Haptic Control of Multi-Axis Robotic Systems

Christopher Mawson, James Mullins, Timothy Black and Saeid Nahavandi

crmawson, mullins, tjblack, nahavand@deakin.edu.au
Intelligent Systems Reaseach Lab.,
School of Engineering and Information Technology
Deakin University, Geelong, Australia

Abstract

Control of tele-operated remote robot's is nothing new; the public was introduced to this 'new' field in 1986 when the Chernobyl cleanup began. Pictures of weird and wonderful robotic workers pouring concrete or moving rubble flooded the world. Integration of force feedback or 'haptics' to remote robot's is a new development and one that is likely to make a big difference in man-machine interaction.

Development of haptic capable tele-operation schema is a challenge. Often platform specific software is developed for one off tasks. This research focussed on the development of an open software platform for haptic control of multiple remote robotic platforms. The software utilises efficient server/client architecture for low data latency, while efficiently performing required kinematic transforms and data manipulation in real time. A description of the algorithm, software interface and hardware is presented in this paper. Preliminary results are encouraging as haptic control has been shown to greatly enhances remote positioning tasks.

1 Introduction

During the development of tele-operated robot control systems careful consideration and modelling is required to ensure system stability. Usually efforts are directed at hardware and firmware improvement and as a result a Graphical User Interface (GUI) is of little concern. The Human Machine Interface (HMI) is not dissimilar. Often a joystick or set of arrow keys is added seemingly as an afterthought. This may be adequate in controlling systems in the two dimensional world but for control in three dimensions they often perform inadequately.

Over the past decade the development of relatively low cost three-dimensional haptic interfaces from several companies has allowed experimentation in control with one important addition; force. Our work has led to the addition of a sensory system to multiple robot's with the goal of aiding the navigation and/or control of each individual controllable axis. Software communicates with these systems and formulates a haptic response before transmitting it to the user through standard teleoperation principles. This process happens at between 500 to 1000Hz as per the requirement of true haptic force reflection [GHOST SDK, 2003]. This paper showcases the algorithms and software successfully used by the Intelligent Systems Research Group within Deakin University for transmitting haptic data to remote systems. Related work is presented in section 2. Section 3 explains our software development. Software integration and platforms are presented in section 4 and 5, followed by results and conclusion in sections 6 and 7 respectively.

2 Related work

In the past few years, the use of haptic technology has been accepted into academia for simulation and modelling [Williams, 2002, Kress et al, 2001]. Occasionally haptic technology has been used for robot control [O'Malley et al, 2003, Williams, 2002] but haptics has yet to make inroads into industry as a tool for rapid three-dimensional programming of point data. Controlling robot arms with haptic devices has known benefits for grasping and positioning tasks [O'Malley et al, 2003, Dennerlein, 2000] but has shown limited acceptance when controlling mobile platforms.

Haptic applications range from remote operations in hazardous environments (generally hazardous material manipulation) [Everett et al, 1999], space and deep-sea research operations [Hsu et al, 1999], [Hirzinger et al, 1993], remote object deployment [Pathirana et al, 2005] and military operations [Ryu et al]. [Lee et al, 2002] provided a novel method of controlling a mobile platform utilising a haptic interface but this methodology required significant effort in order to enable the integration of one type of robot into the haptic controller. [Dennerlein et al, 2000], performs a similar control scheme to Lee but once again limits the feedback to a set number of axis. [Elhajj et al, 2001] examined several issues relating to haptic teleoperation, Stability, synchronization, and transparency. He suggested that event-based planning was adequate for control and haptic feedback. The provision for multi-axis control of a single robot, utilising a single haptic device, can significantly reduce costs while benefiting productivity [Mullins et all, 2002, 2004]. Therefore software being developed to control robot motion must be capable of supporting real time operation without compromising performance.

3 Software

When developing any form of haptic software it is important not to overcomplicate the main functional code loop. Any code redundancy will severely reduce the haptic feedback rate and as a result reduce the quality of the user experience. To facilitate multiple robot platforms, our software has been modularised significantly. Figure 1 shows the interconnection of information passing among networked nodes. The development of a custom OLE Control Extension (OCX) enabled low-level control of haptic devices in real time. Multi-platform server software is then utilised to connect to the OCX and is rapidly configured as per the robot's requirements.

