DEVELOPMENT AND IMPLEMENTATION OF A TELEROBOTIC SYSTEM WITH VISUAL AND HAPTIC FEEDBACK: CURRENT PROGRESS

J. Pretorius[1*] and A.F. van der Merwe[2]
[1]Department of Industrial Engineering
University of Stellenbosch, South Africa
14581264@sun.ac.za

[2]Department of Industrial Engineering
University of Stellenbosch, South Africa
andrevdm@sun.ac.za

ABSTRACT

Telerobotics is a field of robotics interested in controlling robots from a distance. Incorporating visual and haptic (touch) feedback allows the operator greater accuracy in manipulating objects in a remote environment. This project endeavours to develop a telerobotic system with a focus towards telesurgical applications by using two similar industrial robotic manipulators, one acting as a haptic input device, the other as the telerobot. This paper describes the process of converting such robots into a functioning telerobotic system that allows the operator to “see” and “feel” in the remote environment. A partial working model of the telerobotic system can be achieved through experimental procedures.

*Corresponding author

1. INTRODUCTION

Telerobotics is a field of robotics interested in controlling robots from a distance. The ability to manipulate and inspect objects in a remote environment is extremely valuable, especially for tasks that pose a health risk to the human operator or that require specific skills and knowledge to perform correctly [1]. The main issue in telerobotics, though, is preventing the loss of human perception when performing tasks in a remote environment. The loss of perception limits the ability of the operator to perform the task at hand. “Perceptual” feedback, such as visual and haptic feedback is, therefore, essential for any task that requires high precision.

In the medical environment, the ability of a surgeon to haptically perceive (feel) the amount of force he or she is applying when using a surgical tool such as a scalpel is extremely important to successfully perform surgery. The importance of the sense of touch, though, is often overlooked as it is an inherent human ability. The introduction of a remote environment removes the operator’s ability to perceive “touch”. Allowing the operator to “feel” what is happening in the remote environment, therefore, poses a challenge and is currently a popular topic in literature with various approaches currently being considered.

This paper endeavours to develop a telerobotic system for high precision tasks. The efforts will be aimed towards, but not exclusively focused on, telesurgical applications where the force exerted by the operator is accurately conveyed and exercised by the telerobot. A typical procedure will involve that the operator controls a haptic device for manipulating a tool, such as a scalpel (or pen) attached to the end-effector of the telerobot, whilst monitoring the remote environment visually via video feedback on a computer screen.

2. METHODOLOGY

Based on the physical nature of the research project the research methodology follows an empirical study - starting with the analysis of existing telerobotic systems and investigating the technology available to the project. The existing technology can then be adapted to fit the requirements of a telerobotic system at which point the communication and control aspects can be developed to obtain a working telerobotic model. From there, primary data in the form of experiments will be used for improving the developed telerobotic system.

3. PROBLEM STATEMENT

Consider the telerobotic system illustrated in Figure 1. To successfully perform basic telesurgery, such as making an incision with a scalpel, the operator requires three vital aspects from a telerobotic system. Firstly, he needs an interface (haptic device) that can “capture” the action he is performing with his hand in the operator environment. Secondly, he needs perceptual feedback from the remote environment which allows him to see where he wants to make the incision (visual feedback) as well as feel the effort he is exerting (haptic feedback). Thirdly, the communication between the operator and telerobot in the remote environment needs to be in real-time. A long delay between when the operator receives the feedback and when it actually occurred could mean that he overshoots the target position or applied force - both of which could be fatal. These three requirements are necessary to ensure that the system is precise, as all surgical procedures are a delicate matter and often require high precision in the execution phase to minimize patient risk and recovery time.

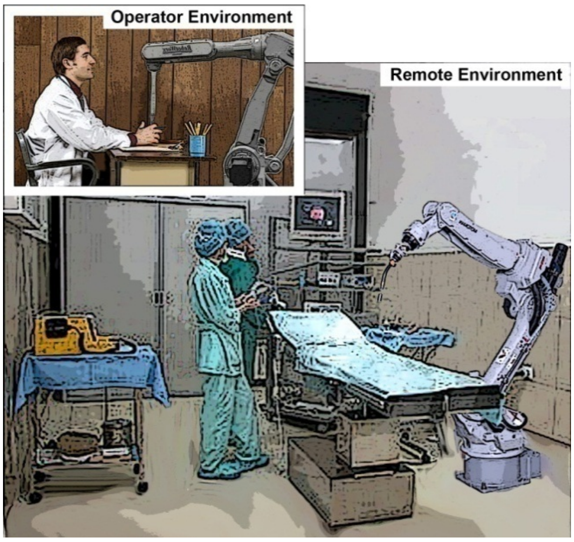

Figure 1: Illustration of Problem Statement

The project aims to accomplish these three aspects using two similar industrial robotic manipulators, one of which will be used as an operator interface (haptic device), while the other assumes the role of the patient interface (telerobot). Both are fitted with the necessary sensors and devices to allow the operator to perceive the remote environment through the sense of touch and the sense of sight.

