Development of a telerobotic test bench system for small-field-of-operation bilateral applications with 3D visual and haptic (kinaesthetic) feedback.

by

André Smith

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering Management at Stellenbosch University

Department of Industrial Engineering,
Stellenbosch University,
Private bag X1, Matieland 7602

Supervisor: Dr. A. F. van der Merwe

April 2014

Declaration

By submitting this assignment electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ....

Abstract

Development of a telerobotic test bench system for small-field-of-operation bilateral applications with 3D visual and haptic (kinaesthetic) feedback.

A. Smit

Department of Industrial Engineering,
Stellenbosch University,
Private bag X1, Matieland 7602

Assignment: MScEng(EngMan)

December 2013

Teleoperation as a field has seen much change since its inception in the early 1940s with Dr. Raymond Goertz producing the first teleoperation system for manipulating radioactive materials. With advances in core and supporting technologies, the systems have grown in complexity and capability, allowing users to perform tasks anywhere in the world irrespective of physical distance. The feasibility of such systems has increased as the drive for use of telepresence robots, exploration robots as in space exploration, search and rescue robots and military systems such as UAVs and UGVs gain popularity.

This prompted the development of a proof of concept modular, user centred telerobotic system. The current project is the second iteration in the development process.

Teleoperation and more specifically telerobotic systems pose a challenge for many system developers. This may be a result of complexity or the wide assortment of knowledge areas that developers must master in order to deliver the final system. Developers have to balance system usability, user requirements, technical design and performance requirements. Several developmental process models are considered in context of Engineering Management (EM). A larger Systems Engineering developmental process is used, with focus on the primary and supportive EM components. The author used a hybrid developmental model that is user focussed in its approach, the User-Centred Systems Design (UCSD) methodology was adopted as the primary model for application within the two distinct developmental categories. The first category hardware and system integration utilised the UCSD model as is. The second - Software development - relied on the use of agile models, rapid application development (RAD) and extreme programming (XP) were discussed with XP being chosen as it could easily incorporate UCSD principles in its development process.

Hardware systems development consisted of mechanical design of end-effectors, configuration management and design, as well as haptic and visual feedback systems design for the overall physical system. Also included is the physical interface design of the input (master) cell. Further software development was broken into, three sections, the first and most important was the graphical user interface, haptic control system with kinematic model and video feedback control.

The force following and matching characteristics of the system were tested and were found to show an improvement over the previous implementation. The force magnitude error at steady state was reduced by 10%. While there was a dramatic improvement in system response, the rise time was reduced by a factor 10. The system did however show a decrease in angular accuracy, which was attributed to control system limitations.

Further human-factor analysis experiments were conducted to test the system in two typical use-case scenarios. The first was a planar experiment and the second a 3D placement task. The factors of interest identified were field-of-view, feedback vision mode, and input modality. Heuristic performance indicators such as time-to-completion and number of collisions for a given task were measured. System performance was only showed significant improvement when used with haptic control. This shows that the research into haptic control systems will prove to be valuable in producing usable systems. The vision factor analysis failed to yield significant results, although they were useful in the qualitative systems analysis.

The feedback from post-experimentation questionnaires showed that users prefer the Point of View as a field of view and 2D viewing over 3D viewing, while the haptic input modality was preferred.

The results from the technical verification process can be used in conjunction with insights gained from user preference and human-factor analysis to provide guidance for future telerobotic systems development at Stellenbosch University.

Uittreksel

Die ontwikkeling van 'n klein werksarea telerobotiese toetsstelsel vir bilaterale toepassings, met 3D visuele en haptiese (kinestetiese) krag terugvoer.

A. Smit
Department Bedryfsingenieurswese,
University Stellenbosch,
Privaat Sak X1, Matieland 7602

Werkstuk: MScIng(IngBes)
Desember 2013

Telewerksverigting as 'n gebied het al vele veranderinge ondergaan vandat die eerste stelsels deur Dr. Raymond Goertz geimplementeer was in die vroeë 1940s vir die hantering van radioaktiewe materiale. Met vordering in kern en ondersteunende tegnologieë, het die telewerksverigtingstelsels toegeneem in kompleksiteit asook gevorder in vermoeënsvaardigheid, wat gebruikers in staat stel om take te verrig vanuit enige plek op aarde, ongeag die fisiese afstand wat die gebruiker en die werksarea skei. Die lewensvatbaarheid van hierdie stelsels het ook toegeneem weens die belangstelling in teleteenwoordigheid-robotte, ruimtevaardige-robotte, reddings-robotte en militêre-robotte soos onbemande-lug-voertuie (OLV) en onbemande-grond-voertuie(OGV).

As gevolg van die belangstelling in telerobotiese stelsels is die ontwikkeling van 'n modulêre, gebruikers-gesentreerde telewerksverigting stelsel onderneem. Die huidige projek is 'n tweede iterasie hiervan.

Telewerksverigting, en meer spesifiek, telerobotika stelsels ontwikelling, vereis dat stelsel-ontwikkelaars 'n verskeidenheid kennisareas bemeester. Die ontwikkelaar moet 'n belans vind tussen gebruiker vereistes, bruikbaarheid asook tegniese ontwerp en prestasie vereistes. Menigde ontwikkelingsproses modelle is oorweeg en behandel in die konteks van Ingenieursbestuur (IB). 'n Stelselsontwikkeling proses is gevolg met 'n fokus op primêre en ondersteunende IB komponente. 'n Gemengde ontwikkeling is toegepass tot die projek wat die gebruiker as 'n hoof komponent van die stelsel in ag neem. Die oorhoofse ontwikkelingsmodel is die *User-centred Systems Design* (UCSD) proses, wat vir beide hardeware en sagteware ontwikkeling gebruik is.

Vir die hardeware ontwikkeling is die UCSD toegepas soos dit uiteengesit is in die literatuur. Die sagteware ontwikkeling is voltooi met behulp van ratse metodes, "*Rapid Application Development*" RAD en "*Extreme Programming*" (XP) was oorweeg en XP

was gekies as ontwikkelingsmodel. XP was die natuurlike keuse weens die gemak waarmee UCSD metodes en prinsiepe kon geinkorporeer word in die ontwikkelings proses.

Hardeware onwikkeling het bestaan uit meganiese ontwerp, manipulasiegereedskap ontwerp, konfigurasie bestuur en ontwikkeling asook haptiese en visueleterugvoer stelselsontwerp van die fisiese stelsel insluitend die fisiese koppelvlakontwerp van die meester sel. Verder is sagtewareontwerp opgedeel in 'n haptiesebeheerstel met 'n kinematiese model ontwikkeling, videoterugvoerbeheer en gebruikersintervlak ontwerp.

Die vermoëe van die stelsel om krag insette na te boots was verbeter met 'n gestadigde verbetering van 10%. Die reaksietyd van die stelsel is verbeter met 'n faktor van 10. Die stelsel het 'n verswakking getoon in die algehele hoekakkuraatheid, die oorsprong van die verswakking kan aan die beheerstelsel teogeken word.

Verdere menslikefaktoranalise eksperimente is voltooi om die stelsel in twee tipiese gebruikgeval scenario's te toets. Die eerste, 'n platvlak-eksperiment en die tweede 'n 3D plasingingstaak eksperiment. Die faktore van belang is ïdentifiseer as, visie-veld, terugvoervisie modus en insette modaliteit. Heuristiese prestasie-aanwysers soos tyd-tot-voltooiing en die aantal botsings vir 'n gegewe taak is gemeet. Stelselprestasie het slegs aansienlike verbetering getoon wanneer die stelsel met die haptiesebeheer modus bedryf word. Die visiefaktor ontleding het geen noemenswaardige resultate opgelewer nie.

Terugvoervorms was na elke eksperiment voltooi. Vraelyste het getoon dat gebruikers die oogpunt van 'n lae hoek en 2D video oor 3D video verkies, terwyl die haptic beheer modaliteit verkies word.