3.1 OLE Control Extension (OCX)

In order to increase the effectiveness of haptic control in robotics a standard needed to be implemented. This standard interface allows the control of multiple robot's without significantly increasing coding time. Without this flexibility haptic control remains a project specific problem. To create the required flexibility without sacrificing functionality we were required to design new software that built upon the original HD and HL libraries that the Sensible haptic devices use. This higher-level language is able to function with most other Windows based software on the market; it also fits the requirement of communicating via TCP/IP, RS232 and UDP. The software is also capable of taking advantage of every property the attached haptic device possesses. The requirements call for low cost haptic devices that can be integrated into a wide range of markets. As a result, the Phantom range of haptic devices from SensAble® were chosen for development.

Significant time is required to learn the essential Application Peripheral Interface (APIs) for programming Sensable devices and C++. It became clear that to open haptics up as a standard for engineering use, required the implementation of a custom OCX. This allows nearly any computer language from Basic to HTML to interact with the haptic world.

Figure 1. Data connection

The resultant OCX can call all the Sensable functions, pass data via multiple protocols, communicate with force/torque sensors and be programmed to pass position, velocity, acceleration and force data to any desired robot efficiently.
Table 1 shows the compression of Sensable's API to into a higher level OCX.

Table 1. API compression to OCX

Sensable's HDAPI / HLAPI	Deakin OCX

bool out;
HDErrorInfo error;
HHD hHD=hdInitDevice(HD_DEFAULT_DEVICE);
if (HD_DEVICE_ERROR(error=hdGetError()))
{
out = FALSE;
Return out;
}
Else
{
out = TRUE;
hdEnable(HD_FORCE_OUTPUT);
hdStartScheduler();
return out;
}
	Dim test as Boolean
	Test=haptics.HapticInit

3.2 Server Software

The server software shown in Figure 2 is the GUI developed as the "haptic server" shown in Figure 1, the software acts as middle-ware between the user and any robot system to be controlled in real time. The server generates a proxy [Abbott et al, 2005] for the haptic device, which maintains records of present and past haptic positions, velocity and forces. The proxys current position is shown on the screen in the top right hand side, overlaid with the controlled robot's current position in space. This gives the operator a visual representation of the small delay in the system between the haptic master device due to latency. This representation is also important as allows for virtual coupling [Colgate et al, 1995] increasing controllability over the attached robot.

Running on Windows, Linux or Apple's OSX, the server is capable of passing requests between several different kinds of robot's and the selected haptic device via the custom OCX and a TCP/IP or UDP connection. Because the server software is modular it enables fast editing through Object Oriented Programming (OOP). Robot's of different configurations can be added through blocks of functional code. Blocks include 6-axis arm (anthropomorphic), SCARA robot (4-axis), camera pan/tilt and a host of mobile platform choices with a variety of haptic feedback solutions.

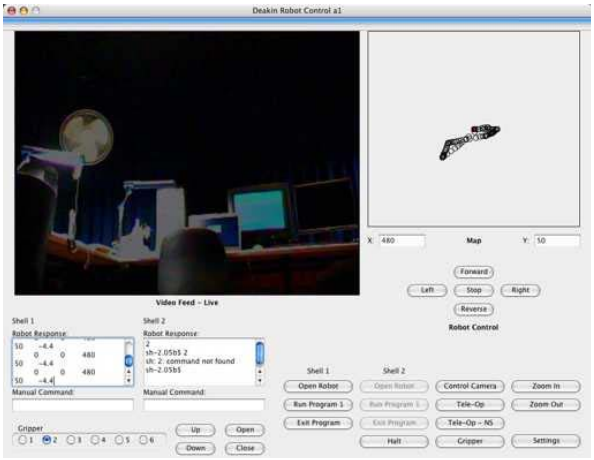

Figure 2. Haptic server software configured for ActivMedia P3-AT robot.