4. SYSTEM DEVELOPMENT

Consider the telerobotic system illustrated in Figure 2. A telerobotic system constitutes two separate environments, an operator environment and a remote environment. The remote environment contains the telerobot to be controlled, along with the various sensors and devices required to provide visual and haptic feedback to the operator. The operator environment, therefore, requires devices capable of conveying these visual and haptic feedbacks in such a way as to excite the operator's sense of sight and touch respectively. Furthermore, the operator environment requires a method to send accurate control instructions to the telerobot based on the feedback from the remote environment.

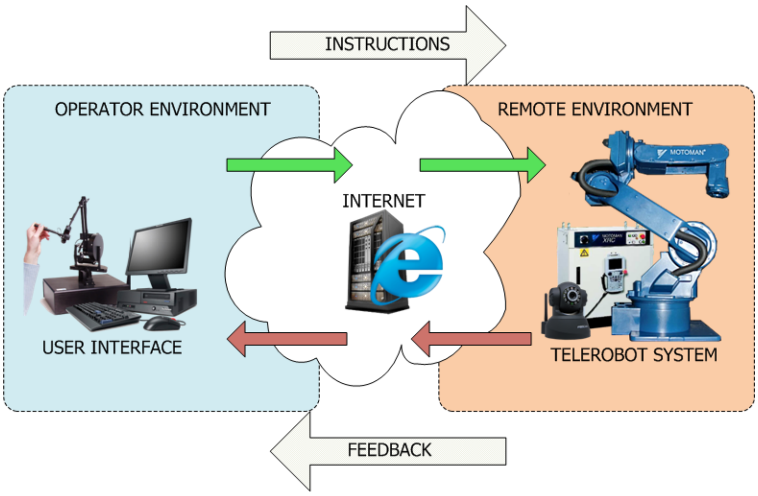

Figure 2: Telerobotic System Overview

From this illustration the primary components of a telerobotic system become apparent. Firstly, two haptic interfaces are needed, capable of sensing the telerobot and operator's haptic input respectively. Secondly, the system requires a control algorithm for manipulating the haptic data to the desired control data. Thirdly, a communication system is required capable of communicating between the various devices in both the operator and remote environments. Finally, the haptic interfaces, control system and communication system need to be combined into a telerobotic system.

4.1 Available Technology

This section provides an overview on the technology available for use in this project. These technologies form the basis of the telerobotic system with all subsequent device considerations made aimed at transforming the available devices into a telerobotic system.

4.1.1 Motoman Robots

The Motoman UP6, Figure 3(a) and Motoman SDA10D, Figure 3(d) are commercially available industrial robotic manipulators developed by Yaskawa Electric Corporation. They are six and seven degree of freedom robots respectively and each degree of freedom is actuated by its own electric servo motor [2; 3].

(a) Motoman UP6 (b) XRC controller (c) DX100 Controller (d) Motoman SDA10D

Figure 3: Motoman Industrial Robots with Controllers- adapted from [2; 3; 4; 5]

Control of the robotic manipulators is accomplished via their respective controllers. The controllers supply power to each servo motor and each contains a logic control unit, through which commands can be given to the Motoman manipulators. The XRC controller, Figure 3(b), contains three different approaches for controlling the Motoman UP6 manipulator of which only one, known as the "Host Control Functions", is capable of remote control via a serial (RS-232) interface. This allows the operator to send specific control instructions to the Motoman UP6 from a host controller, such as a personal computer [4]. The DX100 controller, Figure 3(c), makes use of similar "Host Control Functions"' to instruct the Motoman SDA10D manipulator, however, it is capable of both serial (RS-232) and Ethernet communication protocols [5].

4.1.2 Barionet Communication Device

Thee barionet is a fully programmable network-enabled controller for interfacing various devices to IP-based networks [6]. The barionet has its own high level, interpreted control language used to program the barionet called Barix Control Language (BCL). The BCL syntax is very similar to the well-known BASIC language, with various enhancements specifically for network access such as UDP, TCP and CGI communication protocols [7].