Contents

List of Figures

List of Tables

Nomenclature

Variables

DR	Data Rate	[bps]
FR	Frame Rate	[fps]
R	Resolution (W x H)	[pixels]
DOF	Degrees of Freedom	[unit]

Acronyms

ANOVA	Analysis of variance
API	Application Programming Interface
BCL	Barix Control Language
BW	Black and White
C	Colour Properties
CCTV	Closed-circuit television
CPU	Central Processing Unit
DOF	Degrees of Freedom
DR	data rate
EM	Engineering Management
EOT	end of transmission
EOU	ease-of-use
FF	Full-field
FOV	Field of View
FR	frame rate
GUI	Graphical User Interface
HCI	Human Computer Interaction
HMD	Head Mounted Display
HRI	Human Robot Interaction
HSD	honestly significant difference
HTML	Hypertext Markup Language
IP	Internet Protocol
ISS	International Space Station
LAN	Local Area Network

OR	Operations Research
PC	Personal Computer
PI	Proportional, Integral
PLC	Programmable Logic Controller
POC	proof of concept
POV	Point of View
PRIMUS	Program of Intelligent Mobile Unmanned Systems
PTZ	Pan / Tilt / Zoom
R	resolution
RAUVI	Reconfigurable Autonomous Underwater Vehicle for Intervention
ROV	Remotely Operated Vehicle
SA	Situational Awareness
SARGE	Surveillance And Reconnaissance Ground Equipment
SDK	Software Development Kit
SE	Systems Engineering
SFO	small-field-of-operation
SOT	start of transmission
SRMS	Shuttle Remote Manipulator System
TCP/IP	Transmission Control Protocol/Internet Protocol
TTC	Time-to-completion
UAV	Unmanned Aerial Vehicle
UCSD	User-centred System Design
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
UI	user interface
WAN	Wide Area Network
WSRS	Wilcoxon signed rank sum
XP	Extreme Programming

Chapter 1

Introduction

Telerobotics and resultant technologies are becoming more widely applied in everyday life. As a result complex electronic systems, such as telerobotic systems, have become more feasible due to developments in core and supporting technologies. There will always be a need for human decision making and control, be it motivated by task complexity, moral decisions such as a kill order in the case of attack drones or the lack of observational and situational awareness or analysis on-the-fly - where the cost and time associated with the system required to complete such tasks are not feasible when compared to utilising humans.

The aim of this project is to develop a bilateral haptic telerobotic test bench that can be used for a wide variety of haptic and general telerobotic studies, making use of industrial robots, modular software and flexible vision systems. As a case study the effects of vision mode on human operator performance will be studied for the given system. The study can be used to compare the system with other systems and may potentially highlight system shortcomings and areas in which it excels.

The case study which was conducted is to assess the influence of human-factors on teleoperator performance in the presence of haptic feedback control. The study focuses solely on small field-of-operation scenarios, for use in bilateral robotic system(s). In order to assess the implications of vision mode and its effect on user performance, for the developed telerobotic system, user performance will be tested under varying vision modes with and without haptic control. The interaction between input modality (haptic input vs. joystick input) and its effect on user performance will be assessed in order to aid decision making when developing application specific teleoperation systems.

The effect on operator performance of the vision and control systems plays a crucial role in assessing the viability of the various designs. This also assists in the development of short-term future development goals. Studying the effects of vision systems on user performance will help assess the basic requirements in terms of usability. The need for 3D vision will also be investigated and its effects on user performance assessed at the hand of a usability study. The project does not claim to be a comprehensive study but rather serve as a guide for future development and research at Stellenbosch University in the field of telerobotics.

To this end an existing telerobotic system will be assessed and major issues will be identified and addressed. Further a flexible vision system will be developed that would meet the requirements of the qualitative, subjective study to aid in future development. Systems development will be aimed at using standard (off-the-shelf) components with support structures in place as to reduce life-cycle costs. Moreover haptic telerobotic system interfaces that are capable of both local network and internet based control were developed and will be implemented in order to enable testing. The interfaces posses advanced video control capability for control of web-based streaming video clients, along with a physical manipulator control for point of view applications. This study also strives to streamline the decision making process during the design phase, by showing the trade-off in user performance for various vision and input modalities.

1.1 Research Question and Objectives

Research Question:

Can a usable, modular, software-based telerobotic system be developed for use in human-factor performance analysis, using existing industrial robots while incorporating haptic (force-feedback) and 3D visual feedback.

Research Objectives:

Objective:	**Chapter / Section:**
• Show the applicability of the Systems Engineering approach in the development of generic telerobotic systems test-bench.	§3
• Assess and improve the haptic control system implemented in the initial implementation, to incorporate posture control and rotation about a point for both master and slave robots.	§4.2.2
• Develop a GUI user interface for teleoperations at Stellenbosch University that uses TCP/IP and web based operations in the **C-sharp** programming language. The GUI must include bilateral haptic pose and orientation control using Human Computer Interaction (HCI) and Human Robot Interaction (HRI) principles.	§4.2.5
• Develop a flexible and scalable vision system that is easy to use, in order to compare viewing modes.	§4.2.1

Usability testing of the system:

• Complete an experimental design for the usability performance measures, along with analysis methods to assess the impact of human-factors on system performance.	§3.7
• Compare performance under 2D and 3D vision modes to facilitate decision making in system development and operation	§5.2.5
• Assess the impact of haptic feedback on user performance under **2D** vision with Point of View (POV) and Full-field (FF) vision modes	§5
• Assess the impact of haptic feedback on user performance under **3D** vision with POV and FF vision modes	§5
• Make recommendations for future development goals and testing use cases at the hand of the usability data.	§5.2.7
• Compare system technical performance between initial and current implementations, using the force following and force angle performance measurements along with response time comparisons.	§5.1
• Develop structured post-experimentation questionnaires, to assess user preference and the impact that it may have on user performance when the system is in use. Make use-case recommendations for future experimentation at the hand of user preference data.	Appendix A and §5.2.2

1.2 Limitations

Some design and implementation limitations were applicable in the completion of this project and further study. These limitations were imposed mainly due to financial and time constraints.

- The current controllers and robots must be used as is
- The haptic hardware must be used as is, although changes may be made to the control system.
- The system must first be implemented on the local university network
- Focus on small field-of-operation tasks in a work volume for which both robots have full articulation capabilities
- Consumer level products must be used to implement the peripheral systems such as the vision system and streaming solutions.

1.3 Thesis Layout

Chapter 2 — **Literature Review**

Contextualises the current research project within Engineering Management as a field, with a focus on Systems Engineering and its role in Telerobotic systems development. Further background is given on the need for teleoperation systems with a focus on telerobotic systems along with general market drivers for such systems is discussed in order to motivate further research. Core concepts such as tele-presence, control architecture , teleoperator, haptic input and the master-slave paradigm is also explained.

Chapter 3 — **Methodology**

Here the methodology followed for each section of work is explained in full. This chapter aims to shed light on the theory of the development process with a focus on User-Centred System Design (UCSD). The systems development methodology will be discussed for both software and hardware, as well as hybrid systems. Further usability engineering and analysis tools used to analyse the system performance and user preference is explained, as part of the UCSD approach, as to give the reader a full understanding of the analysis which has been chosen for the evaluation phase of the project. Further the experimental design and testing will be discussed.

Chapter 4 — **System Development**

Here the methodology of Chapter 3 is applied. Development of both the electro-mechanical hardware system as well as the software of the current system will be discussed. This will also help to clarify some system shortcomings, improvements, and architecture changes that was made in the current iteration of the Telerobotics project. The verification process will clarify some of the research aims. The human-factors analysis and procedure will be discussed and applied in context of the current research project.

Chapter 5

Results and Discussion

This chapter will also clarify the results of the quantitative and qualitative experiments conducted in Chapter 4. The results from the Technical verification process, along with force-following and controller tuning will be discussed. Further more the user centric qualitative analysis, usability engineering will be discussed and recommendations on use-case scenarios or preferred testing scenarios will be discussed and compared to the concurrent user-centric system performance analysis.

Chapter 6

Conclusions and future work

Conclusions on system performance, user preference and user-centric system performance will be discussed. This information will be invaluable in the next phase of the ongoing telerobotic development program. Recommendations on focus areas for future development, current system limitations will be identified and possible solutions will be proposed.

Chapter 2

Literature Review

Teleoperation systems development is a complex undertaking. From design to final evaluation, the management of the development process is of paramount importance, in order to ensure timely delivery of a functional and usable system. When undertaking such projects, the various fields of knowledge required to complete the development project successfully should first be considered.