4 Robot Integration

Server software has been developed for passing requests between several different kinds of robot's. In order to control a robot haptically a requirement exists for matching work envelopes between the haptic device and the controlled robot. Without matching envelopes the controlled robot will behave erratically and unpredictably. In order to combat this, an extendable library of conversion values for many robot's has been generated. The library allows for the quick addition of new robot's as the need arises. To match work envelopes the software requires several pieces of information including the number of joints, maximum angle of each joint and maximum angles in desired units to be passed (this last set is optional and only required if the robot receives joint data in a different unit to degrees or radians). Conversion value, scaling value and controlling axis are then calculated.

	R1	R 2	R3	R4	R5	R6
J1	N/A	Angle	1	1	1	1
J2	N/A	Velocity	2	2	2	2
J3	N/A	Velocity	3	3	3	3
J4	N/A	N/A	N/A	4	4	4
J5	N/A	N/A	N/A	N/A	5	5
J6	N/A	N/A	N/A	N/A	N/A	6

Table 2. Robotic joints (1-6) V's Haptic joints (J1-J6).

In table 2 the columns refer to the number of Degrees of freedom the robot possesses. The rows show the corresponding haptic joints and the numbers represents which joint the haptics device will control on the robot.

Table 2 was developed based on results from testing a controller using different axis. It was found that for a user to have adequate control the joints needed to correspond quite similarly to the remote robot. The exception here is if you want to only control an individual joint. For this task we selected joint 6, as it has the largest work envelope of 290° (Phantom Omni™). The other exception is for two axis of response i.e. a mobile platform that can only move in a two dimensional plane. In this case, the preferred control schema was to use axis one to translate the platform and to use joints two and three to control the velocity of the platform.

$$Scale_{axis(x)} = \frac{R\theta_{\max}}{H\theta_{\max}} \qquad [1]$$

Where: $R\theta_{\max}$ Is the maximum degree of movement on the robot (axis x). $H\theta_{\max}$ Is the maximum range of the haptic device being used (axis x).

Scaling allows for a haptic controller to use and exploit the full work envelope of a robot. Scaling should be implemented based on the necessity of the application, as the greater the scaling factor, the larger the robot's response is to small increments of the haptic axis.

$$A = \frac{Pos_{fs}}{\theta_{fs}} \qquad [2]$$

Where: A is the conversion value of the system under test. Pos Is the full-scale position information for joint (x) and θ is the full-scale angle of the device in degrees for joint (x).

5 Robotic and Autonomous Systems

As section 4 discussed, a library was developed for the different types of robot's to be used. A series of industral robot's and autonomous platforms were modelled to validate this work, they are discribed below.

5.1 Epson Pro Six

One of the first robot's to be integrated with the new software architecture was the Pro Six PS3 from Epson Robotics. A six-axis anthropomorphic robot arm with similar reach to that of a human, the Epson is controlled remotely via a Phantom™ 1.5 6D.O.F. The standard Epson robot controller has been upgraded with a faster CPU enabling it to run a tight loop of SPEL+® code (Epson's robot control language). This code loop communicates with the haptic server and updates the robot's position at around 800Hz. A secondary program receives data from the attached JR3 force/torque sensor to provide feedback to the controlling haptic device.

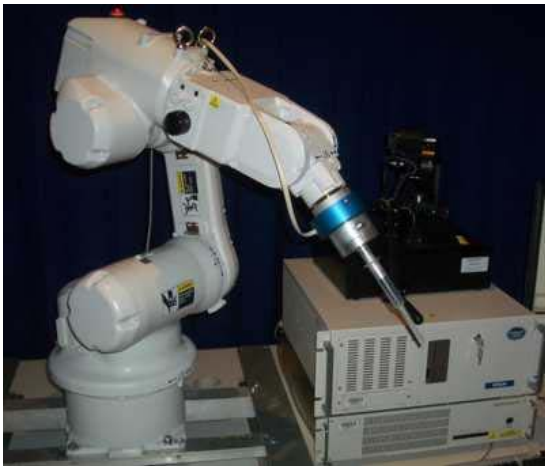

Figure 3. Epson Pro Six fitted with JR3 force/torque sensor ready for a haptically controlled needle insertion task.