4.2 Incorporating Haptics

Haptics refers to the human sense of touch and can be subdivided into two primary components, cutaneous and kinaesthetic touch [8; 9]. Cutaneous touch refers to the human tactile perception specifically regarding pressure experienced by the skin. It enables humans to detect vibration, surface roughness, skin stretch, skin curvature, etc. A typical example of tactile perception is the ability to distinguish between different surface textures. Kinaesthesia on the other hand, refers to the sense of force in the muscles and tendons. It provides awareness of the position and motion of the human body (static and dynamic) as well as larger scale details, such as basic object shape and mechanical properties, such as hardness of materials [10].

A haptic interface consists of a robotic mechanism along with sensors to determine the human operator's motion and actuators to apply a force to the operator. This physical mechanism couples the operator to the remote environment and can take the form of a common computer gaming joystick, a wearable exoskeleton device, or as is the case in this paper, a multiple-degree-of-freedom industrial robotic manipulator.

4.2.1 Sensor Requirements

Incorporating haptics requires a haptic system capable of measuring the human hand capabilities. The hand is capable of complex manoeuvres, all of which fall into 2 main categories. It can exert a load force (pulling, pushing, pinching or grasping) and a shear force (twisting and turning). It is also capable of any combination or multiples of the two, so it is essential that the chosen sensing system be capable of measuring all of these motions in all three Cartesian coordinates.

Extensive research regarding hand/wrist force exertion [11; 12] and tactile/kinaesthetic sensing [13; 14] are available in literature. These findings are summarized in Table 1.

Human Hand Capability	Measurement
Tactile Sensing bandwidth	320 Hz
Kinaesthetic sensing bandwidth	20 to 30 Hz
Force exertion bandwidth	10 to 15 Hz
Force resolution	0.025 to 0.05 N
Finger load force (maximum)	70.84 N

Table 1: Human Hand Perception

To make the sensing system as accurate as possible, the chosen sensor(s) needs to be situated close to the manipulator's end-effector, the point of contact between the human operator's hand and the robotic manipulator, to be able to measure any force and torque applied by the operator as well as meet the human hand perceptual requirements listed in Table 1.

4.2.2 Description of the Net F/T Sensor

ATI Industrial Automation is a leading engineering-based world developer of robotic end-effectors, including multi-axis force/torque sensing systems amongst others [15]. Comparing the haptic requirements to the available models supplied by ATI, a suitable sensor has been chosen. The Net F/T Gamma (SI-130-10) sensor, shown in Figure 4, measures 6 components of force and torque (Fx, Fy, Fz, Tx, Ty and Tz) and relays the information to the Net Box at a rate of up to 7000Hz. The Net Box is equipped with multiple communication interfaces – including CAN bus, Ethernet and Ethernet/IP and can, therefore, easily be integrated into a local area network (LAN) for remote operation and monitoring [16].

Figure 4: Net F/T System: Net Box, Cable and Sensor [16]

The sensing system is capable of sampling data at a rate much higher than the tactile sensing bandwidth and is capable of measuring the full scale of forces (at an acceptable resolution) capable by the human hand for surgical procedures. A brief overview of the sensor's sensing specifications are provided in Table 2.

Sensor	Sensing Range				Resolution			
	Fx,Fy	Fz	Tx, Ty	Tz	Fx,Fy	Fz	Tx, Ty	Tz
Gamma	130N	400N	10Nm	10Nm	0.025N	0.05N	0.00125Nm	0.00125Nm

Table 2: Net F/T Gamma Specifications

4.3 Communication System Development

To communicate between two separate environments, a communication network must be in place to effectively "carry" the data between the two points. For the purposes of this research project, the universities intra-network is used. With a communication channel available to link the two environments the next step is making sure all the devices required for a telerobotic system are "network enabled". A quick look at product manuals for the available devices (multi-axis Net FT sensors, Motoman robots and network camera), indicates that the Motoman UP6 robot only has a serial interface and does not support Ethernet network interfacing. A Barix barionet 100 device will, therefore, serve as a network enabling device for the Motoman robot.

With this in mind, a communication system can be developed for intercepting sensor readings and transmitting commands to the robots via the barionet devices, as shown in Figure 5. Here the computer (Java application) is responsible for collecting the data from the sensors, processing the data into the appropriate commands and then sending them to the barionet devices. This Java application is developed in Netbeans IDE 6.8 running Java development kit 1.6. The NetBeans IDE is an award-winning integrated, open source development environment. It has an extensive community based support which allows developers to rapidly create web, enterprise, desktop, and mobile applications [17].