Here the reader will be taken on a journey to contextualise the development process within Engineering Management (EM) and give the required background information for the various knowledge areas required. The value of the project is in the process as the project has goals inherent in (aligned with) most teleoperation systems. Although most teleoperation systems are developed to be application specific, the current project aims to develop a generic system for small-field-of-operation applications in bilateral anthropomorphic teleoperation and act as a test-bench for future development projects.

2.1 Defining Engineering Management

Engineering management as described by Ohio University Russ College of Engineering and Technology (2012), shows the focus on pragmatic thinking. It also shows the role of Engineering managers as a team leader.

> "*Engineering management is a unique educational path that specifically addresses the skills and requirements that engineers need to become better leaders and engineering team managers.*"
>
> -*Ohio University Russ College of Engineering and Technology (2012)*

There are two main schools of thought when it considering EM. The first approach is pragmatic in its application and the way it interprets EM. The focus is mainly on the management of engineering processes or teams of engineers, thus building management skills through actively engaging in engineering tasks in a management capacity. This can also be interpreted as creating an Engineering manager, which possesses the required technical knowledge for the task at hand, but is also competent in business, i.e. has received training in business. This requires the engineering manager to act to bridge the gap between engineering, technology and the field of business with sufficient leadership skills to lead a technical team of engineers while focussing the work effort on the long term business goals of the enterprise.

The second approach focuses on the application of engineering principles and techniques on business processes and/or practises in a general sense. This can be argued to be management engineering, i.e. the engineering of management tools or processes using engineering principles. Some examples of management engineering are Operations Research (OR), technology management, and decision engineering.

A useful example is decision engineering; decision engineering usually outputs a framework or model with cause-and-effect elements. This is done in order to assist in decision making. As an example, Figure 2.1.1 shows the decision framework for the decision engineering framework, which shows clearly how the elements of decision engineering fit together.

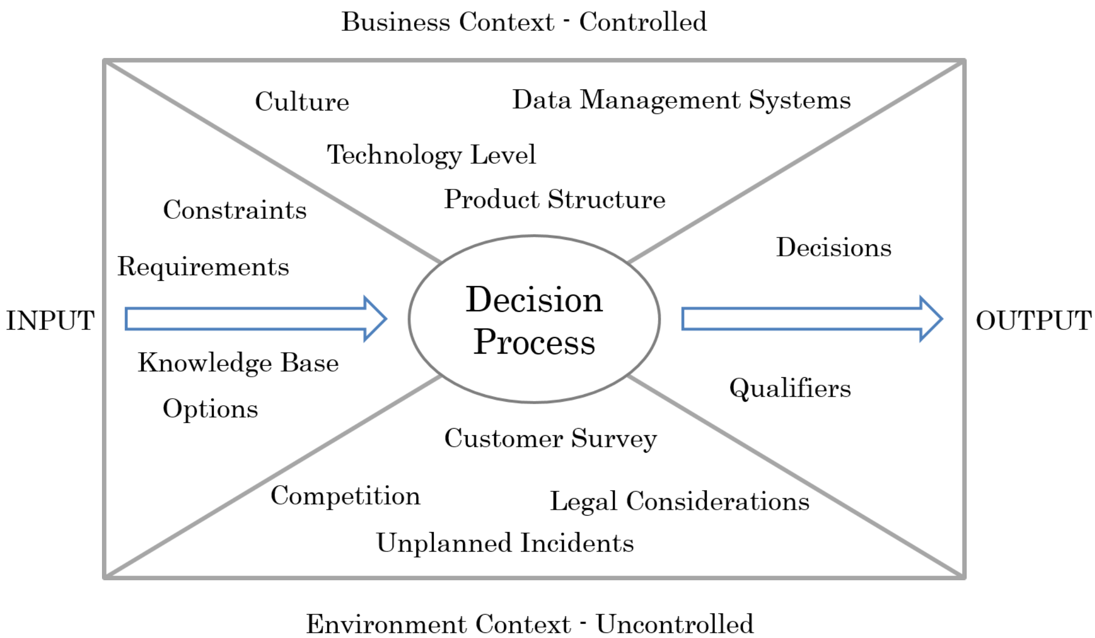


Figure 2.1.1: Decision framework, adapted from Committee on theoretical foundation for decision making in engineering design (2001).

These, at times, divergent definitions show the complexity of defining EM. The approach followed for the purposes of this project, will focus on the pragmatic approach of an Engineering Manager in the developmental process of a telerobotic system within the Systems Engineering (SE) process which be explained in more detail in Section 3.2.

2.2 Teleoperation Defined

Teleoperation constitutes any and all forms of doing work at a distance. Chellali (2009) refers to the example of using a stick to poke the fire to avoid being burned as a simple case of teleoperation. In other words, where the user (input) is separate from the work (output) by making use of some form of tool, it may be defined as being a form of teleoperation.

Teleoperation has come a long way since the early implementations. To gain a complete understanding of teleoperation one first has to consider teleoperation in a general broad frame of reference. In order to fully appreciate the complexity of these systems and the key concepts used in their development we must first consider some concepts specific to teleoperation. One of the forerunners in teleoperation research Dr. T.B. Sheridan defined teleoperation as:

> "*Teleoperations is the extension of a persons sensing and manipulation capability to the remote location. A teleoperator includes at the minimum artificial sensors, arms and hands, a vehicle for carrying these, and communication channels to and from the human operator.*"
>
> - *(Sheridan, 1989)*

This broad definition was first presented in Sheridan (1989), which is still one of the most complete and relevant articles published in the field. Although being nearly 25 years old, this definition lays a foundation for many key concepts that is still widely used in teleoperation research. These include teleoperator, user or operator, input interface and communication channel.

Sheridan's work was one of the earliest that conceptually defined a teleoperator and formalised the terminology. *Teleoperator* is one of the most important components of any teleoperation system. According to Sheridan the teleoperator is the main component of the output work environment, the definition is shown graphically in Figure 2.2.1. Irrespective of the size of the work volume, the teleoperator will require a manipulator, which is most commonly task/application specific. Further the system will require some form of feedback sensor(s), this may be visual only, tactile/haptic or audio-visual. Further, irrespective of the physical distance between the input and output, the system will require some form of communication channel to carry the information between these two environments. In Sheridan's broad teleoperator definition he mentions a vehicle to carry the other components. In most modern applications this will be a robotic system, be it stationary or mobile.

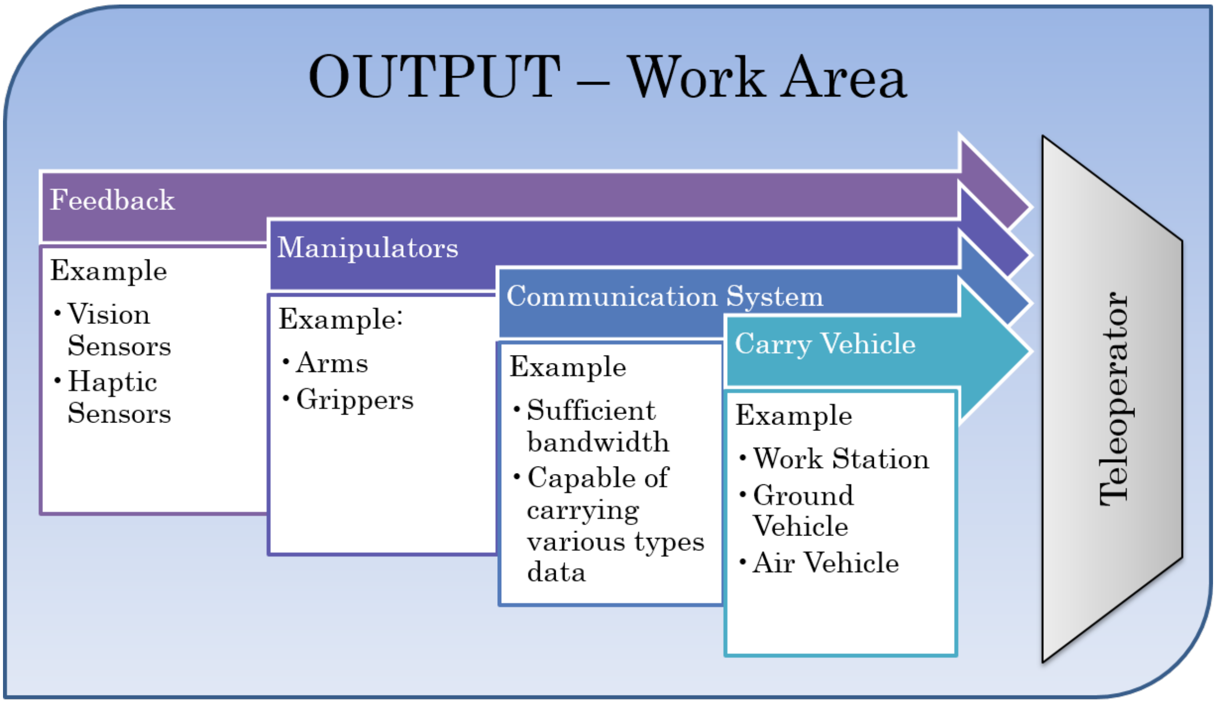


Figure 2.2.1: Key components comprising a Teleoperator

Telerobotics focuses on a scenario of teleoperation where the output is affected by an robotic unit. Within a contemporary context, telerobotics has become synonymous with teleoperation, this can be shown by considering another influential researcher which defined teleoperation specifically with reference to mechanical robotic machines, (Cui *et al.*, 2003).