5.2 Activmedia Platforms

The ActivMedia series of robot's are ideally suited for remote haptic control. The PowerBot, shown in Figure 4 is an extremely manoeuvrable, high payload, high speed robotic mobile platform. This platform has a high degree of computational intelligence, and the ability to transport a payload of up to 100kg. This allows the robot to be used for applications like area monitoring, reconnaissance tasks and material handling. With its built in vision and 3D stereo object recognition systems, a 6 DOF (Degree of freedom) 2 kg payload manipulator, this platform is ideal for remote haptic control. PowerBot has a wide array of onboard sensors and communication devices, including 28 sonar sensors, WiFi, front and rear bumpers, a compas and tilt-position sensor and a SICK laser range finding scanner used for navigating and obstacle avoidance. As well as a stereo camera and a pan and tilt CCD camera for vision processing.

The addition of a haptic control scheme enhances operator control and understanding of the robot's environment. Determining if a wheel is slipping is as simple as feeling vibration accompanied by a swing to the left or right of the haptic device used for control. Similarly, acceleration and direction can be expressed as a series of force vectors to be demonstrated to the user by means of haptic representation. Two ActivMedia robot's have been integrated with the haptic server, the Powerbot and the P3-AT series. Haptic data is gathered onboard the mobile robot's using a pair of 3-axis accelerometers.

These accelerometers are placed across the drive axis at the robot's edge. The location of these sensors is important as detection of wheel slip (vibration) and incline is predominant in these areas.

Figure 4. ActivMedia Powerbot and P3-AT series robot's configured for haptic control. Note the PowerCube arm on the larger Powerbot.

Even though the ActivMedia robot's are fitted with gyroscopes, accelerometers are used for providing haptic feedback to the user because they are faster and already mounted for vibration and incline sensing.

Additional axes can be controlled with the addition of a multiplexing switch. The haptic server can receive data from one haptic device and control multiple devices albeit not simultaneously. The ActivMedia robot description utilises this capability with the attached pan/tilt cameras. Each camera is controlled via two axes on the haptic device while the 'C' key is held on the keyboard. A similar methodology enables the larger Powerbot to use its onboard 6 DOF 'PowerCube' robotic arm in addition to moving its base around the environment. Haptic feedback of roll, pitch, yaws and wheel slippage is directly scaled to the haptic device with a linear force opposing the direction of application. Additional vibration forces can be enabled so that obstacles detected via the robot's onboard sonar ring or laser sensor are detectable to a user haptically. These platforms where then tested using the software library and the algorithms developed.

6 Trials and Algorithms

In the previous sections we discussed the developed software and robot integration. In this section we evaluate our work through physical experiments with different robotic platforms.

Effective response from the Epson robot arm can be achieved by sending position points. This is because the more points sent to the robot over a greater distance of required travel increases the time it takes for the robot to move to a desired point. Figure 5 shows the time required for the robot (joint one) to move over a large rotation with varying numbers of intermediate positional points, while Figure 6 shows positional time required for a small rotation of the robot's joint (one).

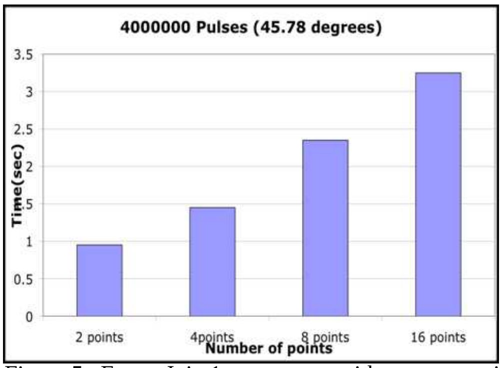

Figure 5. Epson Join 1 movement with respect to time (large rotation)

