On Application of Virtual Fixtures as an Aid for Telemanipulation and Training

Shahram Payandeh and Zoran Stanisic
Experimental Robotics Laboratory (ERL)
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
CANADA, V5A 1S6
shahram@cs.sfu.ca

Abstract

This paper presents a study on the application of virtual fixtures as a control aid for performing telemanipulation or in the training environment. The implementation features both manual and supervisory control modes. It also studies approaches for using virtual fixtures in generating visual and force clues and/or for restricting the motion of the slave using the definition of virtual fixtures. In this preliminary study it was found that, the virtual fixtures could improve the speed and precision of the operator, reduce the operator's workload and the duration of the training phase for the novice operator.

1. Introduction

Virtual fixtures can be defined as a task-dependent aid for telemanipulation and training. Their usage can be extended to a graphical overlay on the actual visual scene and/or as an invisible agent that can act as a tutor or guidance. They can assist the operator or trainee by providing relevant visual and/or force information. Among issues that virtual fixtures have been intended to address are: high operator load, complex and unpredictable algorithms for supervisor control, task performance degradation due to communication time delay and limited reusability of paths generated by manual master/slave control.

In general, control paradigms used to control remote manipulator represent a spectrum from high-level supervisory control to low-level manual control [1]. High-level supervisory control refers to operating the manipulator by specifying infrequent high-level symbolic commands, while low-level manual control assumes continuous teleoperation using a hand controller with and without force feedback.

Previous implementation of virtual fixtures showed that they can increase speed and precision of operation, reduce operator workload, and reduce the effects of communication time delays [2],[3]. Another important area of application of such fixtures is in the area of tutoring and mentoring in the virtual training environment such as laparoscopic training environment [9]. For example, similar to task-oriented tutorial proposed in [4][5], and in combination with force feedback master device [6], virtual fixtures could be used as a source of guidance and navigation. Similar concepts have also been previously proposed in [7] for developing virtual teacher. For surgical training application such fixtures can be used in both training and actual operation through the fixtures overlay. This paper presents preliminary extensions of the previous application of the virtual fixtures in the context of robotic control system and the task decomposition of a typical teleoperation.

In the current application, the control system configures virtual fixtures in the environment based on the model of an environment and information regarding the tasks. During the manual control task, virtual fixtures automatically generate symbolic commands appropriate to the task. These fixtures can also be used as a guidance that can monitor and suggest suitable motions and/or actions. Because each planned path contains task information that the fixture is assigned to, we are able to significantly increase the reusability of their position/force definitions. In general it was found that virtual fixtures offer a bridge between the supervisory and manual control modes, which offers the operator intermediate access to these modes.

This paper is organized as follow: section 2 presents an overview of the system description; section 3 presents the notion of the virtual fixtures used in this paper; section 4 some teleoperation issues related to the implementation and usage; section 5 presents a preliminary evaluation of the implementation of the virtual fixture and finally section 6 presents some concluding remarks.

2. Control System Architecture

Figure 1 shows the graphical representation of a slave device used in this study. The operator workstation consists of model of the interaction world of the slave device. The 3D objects are used to help the operator to visualize the environment; it can display either simulated information as a graphical objects or feedback information from the slave site. The 3D mechanics based objects are also used by the control algorithm, e.g. for path planning or collision detection.