> "*Teleoperation is a means to operate a* ***robot*** *using human intelligence, which requires the availability of adequate human-machine interface. A teleoperation system usually consists of two robot manipulators that are connected in such a way as to allow the human operator control one of the manipulators, which is called the master arm, to generate commands that map to the remote manipulator, which is called the slave arm.*"
>
> \- *Cui* et al. *(2003)*

This definition is far more specific in its application and clearly shows how the attitude of researchers has shifted towards teleoperation. By 2003 the base teleoperation system must have be focussed to incorporate a robotic system of some kind. Cui *et al.* (2003) once again reiterates the need for human-machine interfaces and manipulators. From this point onwards, for the purposes of this project teleoperation and telerobotics are considered to be one and the same.

The need for telerobotics could be attributed to man's constant search to reduce risk to self and to enhance human abilities. The need to improve or supersede human abilities arose from three main issues that required solutions. They are (1) *The ability to perform work in hazardous/inaccessible environments* as discussed by Hurmuzlu and Nwokah (2001) ,(2) *Physical separation* and (3) *Scaling of power/movement or both.*

> "*... teleoperators or the activities of teleoperation extend the manipulative capabilities of the human arm and hand to remote, physically hostile, or dangerous environments. In this sense teleoperation conquers space barriers by performing manipulative mechanical actions at remote sites, as telecommunication conquers space barriers by transmitting information to distant places.*"
>
> \- *Hurmuzlu and Nwokah (2001)*

The enablers for telerobotics are not easily identified. These drivers may be due to some new and emerging market drivers, or the optimisation / innovation of old applications to incorporate new technologies. This also shows that these systems are important in technology management within an enterprise, and may determine or support management strategies, where the need to reduce labour and risk are usually the main enablers for telerobotics to be applied in practice. The main enabler is large scale applications where the environment or more specifically the task environment is variable.

One of the first historical enabler, hazardous environments or inaccessible environments, was emphasised in the early work done by Dr. R. C. Goertz (1940s), for Argonne National Laboratory of the U.S. Atomic Energy Commission (Goertz, 1952), where a remotely operated system was needed for the handling of nuclear materials in hot cells. This implementation was the first telemanipulation system that can be considered as an implementation under the *modern teleoperations* paradigm. Although being rather crude and requiring direct mechanical linkages it was the first well publicised successful implementation. The first implementation was rather slow and many design changes had to be implemented in order to facilitate ease-of-use (EOU). This implementation was later revised to incorporate Closed-circuit television (CCTV) cameras and displays, thus theoretically making the distance of separation between input and output arbitrary. This highlighted key problems that are relevant even to this day, these include but are not limited to latency of video and control input and input environment design for interfaces.

The second driver was that of overcoming physical separation, this is especially applicable to modern telesurgery applications. Marescaux *et al.* (2002) showed how the expertise of a surgeon can be placed anywhere in the world via teleoperation, the first commercially available system was made available in 1992, (Wall *et al.*, 2013). Known as ROBODOC, the system was designed to assist in femur medical procedures. This is also shown by the sudden increase in Unmanned Ground Vehicle (UGV)s and Unmanned Aerial Vehicle (UAV)s.

Further telerobotics has enjoyed much attention from both researchers and industry alike in the past few decades, (Sheridan, 1995; sheng Liu and Li, 2012; lerosen4, 2012) although be it far more focussed and mostly application specific implementations. Some of these application areas will be discussed in Section 2.4. The research has focussed on the following main topics: haptic input technology, advanced control systems to compensate for variable delay, internet based systems, and supervisory control modalities to name but a few.

When these areas are advanced, the base technology available to system developers improve, they are better able to respond to market demands and solve the problems currently facing developers thus gaining competitive advantage. The main motivating factors, market and industry for the implementation of commercial telerobotic systems will be discussed in Section 2.3.

2.3 Motivating Factors for Telerobotic Implementation

Why teleoperation, why not simply implement "intelligent" automated robotics? The need for the human-in-the-loop can be motivated by considering the inherent nature of tasks, (Chen *et al.*, 2007). Tasks are variable in nature; this might be because of environmental factors for example sea currents, windy areas or moving parts. Task complexity also plays a role in the needs assessment for teleoperation, usually dictated by the application space (application specific).

Pre-programmed robotics although useful in environments that can be controlled or remain constant, as is the case in production systems, are not suited to dynamic environments that cannot be modelled fully. These systems can seldom be flexibly adjusted to a changing environment especially if the changes are random. For this reason we require human intervention, the ability of humans to perceive the environment, plan for possible changes to the environment and make decisions based on these changes cannot yet be fully achieved by robots.

Although the technology and supporting technologies have made much progress since its inception in the early 1940's, (Murphy, 1991), (Ralston *et al.*, 2001), we have yet to master the inherent problems present in these complex systems. An example of this is the advances made in video compression and web-based platforms for input to teleoperation systems (Khan and Gu, 2012).

There are many driving factors and enablers for teleoperation systems, the main driving factors are a result of physical limits of human operators or the need to reduce risk to personnel. These driving factors are summarised in Figure 2.3.1.

Other factors are compound in nature as is the case with *reduce risk to human life*. Reducing risk to human life ties narrowly with the original development drivers of telerobotic systems as was explained in Section 2.2, that of hazardous environment. In context, as a driving factor, it applies to any situation that is hazardous to human health or where safety risks far outweigh the reasonable limits. For example, search and rescue situations where buildings might be unstable or military engagement where high risk reconnaissance must be completed in hostile environments.

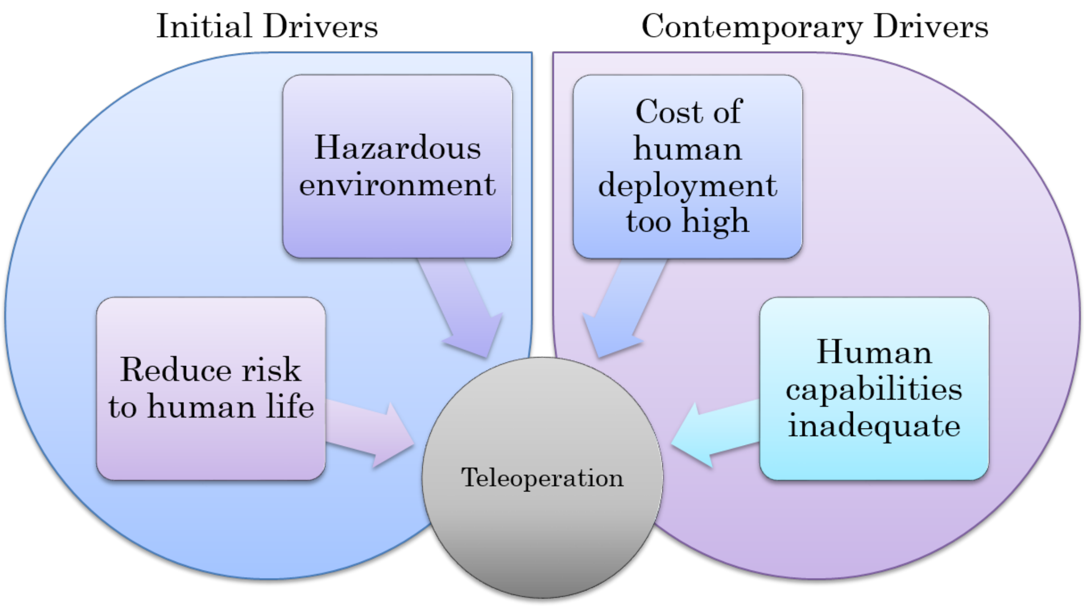


Figure 2.3.1: Drivers for telerobotic implementation

Another compound driving factor is that where *human capabilities are inadequate*. Here inadequate is used to describe any situation where normal human task completion faculties are not sufficient to complete the task. Telerobotics is especially applicable where force scaling or motion scaling is required. This can be seen as some applications that require extremely small motions such as the case in Read (2012) where fine motion was required for micro material handling. These systems can be applied in any situation where micro level motion is required; the downside of this type of system is the limits in performance are usually due to end-effector design.