Communication between the different devices are established by way of Ethernet interfacing, using a UDP protocol for transmitting and receiving data, with an exception of the serial communication required by the Motoman controller (via the barionet) and the video feedback from the network camera. Data is transmitted using a UDP protocol from the Java application to the barionet device after which it is sent over a serial connection to the Motoman controller. The Vivotek PZ7151 network camera [18] is equipped with a web server hosting a website from which the video feedback can be viewed. The Java application, thus, simply makes use of a HTTP protocol for capturing the live video feed from the website and displaying it on the graphical user interface for the operator to see the remote environment.

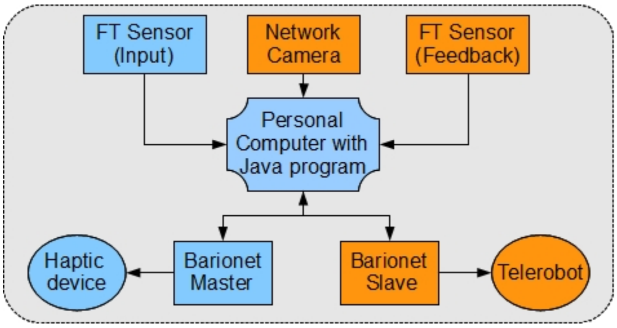

Figure 5: Java Based Communication Architecture

4.4 Control System Formulation

Several concepts regarding haptic control can be found in literature with the most common methods being position control [19], force control [20], hybrid control [21], impedance control [22] and admittance control [23].

Fundamentally, this project will make use of admittance control. Forces and torque values are measured, and are then sent to Java application. Calculations are performed to find the corresponding motion of the end point according to the equations of motion and position control approaches are used to move the Motoman manipulators accordingly. A slight variation of the traditional admittance control approach is required, though, as this project makes use of two industrial robotic manipulators each fitted with a Net FT sensing system. Haptic feedback is achieved by minimizing the force and torque error between the two sensors. The end goal of the control system is, thus, to implement admittance control with an aim at minimizing the resulting force error between the operator and remote environments.

4.4.1 Input Consideration

The coordinate system of the FT sensor remains fixed. This has a profound effect on the sensor readings for a fixed force for any change in the sensor’s orientation as demonstrated in Figure 6. While the magnitude of the force vector is measured accurately by both sensors, the force direction differs considerably as the sensor's orientation effects the individual components (Fx, Fy and Fz) of the measured force. Orientation A will, therefore, measure the bulk of the external force in its x-axis while Orientation B measures the bulk in its z-axis. Proper care must, thus, be taken to account for changes in the sensor's orientation.

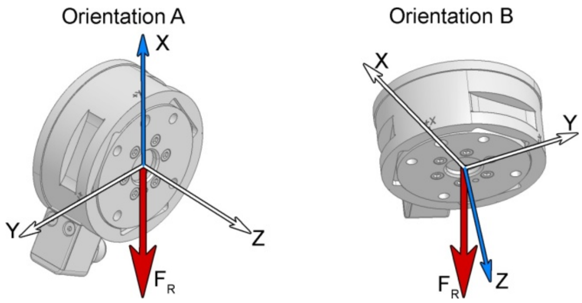

Figure 6: Sensor Orientation Effect

4.4.2 Output Consideration

Several functions are available for controlling the Motoman robots, however, only one, known as IMOV (incremental move), provides an efficient and simple approach to controlling the robotic manipulator. Traditional methods to calculate a robot's end-effector trajectory requires extensive calculations via inverse kinematics to manipulate the end-effector from its current pose (position and orientation) to its desired pose [24]. By using the IMOV function this step can be avoided. This is because the IMOV function makes use of a variable coordinate system (also known as a tool coordinate system) where the reference point is at the tip of the end-effector and the coordinate frame is always perpendicular to the end-effector's current orientation, as illustrated in Figure 6. A further advantage with the sensor mounted to the end-effector is that the coordinate frames of the sensor and end-effector remain fixed relative to each other, regardless of the current orientation of the manipulator. The resulting effect is that the individual components of the measured force (Fx, Fy and Fz) and torque (Tx, Ty and Tz) can be directly linked to a specific axis on the robotic manipulator's coordinate frame.