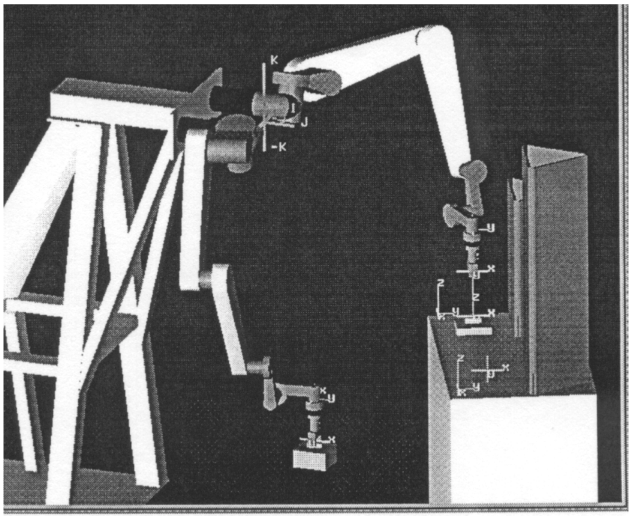
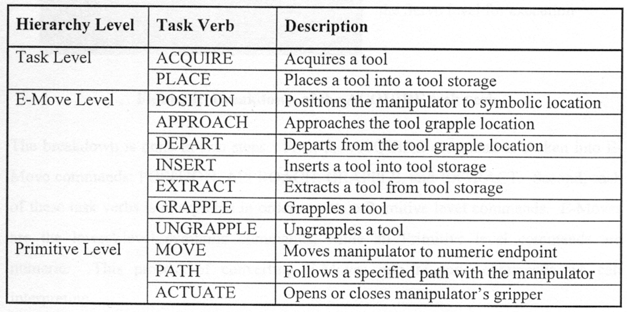


Figure 1 - Graphical mode of the actual manipulator with a model of the interacting environment.

The system supports a number of control modes, ranging from the high-level supervisory control and low-level manual control. During supervisory control operation, the operator issues high level commands (task verbs, e.g. *acquire* and *place, Figure 2*). The system decomposes these commands into lower level symbolic commends, and then interprets them into numerical commands by invoking the trajectory planner. An example of command breakdown for hierarchical tasks of *acquire* task verb is done in two steps. The command is first broken into low-level symbolic commands such as: position, approach, grapple, and extract. Then path and trajectory planner interprets each of the low-level commands in order to obtain numerical commands suitable for execution by the low-level servo controller.



Hierarchy Level	Task Verb	Description
Task Level	ACQUIRE	Acquires a tool
	PLACE	Places a tool into a tool storage
E-Move Level	POSITION	Positions the manipulator to symbolic location
	APPROACH	Approaches the tool grapple location
	DEPART	Departs from the tool grapple location
	INSERT	Inserts a tool into tool storage
	EXTRACT	Extracts a tool from tool storage
	GRAPPLE	Grapples a tool
	UNGRAPPLE	Ungrapples a tool
Primitive Level	MOVE	Moves manipulator to numeric endpoint
	PATH	Follows a specified path with the manipulator
	ACTUATE	Opens or closes manipulator's gripper

Figure 2 Task verb hierarchy.

The robots can also be controlled in manual control mode via a hand controller device, e.g. a master controller with or without haptic feedback. Beside providing supervisory and manual control modes, the teleoperation system attempts to combine the best features of both of these control modes by using the notion of the virtual fixtures which will be discussed in the following section.

We can define the virtual fixture control mode as an extension of the manual control mode, where virtual fixtures are places in the environment based on the task knowledge typically related to supervisory control. For example, if one wants to train a novice operator, one can defined forbidden regions virtual fixtures (FRVF). When the operator moves the slave manipulator in these regions, a number of events can be initiated such as: a) graphical clues can be generated to flag operator of initiation of such events; b) the force feedback can be generated at the haptic master device and/or c) the embedded command in the FRVF can guide or maintain the slave robot at a safe configuration.

3. Virtual Fixtures

Virtual fixtures can be defined as active, task-dependent software agents, whose purpose is to assist the operator during the training or execution of a task. Here task training refers to controlling a robot in a virtual or simulated mode, which can then be mapped to the actual execution. The tutoring and guidelines provided by virtual fixtures may consist of force clues, additional motion commands applied to the manipulator itself, restriction of the motion of the manipulator, or graphical overlay which can be used for visual clues. When the teleoperation system features a haptic force-feedback master device, force clues can be applied to the hand controller [3]. Otherwise, their operation is implemented using motion commands and movement restriction.