Kim *et al.* (2001) showed the importance of scaling capabilities and the human interface when developing a Micro-level teleoperation system. Micro-manipulation also overlaps with micron surgery and manipulation of cells or other micro biological structures as was shown by Tanikawa and Arai (1999) in the development of a two fingered manipulation device for biological micro-structures.

Some authors (sheng Liu and Li, 2012) also name other factors that act as driving factors for teleoperation. They are uncertain tasks in hazardous and less structured environments. The tele-environment can seldom be kept constant. Performance is also hampered by a lack of human situational awareness when considering fixed vision modes.

Teleoperation is a viable alternative to conventional human interaction if the area in which or on which the task is to be performed is inaccessible, i.e. inhospitable for long-term human occupation or hazardous to human health. Although the applications of these systems are many and varied, they are currently only adopted in niche markets where financial constraints can be manipulated due to threat to human life etc.

2.4 Applications

The applications for telerobotics are nearly limitless. The need became for telerobotics became more apparent as tasks that are hazardous to human health, of great complexity on a scale that is beyond human capabilities or where the alternatives are simply to expensive to implement came up with an increasing frequency. Initial projects have been tested and implemented since the early 1940's.

Today the most common applications for teleoperations are for hazardous material handling, remote inspection and repair in deep sea oil mining and exploration as is the case with Remotely Operated Vehicle (ROV) or manipulation of objects in space exploration.

Telerobotics is an ever expanding field with new uses and applications being discovered every day. Most applications have its roots in the human need to go beyond our own limitations for example micro material handling or micro manufacturing Sato *et al.* (1994),Mitsuishi *et al.* (1996). The micro realm also has medical applications, robotics have many advantages over normal micro surgery.

As computing and communications technology becomes more powerful and bandwidth issues become a thing of the past Fabrlzio *et al.* (2000), telerobotics will affect our day to day lives more and more. For the moment complex real-time telerobotics is reserved for researchers and industry, but that will change in the future.

2.4.1 Undersea Applications

Undersea vehicles have been used for the past few years by projects such as the Argo-Jason Project Sheridan (1995). Jason is a ROV that can be used for exploration of the seabed and for inspection of deep sea structures. Similar systems have been used to explore shipwrecks such as the Titanic and have become key components of routine inspection and maintainable of deep sea cables and pipelines.

Not only do these systems have inspection and survey capabilities they are also able to complete installation tasks as highlighted as one of the key categories of ROVs, Elvander and Hawkes (2012). An economic driver for the development and implementation of these systems is that of fossil fuel demand increase and oil companies find it more feasible to explore offshore options for drilling for natural gas and oil. More recently systems such as the AQUA Explorer 2 which is used for geological survey has been launched, (Asai *et al.*, 2000).

As these systems are still rather expensive to implement, the focus of developers have moved to developing easily reconfigurable systems such as the Reconfigurable Autonomous Underwater Vehicle for Intervention (RAUVI) project developed by Sanz *et al.* (2010) which has the potential to increase the accessibility of such systems, by reducing development cost for a task specific system. Projects like the RAUVI increase the accessibility by developing generic reconfigurable robots that can be quickly changed for the needs of an specific application. There is a fine line between the feasibility of a development project and to what degree it is reconfigurable.

2.4.2 Medical Applications

The field of telemedicine has received much attention in recent years, specifically the development on robotic systems for use in telemedicine has been a hot topic. Since one of the first successful telesurgery in 2001, where a doctor was able to do a cholecystectomy on a 68-year old woman, the patient located on the other side of the Atlantic ocean. This was a major proof of the feasibility and technical viability of such systems (Marescaux *et al.*, 2002).

These systems have an appeal for military applications, where trauma victims can be treated in the field, without risking the lives of medical personnel. This also has the added benefit of instantly transporting the medical professional's knowledge and skill to the desired location in the field. An example of such a project is the "Trauma Pod" project that was awarded to SRI International in 2005, which was aimed at developing a semi- to fully automated surgical trauma response unit that will be able to cater to battlefield trauma patients, SRI International (2005).

Rayman *et al.* (2006) showed how telesurgery can be performed even with latency in control and feedback. They along with other authors promote telesurgery for use in remote locations, where specialist healthcare skills are unavailable.

Currently the da Vinci system is the most well known, as it is also the most advanced. Here the system can be classified as a teleoperation system, although the physical separation of the patient and doctor is usually in the order of meters, te system is shown in Figure 2.4.1. The system has intuitive haptic controls and stereoscopic vision systems. The system consists of a control unit and a robotic output unit. The output unit has specialised grippers and tools and allows for very fine motions and actions to be executed. The system may also be used in a wide variety of applications such as cardiology, urology and colorectal to name but a few (da Vinci Surgery, 2012; Ballantyne *et al.*, 2003).

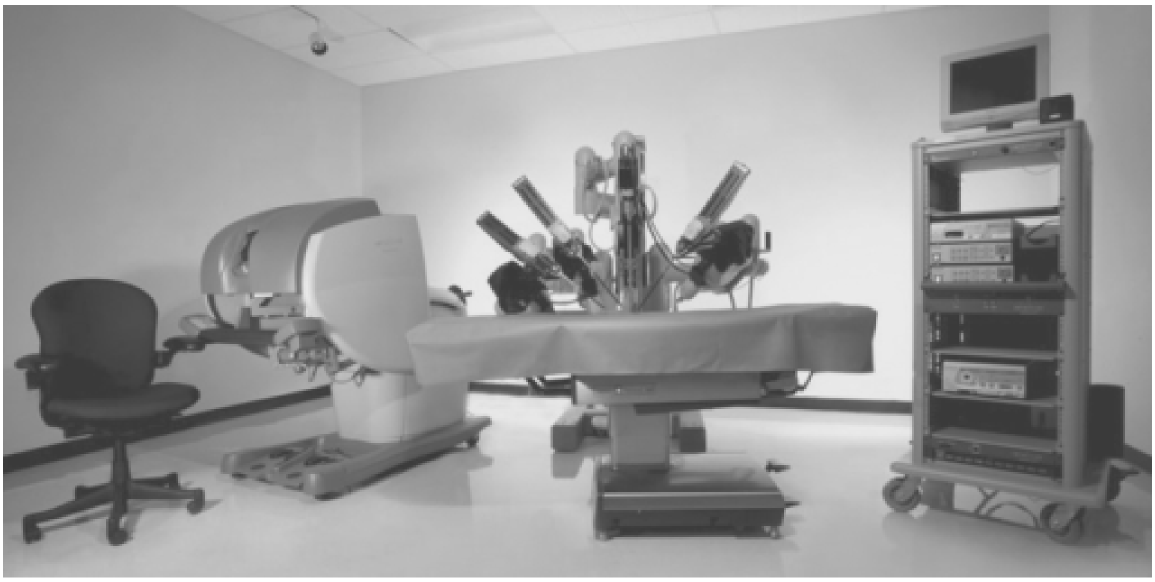


Figure 2.4.1: da Vinci, robotic surgical system, from Reynolds *et al.* (2005)

Currently these systems are receiving much attention. Development of successful teleoperated systems for medical applications has the potential to revolutionise health care. The sharing of knowledge and skills over a distance along with telesurgical systems has the potential to open international markets in healthcare while driving down cost.