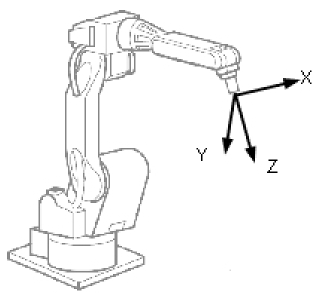

Figure 7: Tool Coordinate System

4.4.3 Control Algorithm

The function of the control algorithm is to make use of the sensor readings to calculate the appropriate command for the manipulators. This involves converting the measured "force and torque" input into a resulting "distance, orientation and speed" output. Making use of Newton's fundamental laws of motion, Equation 4.1 and Equation 4.2 respectively, the relationship between the control input and output can be determined as given in Meriam and Kraige [25].

$$\vec{F} = m\vec{a} \tag{3.1}$$

$$\vec{\tau} = I\vec{\alpha} \tag{3.2}$$

Where the external force vector, $\vec{F}$, is directly proportional to the mass, m, of the objectand its linear acceleration vector, $\vec{a}$. Similarly the external angular moment vector (more commonly known as torque), $\vec{\tau}$, is directly proportional to the objects moment of inertia, I, and angular acceleration, $\vec{\alpha}$.From these two fundamental equations of motion the desired control algorithm equations can be derived. The translational motion due to an externally applied force vector is expressed in Cartesian components, for distance (Equation 4.3) and velocity (Equation 4.4).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{F_x}{2m} \\ \frac{F_y}{2m} \\ \frac{F_z}{2m} \end{bmatrix} t^2 + \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{F_x}{m} \\ \frac{F_y}{m} \\ \frac{F_z}{m} \end{bmatrix} t + \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{4.4}$$

Where, m, is the mass of the object, t, the duration of the applied force, $\dot{x}_0$, $\dot{y}_0$, and $\dot{z}_0$, the initial object velocities in the x, y and z coordinate respectively. Similarly the angular motion due to an externally applied torque is expressed in Cartesian components for orientation (Equation 4.5) and angular velocity (Equation 4.6).

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{2I_{xx}} \\ \frac{\tau_y}{2I_{yy}} \\ \frac{\tau_z}{2I_{zz}} \end{bmatrix} t^2 + \begin{bmatrix} \dot{\theta}_{x_0} \\ \dot{\theta}_{y_0} \\ \dot{\theta}_{z_0} \end{bmatrix} + \begin{bmatrix} \theta_{x0} \\ \theta_{y0} \\ \theta_{z0} \end{bmatrix} \tag{4.5}$$

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{I_{xx}} \\ \frac{\tau_y}{I_{yy}} \\ \frac{\tau_z}{I_{zz}} \end{bmatrix} t + \begin{bmatrix} \dot{\theta}_{x_0} \\ \dot{\theta}_{y_0} \\ \dot{\theta}_{z_0} \end{bmatrix} \tag{4.6}$$

Where, I_{xx}, is the mass moment of inertia about the x-axis, t, the duration of the applied torque, $\dot{\theta}_{x_0}$, $\dot{\theta}_{y_0}$, $\dot{\theta}_{z_0}$, the initial angular velocities about the x-, y- and z-axis respectively.

4.4.4 Sensor Fusion

Incorporating haptic feedback requires that the F/T Sensor on the operator interface (haptic input sensor) be the primary driving force of the two manipulators while the F/T Sensor on the telerobot (feedback sensor) provides resistance in the event of an opposing force or torque in the remote environment. This effectively means that the telerobotic system remains stationary regardless of the feedback sensor readings. Only the haptic input readings can control the telerobotic system and only if these reading are higher than the corresponding feedback sensor readings. These limitations are formulated in Equation 4.7.

$$R_i = \begin{cases} A_i & \text{if components } A_i \text{ and } B_i \text{are aligned} \\ A_i + B_i & \text{if components } A_i \text{ and } B_i \text{ are opposing, with } |A_i| > |B_i| \\ 0 & \text{if components } A_i \text{ and } B_i \text{ are opposing, with } |A_i| < |B_i| \end{cases} \tag{4.7}$$

With, R_i, is the resulting force or torque input, A_i, the haptic input reading, B_i, the feedback reading and i representing the x, y or z Cartesian component of the force (or torque) being considered.

4.5 Harmonizing the System

A Telerobotic system, requiring real-time control and feedback, requires that the developed subsystems, be capable of performing their functions simultaneously. The Java interface must, therefore, be able to read the force and torque data from both the haptic and feedback sensors, calculate the appropriate command, send these commands to the robot via the barionet devices whilst continually displaying video feedback of the remote environment on a graphical user interface as shown in Figure 8.