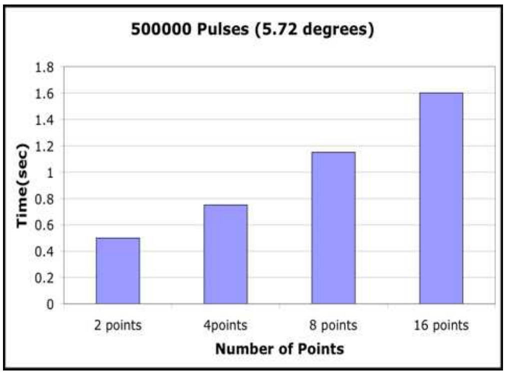

Figure 6. Epson Join 1 movement with respect to time (small rotation)

When making large movements the robot arms positioning time is greatly increased by the number of positional points in its trajectory. Working in a small area, the time response is much quicker. This is a result of the robot's trapezoidal motor controllers' acceleration settings and computational overhead. Therefore when the robot is required to move over a large distance quickly it can't afford to receive too many positions. However, when working in a small area or trying to accurately position a gripper to pick an object up, the robot needs to receive numerous positions to aid in accuracy. To solve this problem angular acceleration, angular velocity, linear velocity and linear acceleration of haptics as well as positional data needed to be gathered. Collecting the instantaneous angular and linear velocity data from the haptics device is expedited by Sensable's software, however the acceleration data needs to be calculated. To achieve this calculation accurately real time position and velocity data is required form the robot.

$$v = \frac{p_1 - p_2}{\Delta t} \quad [3]$$

$$a = \frac{(v - v_i)(v)}{(p_2 - p_1)} \quad [4]$$

Where: $P_1 = A$ in figure 8 (known), $P_2 = B$ in figure 8 (known) and $v_i =$ the initial velocity at p_1. The combination of this data with the results built a set of rules that allowed for fast response from robot's under any circumstances without sacrificing accuracy.

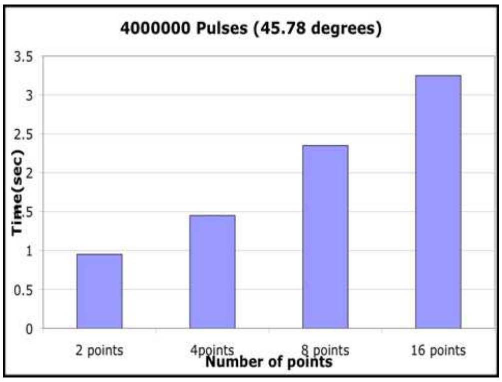

Figure 7. Number of positions V's velocity results.

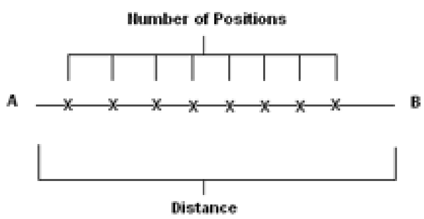

Figure 8. Velocity data capture methodology

During the development of the model for controlling the Epson robot arm, a value of two seconds was chosen to be the upper limit of time required for the robot to move 45 degrees (4M) encoder pulses. This value is based on a learning approach conducted during a number of pick and place tasks. As can be viewed in Figure 7, the maximum number of points that can be sent to the robot controller within the two second time frame is seven. When moving in a smaller work envelope, more points can be sent before the robot is slowed by data overload. Figure 7 also shows the exponential nature of the robot controller when dealing with multiple sets of data. When flooded with points the robot's response time is exponentially increased. Also of note is the minimum half-second delay in the haptic feedback loop. This is due to latency in the haptic server, transmission medium and robot controller. Data was collected by calculating the time taken for the robot arm (joint one) to move from point A to B with X number of points in-between (Figure 8). Selecting a time considered acceptable for robot response enabled the calculation of the number of positions the robot can be sent for a particular velocity. Correlating the results from Figure 7 enabled the development of an algorithm that controls the number of positions that can be sent to the robot for a calculated velocity of the haptics device. This controller uses a low pass filter algorithm as shown in Equation 5.

