Needle-passing time taken

(d) Needle-passing RMS error

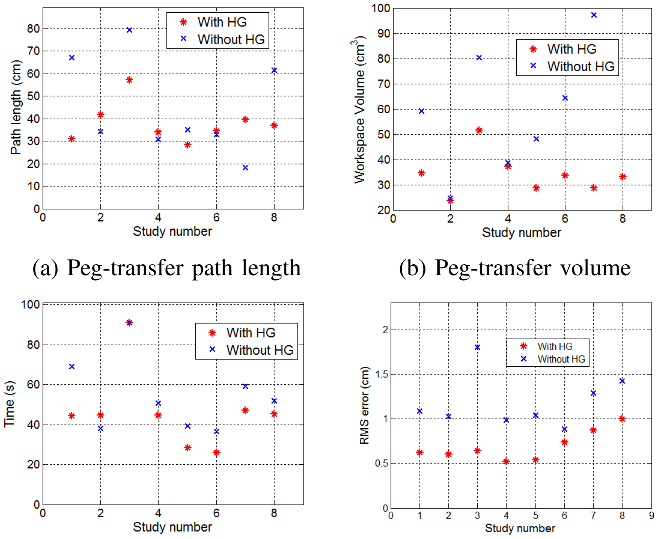
Fig. 11: Needle-passing task analysis

TABLE I: Individual user improvement with haptic guidance

Individual user task improvement with guidance	*Needle-Passing*	*Peg-Transfer*
Path length	7/8	4/8
Workspace volume	8/8	8/8
Time taken	6/8	7/8
RMS error	8/8	8/8

The results of these for the eight user studies for the needle-passing task are plotted in Fig. 11 and for the peg-transfer task in Fig. 12. There is a statistically significant improvement in the workspace volume and RMS for both tasks ($p \leq 0.05$). As expected, the guided tasks result in far smaller RMS error. This metric proves that for the guided cases there is good consistency between the different users. For the unguided cases there is a large variation in the trajectories between users. For all four metrics, upon inspecting the results of individual users, it can be seen that in the most cases the haptic guidance improves the performance in all measured respects. Table I shows that, in the majority of cases, there is an improvement made by each user with haptic guidance.

Overall, based on these user studies, the contribution of haptic guidance appears to be greater for the needle-passing



(a) Peg-transfer path length

(b) Peg-transfer volume

(c) Peg-transfer time taken

(d) Peg-transfer RMS error

Fig. 12: Peg-transfer task analysis

task. This is due to the fact that there is a lot more room for error caused by poor depth perception in this task. The peg-transfer task is much simpler and the pegs serve as depth cues to the user. In the case of the needle-passing task, where there were fewer landmarks to use as visual aids, the guidance particularly helped by keeping the user within the correct plane of the task and ensures that the instruments were kept roughly in line with each other. Therefore, the value in this framework lies with more complex tasks that benefit from help with depth awareness.

Another aspect of the user studies was to evaluate the ability of the framework to recognise the beginning of the learned model for automatic engagement of haptic guidance. In some cases this took more than one attempt, in which case the end-effector position were re-initialised to the fixed starting position before starting the task again. All instances of guided task execution were initiated using the sequence recognition procedure.

The strength of the force feedback applied was proportional to the error between the current position and the closest point on the current desired trajectory. The scaling factor is parameter which could be experimented with in future studies. In addition, instead of proportional force-feedback, a number of alternative functions could be used to modulate the force, including Gaussian covariance based or velocity based force feedback.

V. Conclusions and Future Work

In this paper, we have investigated cooperative control based on LfD for surgical task execution using haptic guidance. Surgical robotics have already proven their value by bringing many benefits to the operating theatre, but there is a great potential to further enhance human-robot interaction beyond the current paradigm of master-slave control. The main contributions of this work include 1) a LfD framework with a generalised method of surgical task encoding with minimal manual input, 2) a method of sequence recognition based on the proposed surgical task model encoding which looks at both full and partial trajectories to identify the start of learned model. When the start of a learned model is identified, guidance is initiated and 3) a cooperative control framework incorporating haptic guidance for the learned task models on the Raven II robotic platform. The results from two user studies have shown statistically significant improvements in workspace volume and RMS error. Path length and time taken were also improved, in general, on an individual-basis.

This proposed framework has scope for applications in several areas of surgical task execution, from surgical training to skills evaluation to preoperative assistance. The immediate applications of the proposed framework lies in skills transfer and, with further enhancements, applications in real surgical scenarios. For future experiments, an interesting investigation would be to compare the rate of improvement for task performance using haptic guidance for training versus performance improvement without haptic guidance using two groups of subjects. The hypothesis would be that guided demonstrations of tasks will speed up the rate of performance improvement versus learning the task through observation alone.

Acknowledgements

The authors would like to thank Hawkeye King for his help with the Raven II robot.

References

[1] J. D. Wright, W. M. Burke, E. T. Wilde, S. N. Lewin, A. S. Charles, J. H. Kim, N. Goldman, A. I. Neugut, T. J. Herzog, and D. L. Hershman, "Comparative effectiveness of robotic versus laparoscopic hysterectomy for endometrial cancer.," *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, vol. 30, pp. 783–91, Mar. 2012.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, May 2009.

[3] N. Padoy and G. D. Hager, "Human-Machine Collaborative surgery using learned models," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5285–5292, IEEE, May 2011.

[4] C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: learning from expert demonstrations.," in *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2010, pp. 967–70, Jan. 2010.

[5] A. Knoll, H. Mayer, C. Staub, and R. Bauernschmitt, "Selective automation and skill transfer in medical robotics : a demonstration on surgical knot-tying," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. May, pp. 384–397, 2012.

[6] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2074–2081, IEEE, May 2010.

[7] P.-l. Yen, "Intelligent human-machine cooperative robot for orthopaedic surgery," *APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems*, vol. 1, pp. 741–744, Nov. 2008.

[8] F. Ryden and H. J. Chizeck, "Forbidden-Region Virtual Fixtures from Streaming Point Clouds : Remotely Touching and Protecting a Beating Heart," in *International Conference on Robotics and Automation*, pp. 3308–3313, 2012.

[9] flsprogram.org (last accessed 18/09/2014), "FLS Manual Skills Written Instructions and Performance Guidelines," tech. rep.