2.4.3 Military and Security Applications

Military applications for telerobotics can be categorised into two main categories, land vehicles and air vehicles, commonly referred to as UGV and UAV. These implementations of teleoperation are changing the face of warfare. Teleoperated systems can full-fill many roles on the battlefield, they are:

- Reconnaissance or observation
- Support Vehicles - Supply carriers, convoy support vehicles.
- Tactical response - unmanned systems such as the Raptor UAV which has armaments on board for engagement of hostile contacts.

UGV systems, as the name suggests is a land-based system that is designed to carry equipment and NOT humans. These systems evolved from the need to reduce personnel and increase capabilities of tactical units without increasing the risk for deployed assets and risk of loss of life such as the Surveillance And Reconnaissance Ground Equipment (SARGE) system (Gage, 1995).

An example of UGV application is the Program of Intelligent Mobile Unmanned Systems (PRIMUS) project, which was aimed at removing soldiers from high risk encounters, reducing personnel and increasing the performance of semi-autonomous and autonomous unmanned systems. The project also proved the viability and set a base performance level for autonomous driving and obstacle avoidance systems (Schwartz, 2000).

These systems may be self guided or directly controlled, as discussed previously many research projects have focussed on automated guidance where the operator is in a decision making capacity for the end task. Getting to the location is the system's local responsibility - this categorises the system as a supervisory/semi-autonomous application.

The second category, that of UAVs has two main roles. The first is that of reconnaissance and the second is that of attack drones. Attack UAVs developed from reconnaissance drones, requiring larger engines to carry armaments, their range is less than that of their "spy" counterparts. They can also be used as data and radio relays (Sarris and ATLAS, 2001).

When considered in the special use-case as in military applications, human assets can be more effectively used when tasks such as navigation to a predetermined location can be automated and decisions made as a result of observations can be left to a human handler.

2.4.4 Search and Rescue and Law Enforcement

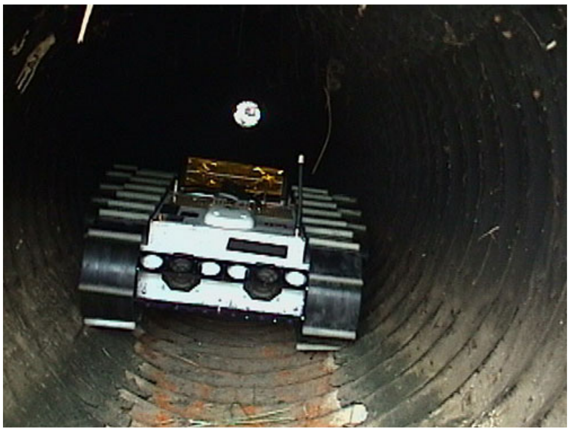
This sub-field in teleoperation has received much attention since the September 11th terrorist attacks. Here some of the first real-world applications of small remotely operated vehicles where used. The main tasks of such systems are to be inserted into small openings in rubble and be able to navigate as far as possible to look for humans that are trapped, Casper and Murphy (2003). These mobile robots vary in size, complexity and usually require only one operator.

Some robots are even able to change shape as showed by Li *et al.* (2009). Shape-shifting vastly reduces the robot's physical limitations for search and rescue scenarios. These systems have been in development for many years and are found is two basic modalities, UAV and UGV, Table 2.4.1 shows real world situations where this technology has been used successfully.

Table 2.4.1: Real-world disaster response where search and rescue robots were used, as adapted from Li *et al.* (2009)

	Disaster Deployment	UAV	UGV
2001	World Trade Center, New York, USF		Man - packable ; Man - portable
2005	California mudslides, La Conchita, USF		Man - packable
2005	Hurricane Katrina, USA	Man - packable, fixed-wing ; Man - portable, rotary wing	
2005	Hurricane Rita, USA	Maxi fixed-wing	
2005	Hurricane Wilma, USA	Maxi fixed-wing	
2006	Sago Mine, West Virginia, USA		Maxi

These disasters acted as a real-world test bed for these systems and allowed for valuable data to be collected. These systems have proven to be very effective and have helped to save lives. Some example systems are shown in Figure 2.4.2. These figures also show how these systems can overcome obstacles that might be in their path, with adaptable running gear.



(a) Example of a small robot used in World Trade Center rescue operations,from Kahney (2001)

(b) All terrain search and rescue robot, from Science Daily (2007)

Figure 2.4.2: Search and rescue robot examples

2.4.5 Space Applications

Deriving from the one of the base drivers for teleoperation, *work in hazardous environments*, one of the application areas that have received most of the attention is teleoperation in Space applications. The final frontier holds many challenges for a teleoperation system designer, not only must the system contend with temperature extremes but it must also be as light as possible (to reduce transport cost) and as durable as possible as micro-meteorites and other space debris may damage the manipulators or system controllers.

Lichiardopol (2007) identified three main categories for space teleoperation:

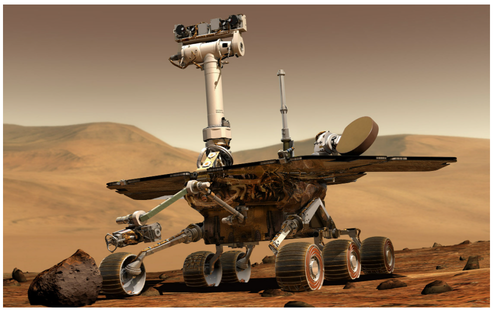
1. Satellites - probes
2. Exploration robots - rovers and probes
3. Outer-space robotic arms

Although it can be argued that satellites are not strictly teleoperation systems, probes such as the Voyager satellites still perform tasks unfeasible for humans to complete in the most hazardous of environments known to man. Satellites can be categorised under supervisory control - this teleoperations command structure will be explained in full in the following sections.

Rovers - Exploration rovers such as those sent to Mars are durable and complex. *Oppertunity* - celebrates its 8th anniversary in 2012, according to Webster (2012). These rovers (*Spirit* and *Oppertunity*) were sent to Mars to perform Geological experiments and to explore the alien planet for signs that there has been life on Mars (Jet Propulsion Laboratory, 2012). Due to the vast time delay, caused by physical distance and minimal processing and communication power of the rovers, a supervisory approach to teleoperation was followed.

Robotic Arms - Robotic arms are widely used in space applications, to reduce risk to people when compared to space walks to complete the same tasks. Robot arms were used on many space shuttle missions Shuttle Remote Manipulator System (SRMS), to launch and locate satellites and complete experiments (Nawrocki, 2003).

Next generation space exploration - This involves an ongoing research project known as Robonaut, which is the result of collaborative efforts between GM and NASA. The aim is to make space walks, which are extremely high risk, redundant by replacing the person with a synthetic humanoid which can be remotely controlled, Ambrose *et al.* (2000). Robonaut R2 was launched and activated on the International Space Station (ISS) as a semi-autonomous robot assistant. The Robonaut has the potential, in the long run, to eliminate the need to send humans into space, shown in Figure 2.4.3. These systems may help to increase humanity's presence in space.



(a) Mars Rover, from Jet Propulsion Laboratory (2012)

(b) Robonaut, from Mansfield (2005)

Figure 2.4.3: Space telerobotic examples

2.4.6 Social (Tele)Robotics

Telerobotics may have many applications in the non-professional or task driven environments. Telerobotics may be used as a tool for social interaction in work environments, where some employees can work from home and use robotic avatars / or telepresence robots in order to complete work related tasks and communicate with colleagues face-to-face.

These systems may allow for employees to work in various offices across the globe in a given day, achieving goals more effectively. This may also be used to in education to enable students from many various cultures to explore and participate in learning environments and classrooms in many countries, (Tanaka and Takahashi, 2011).

This form of telerobotics may make is possible for disabled individuals to still experience day to day life, albeit through a robotic surrogate. Some examples of surrogate telerobots are shown in Figure 2.4.4.

Figure 2.4.4: Telepresence Robots, lerosen4 (2012)

This application of telerobotics brings more concepts to the fore. These concepts deal with the feeling of presence in a remote environment and the ability to use these to improve the user experience; they are Telepresence and Immersion which will be discussed in the following section.