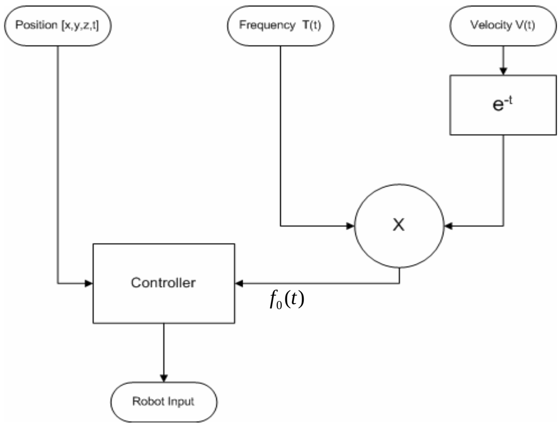

Figure 9. Haptic to robot controller

$$fo = f(t)e^{-aV(t)} \quad [5]$$

Where f(t) is the frequency of data being passed by haptics (Averaging around 1000hz) fo(t) is the frequency of data being passed to the epson robot. V(t) is the velocity of the omni haptic device.

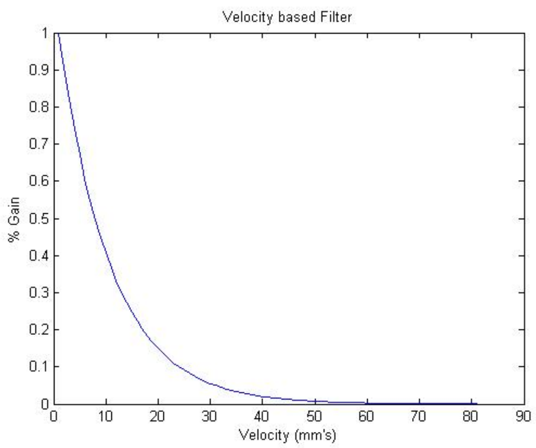

Figure 10. Controller Results

Figure 10 shows the results of the control algorithm for data rates to be passed. As can be seen as the velocity of the haptics device increases, the frequency data passed decreases exponentially to 0.

The controller acts as a scheduler between the haptic positional output and the robot positional input. This compares the desired frequency f_o with the actual frequency and schedules the positional data at the desired frequency, allowing for the linking of haptics and robot's in an open loop schema.

By comparison haptic control of robot's compared to other interfacing systems such as a keyboard or mouse has an obvious advantage of being a three dimensional (3D) input system linked to a 3D output system. Using a keyboard and mouse to control a robot suffers from all the same disadvantages that require data rate control (equation 5). Keyboards and mice are also limited in user transparency to the robot, i.e. a robot with 6 DOF is harder for a user to visualize control movements in response to mouse movements or key presses in one or two dimensions. Concurrently, the main controllability advantage haptic technology offers to robot control is force feedback. Force control feedback is beyond the scope of this paper but allows for a force feedback loop to convey more detail of what the robot is experiencing than can be experienced with a standard mouse or keyboard.

7 Conclusion

Primarily focussed on the addition of haptic technology to the robotic world, the haptic server software also allows researchers to explore teleoperation control theories and latency problems with ease. Development of an easily modifiable G.U.I and H.M.I aids in rapid programming of new control architectures and the integration of new robot's to the haptic world. Haptic technology has also been shown to add redundancy to the normal methods of visual control in a simple pick and place operation. The multi-platform nature of the haptic server will allow a diverse field of researchers to experiment with haptic technology. This is important now that haptic technology is becoming affordable and easy to integrate into many technologies.

Acknowledgements

This work is supported by the ISR: Intelligent Systems Research Labortory, at Deakin University Australia.

References

[O'Malley et al, 2003] O'Malley M. Ambrose R. "Haptic Feedback Applications for Robonaut". *Industrial Robot Journal* Volume30. Number 6 2003 pages 531–542

[GHOST, 2003] GHOST SDK Reference Manual GHOST ® SDK API Reference Version 4.0 *SensAble Technologies*, Inc. 2003.

[Dennerlein, 2000] Dennerlein, J.T. (2000), "Force-feedback improves Performance for steering and combined steering targeting tasks", *Proceedings of the Conference on Human Factors in Computer Systems*, pp.423-9.

[Williams et al, 2002] Williams, L., Loftin, R., Aldridge, H., Leiss, E. and Bluethmann, W. (2002), "Kinaesthetic and visual force display for telerobotics", *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol.2, pp.1249-54.