Virtual fixtures represent a task-dependent geometry overlaid onto the environment, and typically attached to an object in that environment. For example, a virtual fixture can be attached to a moveable object to guide the robot gripper to the object's grasping point. As a geometrical constraint in the task space configuration manifold (configuration space or C-Space), the fixture imposes restrictions to the robot's movements, thus ensuring that the robot does not collide with the object. Furthermore, the fixture can applied additional motion commands to the robot to attract it to the grasping point, therefore simplifying the positioning process. Although their control properties have not been studied, but this type of fixtures can be further abstracted as a sliding surface. Here, a point in the C-Space that represents the configuration space of the robot gripper can move (slide) along the sliding surface toward the goal. Through interaction using a haptic device, the user is able to slide along the surface toward the goal while feeling the virtual reaction force normal to this surface.

The key requirement for application of the virtual fixture is task dependency, which has a major influence on our implementation. For example, in our case study, we incorporated virtual fixtures into the task knowledge base. The task knowledge base consists of a number of low-level symbolic commands, each of them containing information relevant to preparation or execution of the associated robotic task. The virtual fixture data is added to the task high-level knowledge base, and describes ways in which the system can assist the operator during the execution of the task using a virtual fixture control paradigm.

Some of the responsibilities of virtual fixture in our system are:

- Generating additional movements of the robots, e.g. homing motions
- Restricting motion of DOFs of the robots
- Setting low-level controller parameters, e.g. impedance parameters for the haptic feedback device
- Providing cooperative control for the dual robots

Let us consider how fixtures assist the operator during the execution of the position or approach task. The operator normally provides inputs to the hand controller master device, while the fixture provides additional motion commands. These motion commands are presented using the arrows; see Figures 3-5 (note that the figures only show commands that related to achieving the proper position of the end-effector in the workspace, while the commands that relate to achieving the proper orientation are not shown).

3.1 Generating Visual Clues

While operating, it is beneficial if the fixtures communicate to the operator the fact that they are assisting in the task and acting as a mentor or a tutor. Superimposing some visual cues overlaid on the work cell can offer such clues. Figure 3 shows a very simple example of the visual clues for position and approach fixtures. It shows the relevant end-effecotr frame and the direction of translation from the position configuration during the approach phase.

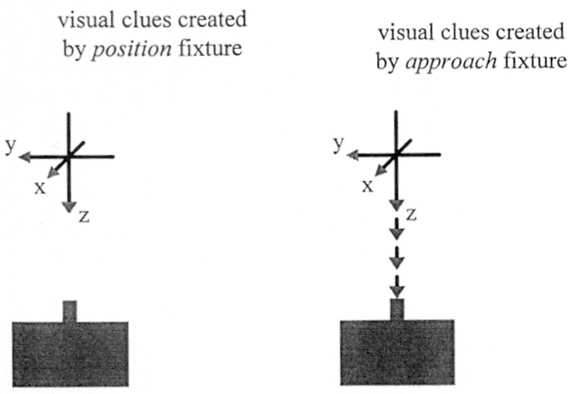

visual clues created by *position* fixture
visual clues created by *approach* fixture
y
x
z
y
x
z

Figure 3 Visual clues generated by position and approach fixtures.

3.2 Generating Force Clues

One of the most important aspects of fixture operation is the force clue that the fixtures is providing. These force clues for example can be provided directly to the operator using haptic force feedback device as shown for example in [3]. They can also be applied directly to the manipulator, when we use a master controller without force feedback. The force clues generated by position and approach fixtures are shown in Figure 4. The forces are applied to the manipulator's end-effector, in order to bring it to the approach position in case of the position fixture, or to the grasp position in case of the approach fixture. They can be generated through impedance control methodologies.