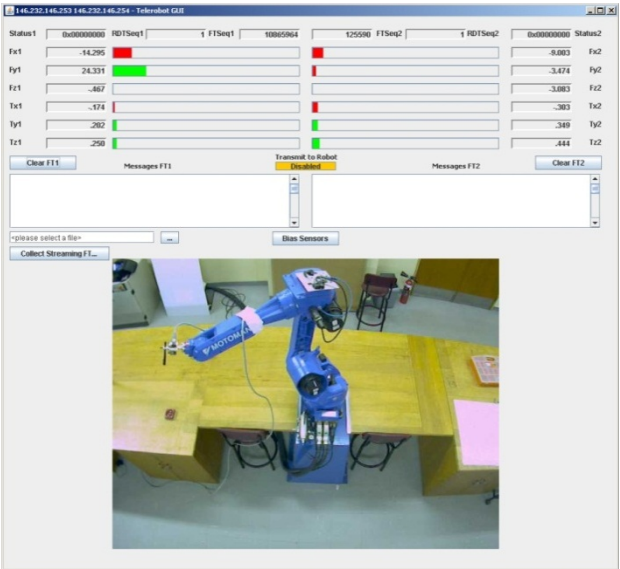

Figure 8: Graphical User Interface - adapted from [26]

To accomplish this, the Java application is divided into several smaller processes (known as threads) run by a single processor. A Java application would usually execute code sequentially, however, when using multiple threads (multi-threading) several bits of code can be executed concurrently. The diagram in Figure 9 illustrates the resulting process used to achieve communication and control of the telerobotic system in real-time.

Each branch indicates a separate thread. The main thread (left most branch) initializes the Java application and displays the GUI. After which it is responsible for the control system algorithms and sending the calculated robot commands to the relevant barionet devices. The second and third thread continually receives data from the haptic and feedback F/T sensors via UDP protocols respectively, each then updating the GUI and passing the formatted data to the control system. The final thread deals with the video feedback algorithms, by retrieving the video feed from the network camera's website and displaying it on the GUI. The figure also indicates the single process running on the barionet device – which is to receive data (robot command) via UDP and then passing it on through serial communication to the Motoman controller. A response is then sent back to the Java environment to indicate that the barionet is ready to receive the next robot command.

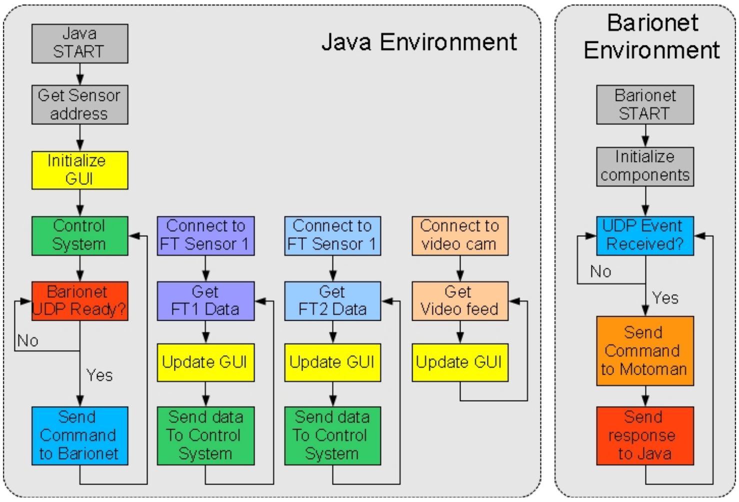

Figure 9: Telerobotic System Process Flow

5. EXPERIMENTATION

This section details the initial experiments performed on the telerobotic system. These tests are carried out before the second robotic manipulator (Motoman SDA10) is available for the project. As such haptic feedback evaluation is not yet considered. The experimental procedures discussed in this section, therefore, aims at evaluating robot motion (excluding torque) of the available Motoman UP6 robot-specifically with regard to real-time haptic control by a human operator based on the operator's haptic and visual judgment.

5.1 Motion Evaluation Methodology

The ISO 9283 International Standard is commonly used to evaluate industrial robotic manipulators as it provides a clear methodology and simplistic approach – as all concerning motion attributes can be evaluated on a single test plane [27]. The diagonal test plane is illustrated in Figure 11 and allows the evaluation of all three Cartesian coordinates simultaneously. On this test plane the command points (P1 to P5) and connecting paths can be specified as illustrated in Figure 10.

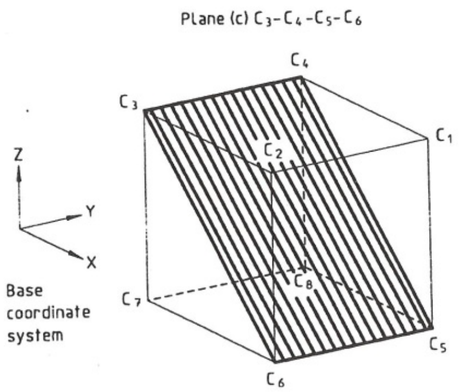

Figure 11: ISO 9283 Test plane [28]