2.5 Telepresence and Immersion

The field of Telepresence research has given rise to many concepts, some of the more notable ones are: Immersion, situational awareness, cognitive processing. Since the early days of telerobotics researchers were aware of what is now known as telepresence. This component of telerobotics comes from the realm of social sciences. Sheridan defined telepresence as:

> ***Telepresence*** *- is the sense of being at a real location other than where one actually is."*
>
> \- Sheridan (1995)

Telepresence in general focuses on making the user/operator's experience with the system more natural. Telepresence lies at the intersection of social sciences and engineering, where social science principles and knowledge must be used in order to specify and guide the design of not only the technical system but also interaction between user and the system.

Its role in the design and implementation of input devices and environments cannot be ignored. The incorporation of the "feeling of presence" in these systems result in systems that are easier and more natural to use (Halme *et al.*, 1999). An increase in usability, the ease of use, can also be shown to influence the success of the system when commercialisation takes place.

Telepresence plays such a important role in telerobotics and teleoperation, that it has sparked an entirely separate research field. Telepresence is a key measure of modern telerobotic systems and has shown great promise for user performance as shown in Kaber *et al.* (2000), where user performance increase the larger the degree of telepresence. Telepresence for the purposes of the work done here describes the level of immersion in the remote environment. It also covers the perceived presence at the remote terminal by users/operators that interact with the terminal. This is just one of the key concepts that is central to the development of a successful telerobotic system in a larger teleoperation framework.

Contemporary systems do not yet allow for fully immersive tele-presence, but much research has been done on the effects of incorporating it in teleoperation systems, Ballantyne (2002), Kim and Biocca (1997). This showed performance improvements in the user driven capabilities of the system when there is a larger degree of immersion.

Immersion forms part of the concept of telepresence. Put simply telepresence is to make a user feel present in the remote environment. Telepresence was formally defined by Steuer (1992) by using the contrast between presence and telepresence:

> "*... 'presence' refers to the natural perception of an environment, and 'telepresence' refers to the mediated perception of an environment.* "

Further he also includes a telepresence specific component, the communication medium:

> *... the experience of presence in an environment by means of a communication medium.*

A fully immersive telepresence system or *transparent* system is defined by the user being unable to identify the boundaries of the environment the user is currently in and the remote environment. Telepresence is a key concept that must be taken into account when developing a teleoperation/ telerobotic system. The user will be given feedback from the remote environment, the more seamless and telepresence like this is done, the more effortless the human computer / robot interaction becomes.

It is logical to deduce that this is only possible where communication bandwidth is sufficient as to support lossless and real-time communication of video, audio and control signals within a telerobotics frame of reference. When developing a interface for the system, combined effects of physical input, vision and display, and audio feedback need to be considered as these all have an effect on the level of immersion and the degree of telepresence experienced by the user.

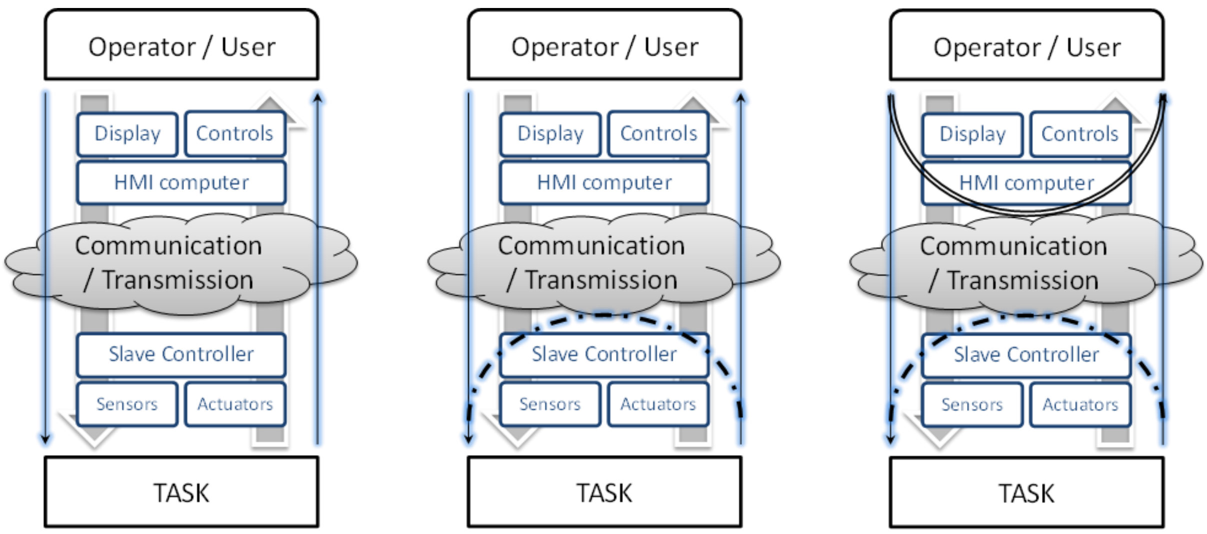
2.6 Control Architectures

When considering simple telerobotic systems one first must consider the master-slave paradigm. This model underpins the basis of most telerobotic systems. The model shown in Figure 2.6.1, shows the base conceptual components. The master-slave paradigm as the name suggests consists of two environments the one that generates the control signals and desired change characteristics (master) and the other attempts to affect the change characteristics generated (slave). The communication between them may be unilateral, as shown in Figure 2.6.1, or bilateral; bilateral implies some level of feedback from the slave environment.

Figure 2.6.1: Master, Slave, Communication structure

When implementing control for telerobotic systems, there are many factors to consider. Among these are the purpose of the system, what degree of autonomy required or would be helpful to have, and the role of the operator. For this purpose control architectures have been developed. The need for control architectures also arose out of inherent system factors that negatively impacted system performance. System factors such as control signal latency and video lag, they were developed to overcome or negate these factors.

Three main control architectures are readily used in modern telerobotic systems. Simplest of the three is direct control, this type of control architecture is employed in the da Vinci robotic surgery system (Ballantyne *et al.*, 2003) and is a direct implementation of the master-slave paradigm shown in Figure 2.6.1. This architecture, as the name suggests, relies on direct information flow between the input (user/operator) and output(teleoperator), this information flow may be unilateral or bilateral. Although in modern systems the information, data and control signal flow is almost always bilateral.



(a) Direct Control (b) Coordinated Control (c) Supervisory Control

Figure 2.6.2: Control architectures used in teleoperation / telerobotics

It also outlines the basic system architecture that has been adopted by many system developers as it is the most logical. This architecture has been expanded on to include multiple-input, single-output as is the case for collaborative control, Lee and Spong (2005).

An example of direct control is under-sea unmanned repair robots. The robot is directed by a human, using joystick inputs in order to achieve a desired goal. In the presence of control hardware etc. the user's role is defined by having to be responsible for all decision making, all change and all the aspects that make up control. For a perfect system, the user performance is the system performance; similarly, the user performance will dictate the overall system performance if system limitations are present, (Fong *et al.*, 2002).

Direct control stands in contrast to coordinated control, where there is some level of autonomy on the output side of the system, shown in Figure 2.6.2. The decision making is shared between the user and the system although, the user or in a distributed system users can take control of the system. The autonomous decisions are of less importance and usually involve tasks such as obstacle avoidance. The user's role becomes more indirect and the cognitive of the load of the user is reduced (Lichiardopol, 2007).

Coordinated control is a direct control system with an added control loop local to the slave side. This control loop ensures some level of autonomy for the slave side.

Supervisory control on the other hand, occupies the space between coordinated systems and that which is completely autonomous systems, (Sheridan (1986), Fong *et al.* (2002), Siciliano and Khatib (2008*a*)). The user's role is reduced further to that of a supervisor, the user, sets goals and the system is able to execute the task in order to reach said goal. Although the user can still change goals and the manner in which goals are reached. For example, the user directs a underwater robot to pick up a piece of equipment that fell. As the robot reaches for the target item, the user realises that if the robot is to place the object on the on-board container there is a chance that the umbilical might tangle around the manipulator. The user then adjusts the angle at which the robot will intersect with the object as to avoid the tangle.

Supervisory control is the preferred control architecture for modern teleoperation systems, as the demands on users are far less than what is required by other control architectures.

2.7 Shortfalls in the Research

Because of the wide area of application of telerobotic systems and the various groups of teleoperation and more specifically telerobotics, the research has become far more specialised in its problem definitions. Research focuses on solving or investigating a specific aspect of the system as part of a larger research program and rarely pursues the entire project.