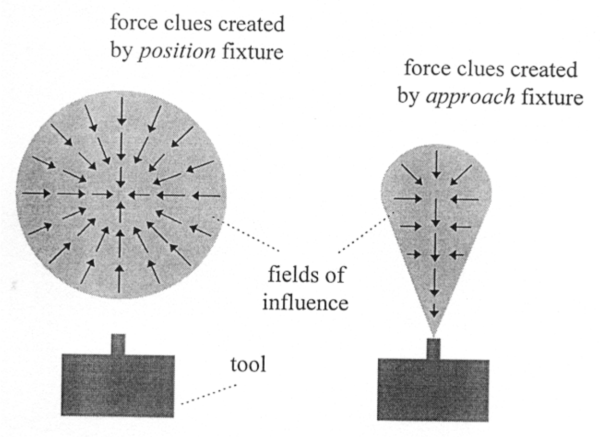

force clues created by *position* fixture
force clues created by *approach* fixture
fields of influence
tool

Figure 4 Force clues generated by position and approach fixtures.

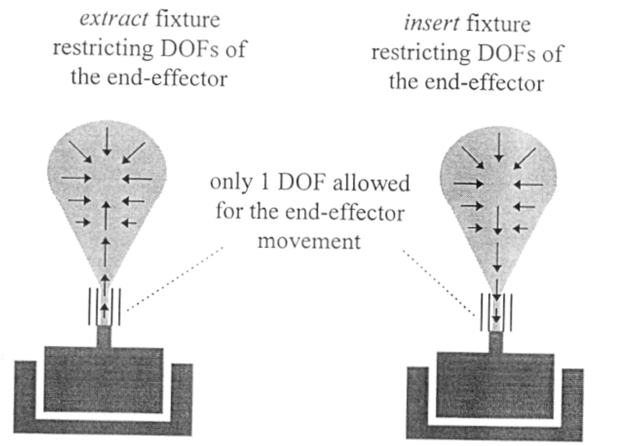

extract fixture restricting DOFs of the end-effector
insert fixture restricting DOFs of the end-effector
only 1 DOF allowed for the end-effector movement

Figure 5 Fixtures restricting degrees of freedom.

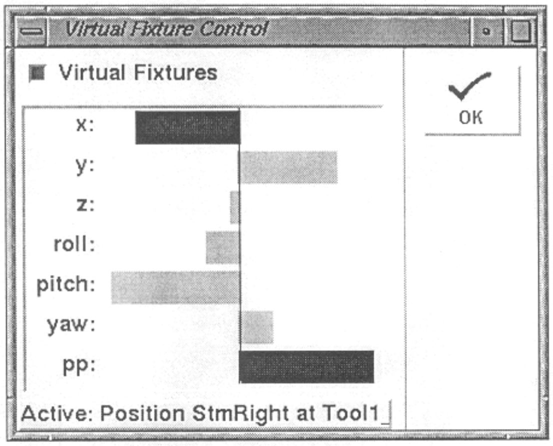
3.3 Restricting Manipulator Motion

Besides providing force clues, fixtures can help the operator by restricting the motion of the manipulator. A possible implementation of fixtures restricting the motion of the manipulator is shown in Figure 5. As shown, extract and insert fixtures restrict the motion of the end-effector when it is close to the tool grasping position. This behavior is implemented in order to avoid a collision of the manipulator with the tool, while allowing the operator to quickly extract/insert the grasping position.

To avoid conflict with the operator's commands, the fixture considers only the degrees of freedom not specified by the operator. Thus, if the operator releases the master controller, the fixture will provide motion commands for all DOFs of interest. For example as shown in Figure 5, the virtual fixture control dialog box shows the hand controller input from the operator (the dark bar), and the motion commands generated by active virtual fixture (light bar). In this example, the operator sets the x coordinate and the pitch plane angle (pp), while the fixture controls the other coordinates in the task space.