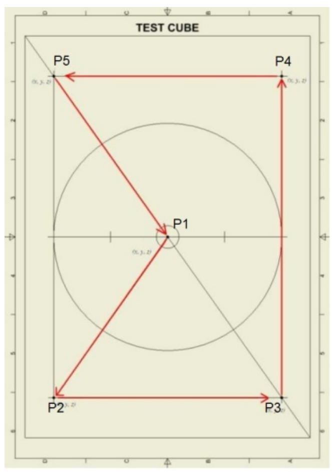

Figure 10: Command Points

In order to evaluate robot motion it is first required to define these command points. This is accomplished by using the Motoman supplied handheld programming pendant to manipulate the end-effector to the desired position. At this position, the Cartesian coordinates (referred to as the Command coordinates) can be recorded from the display of the programming pendant.

In order to do the position evaluation of the robot, one F/T Sensor is used. The F/T Sensor, referred to as the haptic input sensor, is attached to the end-effector on the Motoman UP6 robot and is used to apply the operator's input force. The second F/T sensor's readings are set to zero to prevent the sensor from influencing the positional control of the manipulator. As illustrated in Figure 10, starting form point, P1, the operator must move the end-effector by hand to points P2, P3, P4, P5 and back to P1. For every arrival at each point, the Cartesian coordinates are recorded. These recorded coordinates are the attained points. Positional accuracy is then calculated by comparing the attained point coordinates to the command point coordinates using Equation 5.1 [28].

$$AP_p = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2} \quad (5.1)$$

With x_c, y_c and z_c the command point coordinates of the point being evaluated, whilst $\bar{x}$, $\bar{y}$ and $\bar{z}$ are the mean values of the attained points, calculated according to Equation's 5.2 through 5.4. In these equations, x_j, y_j andz_j represent the coordinates for the j-th attained point, with n being the total number of cycles measured.

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \quad (5.2)$$

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j \quad (5.3)$$

$$\bar{z} = \frac{1}{n}\sum_{j=1}^{n} z_j \quad (5.4)$$

5.2 Initial Results

Using the methodology explained in the previous section the command points are measured by recording the Cartesian coordinates from the programming pendant display. A human subject, without prior training in using the interface, is used to perform the experiment. After performing 15 cycles of moving the end–effector to points P1 to P5, recording each coordinate along the way, Equations 5.1 through 5.4 have been implemented and the results are summarised in Table 3.

Point	Positional Accuracy [mm]
P_1	4.261
P_2	3.120
P_3	2.422
P_4	4.689
P_5	4.064
Average	3.711

Table 3: Preliminary Results for Positional Accuracy

These results indicate that, on average, an untrained operator successfully manages to move the haptic input device (Motoman UP6 robot), by applying a force to its end-effector, to within 3.711mm of the desired position. Note that currently these results are not statistically assured. They are based on a "pilot run" and serve merely to validate that the current developed system is functioning in the correct manner. The inclusion of the second robotic manipulator along with further refinement is required before thorough experimentations can be carried out.

6. CONCLUSION

This paper describes the initial development of a telerobotic system of which the efforts are aimed towards, but not exclusively focused on, telesurgical applications. In the medical environment, the ability of a surgeon to haptically perceive (feel) the amount of force he or she applies when using a surgical tool such as a scalpel is extremely important to successfully perform surgery. To achieve this haptic perception from a remote environment, the human operator requires a sensor capable of measuring the force applied by the telerobot as well as a haptic device capable of exerting that measured force back to the operator. A haptic device consists of a robotic mechanism along with sensors to determine the operator's motion and actuators to apply a force to the operator.

This project makes use of two similar industrial robotic manipulators, each fitted with a multi-axis force and torque sensor. One of which will be used as a haptic device, while the other assumes the role of the telerobot.

A partial "working" model of the telerobotic system has been achieved. The haptic input device (Motoman UP-6 robot) can be controlled through force control by using a FT sensor mounted to the end-effector of the manipulator. The force and torque readings are captured and sent to a Java application on a remote computer via a UDP protocol. The Java application then transforms the received data into a robot command by implementing the control algorithms. This command is then sent via the barionet communication device to the controller of the Motoman which manipulates the robot in the appropriate fashion.

Experimental procedures have been put in place, primarily to identify current issues with the initial development while waiting for the second robotic manipulator to become available. These procedures are aimed at determining the accuracy by which the operator can move the Motoman manipulator to pre-determined points by applying a force to the end-effector of the manipulator. Preliminary results indicate that an untrained operator is capable of moving the robotic end-effector to within 3.711mm of the desired point.

7. REFERENCES

[1] **Thompson, J.M., Ottensmeyer, M.P., Sheridan, T.B.** 1999. Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with localmanipulative assistance. *Telemed Journal*, 5(1), pp 129–137.