[Lee et al, 2002] Lee, S; Sukhatme, G.S.; Kim, G.J.; Chan-Mo Park; "Haptic control of a mobile robot: a user study", *Intelligent Robot's and System, 2002. IEEE/RSJ International Conference on*, Volume: 3, 30 Sept.-5 Oct. 2002 Pages: 2867 - 2874 vol.3

[Mullins et al, 2004] J. Mullins, H, Trinh, S. Nahavandi, "A Real Time Haptic Model Generation Technique for Surgical Needle Insertion Simulation", *Proceedings of IEEE Mechatronics and Robotics '04*. Aachen Germany, September 2004. Pages: 350-354.

[Mullins et al, 2004] J. Mullins, H. Trinh, S. Nahavandi, "Soft tissue modelling for rendering in virtual environments", *Proceedings of Mechatronics 2004*, Ankara Turkey, September 2004. Pages: 693-698.

[Colgate et al, 1995] J. E. Colgate, M. C. Stanley, and J. M. Brown "Issues in the haptic display of tool use". Proceedings of IEEE/RSJ Int'l Conf. on Intelligent Robot's and Systems pages 140 -145, 1995.

[Abbott et al, 2005] Jake J. Abbott "Virtual Fixtures for Bilateral Telemanipulation" dissatation, John Hopkins university, Augst 2005.

[Fong et al, 2000] T. Fong, F. Conti, S. Grange, and C. Baur, "Novel interfaces for remote driving: gesture, haptic and pda," in *SPIE Telemanipulator and Telepresence VII*, Boston, MA, USA, November 2000.

[Roth et al, 2002] H. Roth, K. Schilling, and O. Rosch, "Haptic interfaces for remote control of mobile robot's," in *Proceedings of 15th IFAC World Congress*, Barcelona, Spain, July 2002.

[Hill, 1979] Hill, J.W. "Study of modelling and evaluation of remote manipulation tasks with force feedback", NASA-CR-158721, SRI International Corp. for JPL, 1979

[HAVE, 2002] HAVE 2002 IEEE Int. *Workshop on Haptic Virtual Environments and Their Applications* Ottawa, Ontario, Canada, 17-18 November 2002

[Williams et al, 2000] Williams, L. Seaton, M. "Haptics-Augmented High School Physics Tutorials*" International Journal of Virtual Reality October*, 2000.

[Kress et al, 2001] Kress, G. and C. Jewitt, Ogborn, J., Tsatsarelis, C. "Multimodal teaching and learning: the rhetoric's of the science classroom". London, Continuum. 2001

[Everett et al, 1999] S. E. Everett and R. V. Dubey, "Model-based variable position mapping for telerobotic assistance in a cylindrical environment," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, May 1999, pp. 2197–2202.

[Hsu et al, 1999] L. Hsu, R. Costa, F. Lizarralde, and J. Soares, "Passive arm based dynamic positioning system for remotely operated underwater vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, May 1999, pp. 407–412.

[Hirzinger et al, 1993] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics-ROTEX and its telerobotic features," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 649–663, Oct. 1993

[Pathirana et al, 2005] Pathirana, P.N.; Black, T.J.; Nahavandi, S "Terminal approach navigation for autonomous robot's in a sensory network".; *Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on* 10-12 Aug. 2005 Page(s):552 - 557

[Elhajj et al, 2001] Elhajj, I.; Xi, N.; Wai Keung Fung; Yun Hui Liu; Li, W.J.; Kaga, T.; Fukuda, T.; "Haptic information in Internet-based teleoperation Mechatronics", *IEEE/ASME Transactions on Volume 6, Issue 3*, Sept. 2001 Page(s):295 – 304

[Ryu et al] Ryu, D., Kang, S., Kim M., Song, J., "Multi-modal User Interface for Teleoperation of ROBHAZ-DT2 Field Robot System" *IEEE/RSJ International Conference on Intelligent Robot's and Systems, Proceedings of*, September 28 -October 2, 2004, Sendal, Japan