For some tasks it is beneficial to restrict the manipulator's motion (because of the interaction between robot and environment). For the approach task, we want to avoid collisions with the target, while allowing the operator to quickly approach the grasping position. For the extract task we want to simplify the extraction process, thus we restrict the degrees of freedom in the task space. The restriction of the motion is also shown by double bar in for example in Figure 5.

In the control system in general, the definition of virtual fixtures can be further extended so they can be responsible for setting low-level control system parameters. For example, we want a different type of robot control algorithm when the robot is in free space compared to when the robot is in impact and then contact with the environment [10]. Based on the task knowledge, the virtual fixtures set the appropriate control parameters for the task being executed.



Active: Position StmRight at Tool1

Figure 6 Interaction with the operator during task preparation.

Finally, virtual fixtures are used to simplify the simultaneous control of dual arm robots. Suppose we want to perform a task where the coordinated motion of both robotic arms is required (e.g. to carry a heavy object). The virtual fixture is embedded with the knowledge that both robots need to be moved at the same time while satisfying the constraint that they are coupled together.

4. Implementation Issues

The virtual fixture control mode can be used within two contexts: when providing an alternative for path planners during supervisory control, or when providing a task-dependent aid for manual control during training or actual execution in the master/slave mode.

When used as an alternative for path planners in supervisory control, the task is known in advance for execution, i.e. it is specified by the operator. The appropriate fixture is placed in the environment (activated) when the operator starts the execution of the task using the virtual fixture control mode.

When used as an aid for manual control, the work task is not known in advance. Activation of virtual fixtures in this case is done reactively based on the task preconditions. Basically, one fixture is activated for each task whose preconditions (defined in the task knowledge based) are satisfied. When the task is ended or the failure conditions are satisfied, the fixture is deactivated.

Whether used as an alternative for path planners, or as an aid for manual control, virtual fixture control combines the supervisory and manual control approaches. When used as alternative for path planners, the fixtures make sure that the manually generated path plans are placed within the constraints of the workspace. Also, during the generation process, task knowledge is used to simplify the operation, i.e. by placing appropriate virtual fixtures.

During manual control, virtual fixtures can generate new high-level commands to describe the actions that were specified by the operator during the work task. This high level commands generation is performed when a virtual fixture is completed, i.e. when the task post conditions are satisfied. For example, consider the acquire task consisting of the position, approach, grasp, and extract subs tasks, Figure 7.

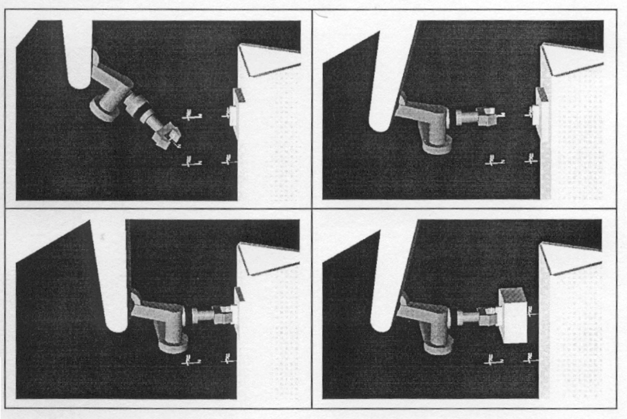


Figure 7 The acquire process sub task.

Whenever a subtask is finished, the appropriate task commands is assigned to the path that was specified by the operator. The positions of the end-effector at which these subtasks were completed are visually represented by black circles (Figure 8). When all four subtasks are complete, the acquire task command is completed, the acquire task command is generated, and the previously generated subtasks are assigned to it.