[2] **Motoman**. 2011. *Motoman up6 specifications* (data sheet). [Online]. Available at: www.micromech.co.uk/dir_products/pdf/motoman/up6-series_robot.pdf

[3] **Motoman**. 2011. *Motoman sda10 specifications* (data sheet). [Online]. Available at: www.motoman.com/datasheets/SDA10D.pdf

[4] **Motoman**. 2011. *Motoman xrc specifications* (data sheet). [Online]. Available at: www.motoman.com/datasheets/XRC%20Controller.pdf

[5] **Motoman**. 2011. *Motoman dx100 specifications* (data sheet). [Online]. Available at: www.motoman.com/datasheets/DX100%20Controller.pdf

[6] **Barionet**. 2011.*Barionet 100 / barionet 50 user manual*. [Online]. Available at: http://www.barix.com/downloads/Barionet_Family/51/

[7] **Barionet**. 2011.*Barix control language (bcl) programmers manual*. [Online]. Available at: http://www.barix.com/downloads/Barionet_Family/51/

[8] **Smith, C.M**. 1997. Human factors in haptic interfaces. *Crossroads*, 3(1), pp 14–16

[9] **Wall, S.A. and Harwin, W.** 2001. A high bandwidth interface for haptic human computer interaction. *Mechatronics*, 11(4), pp 371 – 387.

[10] **Klomp, F.** 2006.*Haptic control for dummies: An introduction and analysis*. Master's thesis, Eindhoven University of Technology.

[11] **Astin, A.** 1999. *Finger force capability: measurement and prediction using anthropometric and myoelectric measures.* Master's thesis, Faculty of the Virginia Polytechnic Institute and State University.

[12] **Hallbeck, M.** 1994. Flexion and extension forces generated by wrist-dedicated muscles over the range of motion, *Applied Ergonomics*, 25(6), pp379–385.

[13] **Shimoga, K.** 1992. Finger force and touch feedback issues in dexterous telemanipulation. *In: Intelligent Robotic Systems for Space Exploration*. Proceedings, Fourth Annual Conference. pp. 159 –178.

[14] **Sutter, P.H., Iatridis, J. and Thakor, N.** 1989. Response to reflected-force feedback to fingers in teleoperation. *In: Space Telerobotics.* Proceedings, NASA Conference on, pp 157–66.

[15] **ATI Industrial Automation, Inc**. 2011. *ATI Industrial Automation: Robotic End Effectors and Automation Tooling*. [Online]. Available at: http://www.ati-ia.com

[16] **ATI Industrial Automation, Inc**. 2011. *ATI Industrial Automation: Net F/T:* [Online]. Available at: http://www.ati-ia.com/products/ft/ft_NetFT.aspx

[17] **Netbeans**. 2011. *Netbeans ide 6.8 release information*. [Online]. Available at: http://netbeans.org/community/releases/68/

[18] **VIVOTEK, Inc.** 2011. *Network camera user's manual.* [Online]. Available at: http://www.vivotek.com/products/model.php?network_camera=pz7151

[19] **Fite, K. and Goldfarb, M.** 1999. Position control of a compliant mechanism based micromanipulator. In: *ICRA'99*, pp 2122–2127.

[20] **Volpe, R. and Khosla, P.**1993. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(1), pp1634–1650.

[21] **Raibert, M.H. and Craig, J.J.** 1981. Hybrid position/force control of manipulators. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme,* 103(2).

[22] **Hogan, N.** 1985. Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. *ASME Transactions Journal of Dynamic Systems andMeasurement Control B*, 107(2), pp 1–24.

[23] **Seraji, H.** 1994. Adaptive admittance control: An approach to explicit force control in compliant motion. In: *ICRA'94*, pp 2705–2712.

[24] **Lloyd, J.E. and Hayward, V.** 1988. Kinematics of common industrial robots. *Robotics and Autonomous Systems*, 4(1), pp 169–191.

[25] **Meriam, J.L. and Kraige, L.G.** 2003. *Engineering Mechanics: Dynamics*. John Wiley and Sons, Incorporated.

[26] **ATI Industrial Automation, Inc**. 2011. *Self-contained Demonstration Application.* [Online]. Available at:http://www.ati-ia.com/products/ft/ft_NetFT.aspx

[27] **Brussel, H.V.** 1990. Evaluation and testing of robots. *CIRP Annals - Manufacturing Technology,*39(2), pp 657 – 664.

[28] **Standard, International**. *Manipulating Industrial Robots - Performance Criteria and Related Test Methods, ISO 9283.*