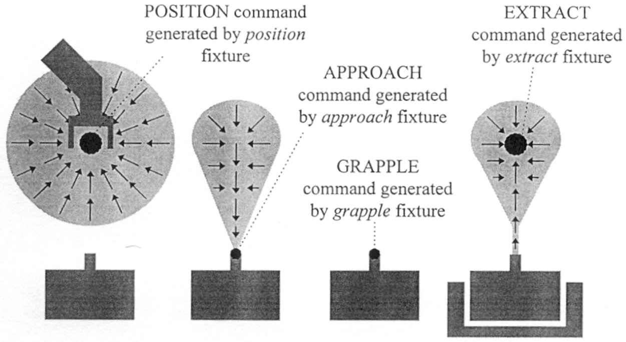

POSITION command generated by *position* fixture
APPROACH command generated by *approach* fixture
GRAPPLE command generated by *grapple* fixture
EXTRACT command generated by *extract* fixture

Figure 8 Command generation by acquire fixture.

5. Preliminary System Evaluation

One of the benefits of virtual fixture control as compared with the manual control is reduced operator workload. The fixture participates during the execution of the task, and offloads parts of the work from the operator. For example, the operator could take care of setting the position coordinates, while the fixture could set the orientation. This would be an example of shared control of the manipulator [11].

In some cases, the fixture would be able to perform the task appropriately without any input from the operator. Thus the control responsibility would switch between the operator and the virtual fixture.

Consider the example of the acquire task performed using virtual fixture control (Figure 9). The operator is mostly involved in the first phase of the process, i.e. generating a collision free path. Once the end-effector is closed to the fixture, a period of shared control exists where the operator for example sets the pitch plane angle, Figure 8. Hence, the virtual fixtures can also increase the speed and precision of robot operation.

We conducted an initial evaluation of such implementation of virtual fixture where several operators (six subjects) with skills ranging from novice to skilled performing a positioning task at their own pace,[12]. The task was first performed with virtual fixtures and then without them, and duration of operation was measured.

As shown in Figure 10, the results indicate that the virtual fixtures can improved the operation speed for novice and intermediate operators. Furthermore during this preliminary users study it was noted that all the operators considered virtual fixture control easier than manual control.

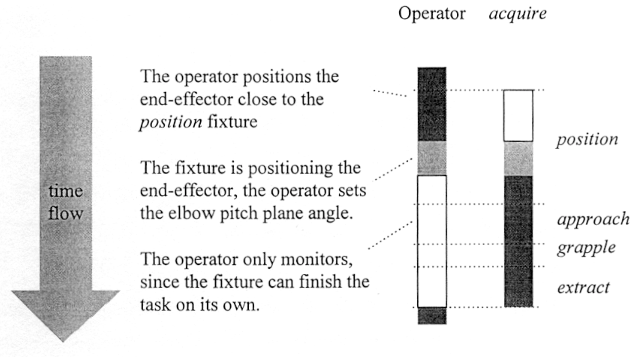

Operator
acquire
The operator positions the end-effector close to the *position* fixture
The fixture is positioning the end-effector, the operator sets the elbow pitch plane angle.
The operator only monitors, since the fixture can finish the task on its own.
time flow
position
approach
grapple
extract

Figure 9 An example where it shows how acquire fixture reduces operators' load.

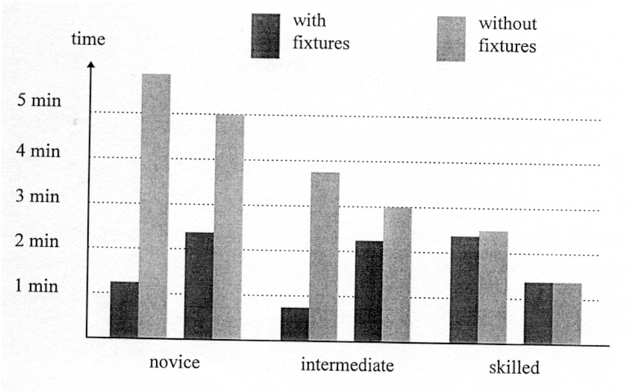


Figure 10 Performance measurements for the Position task.

6. Conclusion and Future Work

In this initial study it was found that the benefits of implementing virtual fixtures in our teleoperation system are numerous. With virtual fixture control we obtained a performance increase and reduced operator workload compared with ordinary manual control.

The virtual fixture control can provide a scheme where manual control can be used as an alternative to path planners, while preserving the task information typically related to supervisory control. It also provided automatic command generation when used instead of manual control, thus establishing a closer relation between the supervisory and manual control paradigms.

For our study we have found that we can increase the reusability of paths generated with manual control. Since during the operator of the manipulator we obtained not just numeric paths, but their corresponding symbolic commands as well, we can easily check whether those path plans can be reused when that command is executed again.

Because of the initial modeling required for the virtual fixtures, they are appropriate for repetitive tasks, or those where speed and precision are essential. An example of future work can be an application of virtual fixtures to remote laparoscopic surgery and development of training and tutoring workstation.

The findings of this study are being evaluated in context of the development of laparoscopic Training Environment (LTE). Here the notion of the virtual fixture is being defined in the context of tutoring and mentoring agent. In this environment, various modules are being defined as a part of the training curriculum.

7. Acknowledgement

This work is partially based on [8] and the results evaluated during the authors sabbatical stay with the International Submarine Engineering (ISE) Inc. in Port Coquitlam, BC, Canada. The examples and hardware used for obtaining the results of this paper were carried during the stay at ISE. The cooperation of Mr. Eric Jackson is highly appreciated.

8. References

[1] J. Albus, H. McCain and R. Lumia, R.,"NASA/NBS Standard Reference Model for Telerobot Control System Architecture", National Bureau of Standard Robot Systems Div., Dec. 1986

[2] L. Rosenberg, "The use of Virtual Fixtures to Enhance Telemanipulation with Time Delay", Proceedings of ASME Advances in Robotics, Mechatronics, and Haptic Interfaces, DSC-Vol. 49, 1993

[3] C. Sayers and R. Paul, "An Operator Interface for Teleprogramming Employing Synthetic Fixtures", Presence, Vol. 3, No. 4, 1994, pp. 309-320

[4] A. Corbett and H. Trask, "Instructional Interventions in Computer-Based Tutoring: Different Impact on Learning Time and Accuracy", Proceedings of CHI 2000, pp. 97-104

[5] J. Rickel and W. Johnson, "Animated agents for procedural Training in Virtual Reality: Perception, cognition, and motor Control", Applied Artificial Intelligence, Vo. 13, 1999, pp. 343-382

[6] J. Dennerlein, D. Martin and C. Hasser, "Force-Feedback Improves Performance for Steering and Combined Steering-Targeting Tasks", Proceeding of CHI, 2000, pp. 423-429

[7] R. Gillespie and M. O'Modhrain, P. Tang, D. Zaretzky and C. Pham, "The Virtual teacher", Proceedings of ASME Dynamic Systems and Control Division, Symposium on Haptic Interface for Virtual Environment and teleoperator Systems, 1998, pp. 171-178

[8] Z. Stanisic, "Virtual Fixtures as an Aid for Teleoperation", Master of Science Thesis, School of Engineering Science, Simon Fraser University, 1996

[9] C. Cao, C. MacKenzie and S. Payandeh, "Task and Motion Analysis in Endoscopic Surgery", Fifth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator System, 1996, pp. 583-590

[10] S. Payandeh, "A Natural Framework for Designing a Bounceless controller for Robotic Contact Tasks", Proceedings of IFAC, SYROCO conference, 1997, pp. 709-714

[11] S. Payandeh, "Application of Shared Control Strategy in the Design of Robotic System", Proceedings of American, 2001

[12] H. Kerbs, N. Hogan, M. Aisen and B. Volpe, "Robot-Aided Neuro-Rehabilitation", IEEE Trans. On Rehabilitation Engineering, Vol. 6, No. 1, 1998, pp. 75-78