Sheridan (Sheridan, 1989) highlighted some research areas that required the most attention in his early work. He broke them down into four main categories:

- Telesensing
- Teleactuating
- Computer-aiding in supervisory control
- Meta analysis of human/computer/teleoperator/task interaction.

The work done in this thesis focuses on two of the four categories they are namely, Telesensing and Meta analysis. We explore the reaches of a basic 6 degree-of-freedom, haptic control system. The system focuses primarily on a usable, scalable input scheme that allows for natural control of the output end-effector. The system excludes actuated end-effector design and implementation, although this limits generalisation of the results it will show the significant differences in user performance for this system and will guide future development of the teleoperation test bench.

Meta analysis on the other hand will be done by focussing on Human-Computer interaction aspects and evaluating the interface between user and computer. Further Human-robot interaction principles and heuristic evaluation techniques from this field will be employed to evaluate the current system. The heuristic evaluation of systems is a subjective measure of performance for a given system and assists in measuring the usability of the system. Usability is an important factor in when determining the user experience of the current system. This qualitative approach will deliver a usable system although there is not always a guarantee that it will be technically superior.

Many of the needs of the telerobotic systems are born from the inherent complexities of transferring information over very long distances. When considering a generic telerobotic implementation the following aspects must be considered: Vision, Command, Audio, Actuation and the common link and means for all the various signals to travel, the communications link. The concept of ideal teleoperation is based on the premise that there is real-time/ near real-time control and feedback to and from the actuation site to the input side. In reality there will always be a delay, although in the modern applications the delay can be minimised using models, predictive controllers and extremely high bandwidth communication speeds. Most users are most sensitive to delays in vision.

Control latency also plays a role in the final system performance, even with a learning curve most users will have issues resulting from the latency for responses over 700ms as was found by Fabrlzio *et al.* (2000).

Conceptually telerobotic systems are relatively easy to define in human terms; it is the translation from a human reference to the electro-mechanical machine system that poses a challenge. The human body is nearly impossible to replicate electro-mechanically and has many complex systems with specifications that are simply to complex and advanced to duplicate feasibly.

Thus an analogous approach for development and implementation must be adopted, the result is that there will be a far steeper learning curve in operation and that the system will only be approved for use if the user is able to overcome the inherent flaws within the system. The major loss of input sensation and positional awareness may be solved using haptic devices and systems as described in the next section.

2.8 Haptics in Robotics

The term *Haptics* comes from the Greek word "*Haptikos*" meaning "able to come in contact with". As the Greek meaning, modern haptics focus on no verbal communication in the form of touch. Touch is one of human beings fundamental inputs, these technologies have been utilised for some time. One of first mass market examples can be shown to be the "‘rumble" function in some gaming controllers. This is a simple vibrator that is placed in the hand controller of a console in order to add another level of feedback of the user's status within the game.

Haptic interfaces, implying interfaces that require touch to give or receive information. Haptic information is given in two components, which are defined as tactile and kinaesthetic. When considering tactile feedback, it can be described as everything that can be interpreted by the skin. This includes vibrations, the perception of textures and many others. Put simply, tactile feedback is the way an object or surface "*feels*".

Kinaesthetic feedback is the positional and orientational feedback when touching an object or surface. This gives information such as the hardness of a surface or object. This will also include information pertaining to the contours of the surface, this is given by the ability to perceive one's position in space, this is also known as proprioception (Hayward *et al.*, 2004).

Hayward and Astley (2000) also discussed the bi-directionality of haptic systems, they suggest that a haptic interface must be able to both "read and write" to a human hand or other other limb. This implies that the haptic interface reacts to not only to user input but also to the virtual/real output environment. Put simply, haptic systems will always incorporate some level of feedback.

In the evolution of telerobotics, the application or task complexity has increased steadily and has been in pace with technology developments that will enable many new markets to be explored. This has become an enabling factor for developing more accurate haptic displays and interfaces. As part of the modern haptic paradigm one can consider the input environment to incorporate haptic input in some form or another. This was evident even in the early implementations where "haptic" feedback was given by mechanical resistance in the marionette system.

Haptic interface allow for tactile exploration of the remote environment. This to an extent compensates for the lack of Situational Awareness (SA), which allows users to adapt faster and overcome shortcomings in the rest of the environmental interaction of the robot.

These systems have become popular as haptics can expand on classic input-output models. Haptics allow for simulation of real-world simulation or actual interaction. the virtual simulation has many applications in training situations. This has been applied successfully in training for many medical procedures. Haptics allows simulation of minor tactile and kinaesthetic simulation in a virtual training situation, where the surgeons are required to assess the condition or perform treatment based on feel.

The system allows for knowledge sharing and training without endangering any human lives and the virtual environment can be simulated repeatedly, allowing for honing of skills.

Various haptic input devices have been developed. Each of these has their own set of challenges for the user / operator, although all strive towards a natural input modality, that requires minimal adaptive learning in order to master.

There are many categories of input technologies, among them is the exoskeleton design shown by Rait and Mukherjee (n.a) as part of an ongoing research project. Some well-known exo-skeletal designs have been produced by Cyber Glove Systems LLC., such as the CyberGrasp haptic glove that can provide a 12N continuous simulated force for interaction with virtual objects in the computer environment. The technology added another dimension of realism with the advent of the CyberForce, which could also give positional feedback with a maximum of 8N, which is equivalent to a small bump (Pretorius, 2012).

A popular system is the wired manipulator, these systems rely on an object that is grasped by the user and manipulated within a work volume. The object is tied via cables to motors that apply the feedback forces from the output environment on the user's hand. This allows for good kinaesthetic feedback, although it is not good for tactile feedback. This is the case for most haptic systems, at best the system allows for vibratory impacts to simulate the roughness of a surface.

The most popular, haptic systems in use today are the stylus based systems that is connected to a multi-dimensional actuator. These systems are easily understood by a user/-operator and allows for realistic feedback to be applied to the user's hand. An example of such systems is shown in Figure 2.8.1, this specific model allows for three dimensional actuation and haptic feedback.

Figure 2.8.1: Phantom Desktop, geomagic (2013)

Alternative input methods:
Many alternative input technologies exist today; one that stands out above the rest is the Brain-computer interface (BCI). This non-invasive device monitors the user's brain waves and can be applied in the field on industrial robotics and by extension the field of teleoperation Zhang *et al.* (2010). Although these systems may have some advantages that make it indispensable in future applications, the current role of an operator in teleoperation requires haptic feedback that can be felt by the operator in order to add information necessary to make decisions.

Chapter 3

Methodology

In order to understand the research taxonomy employed in this thesis one first has to clearly define the researcher's definition of methodology. Researchers are confronted by varying opinions on classifications of systems development; each develops this definition intuitively and is prone to some biasing factors, such as personal experience. Wynekoop and Russo (1997) showed that the classification of a systems development methodology is difficult as many system developer's definitions of what a methodology is, was found to be inconsistent. For the purposes of this thesis *methodology* in the context of systems development is as follows:

> *A methodology is a systematic approach to conducting at least one complete phase (e.g. requirements analysis,design) of system development, consisting of a set of guidelines, activities, techniques and tools based on a particular philosophy of system development and the target system.*
>
> \- *Wynekoop and Russo (1997)*

The work presented focuses on the development and testing of a bilateral telerobotic systems test bench. The development of new proof of concept (POC) telerobotic system will help guide future development at the university (from a user-centred-perspective), it may also act as an enabler for funding. The test bench focuses on small-field-of-operation tasks and for the purposes of this iteration will be used to study common human factors and their effect on system performance.

The effect of human factors such as ***viewing-mode***, ***input modality*** and ***field-of-view*** on user performance will give valuable insights into the system specific (user-centric) human-factors performance measures. One can argue that these systems have limited applications in the modality presented. However this modality serves to show a POC and with further development may show promise for task specific applications in a wide variety of areas. The case study into human factors and user preference will also assist in focussing development efforts for future development projects and aligns itself with the user-centric design methodology followed throughout the project (see Section 3.2.2 for full discussion).

Many methodologies exist for the development of complex systems such as a telerobotic system. Systems Engineering (SE) is a field pertaining to the design and implementation of a system for the entire life cycle of the system which may be dependent on some process or a part there of. This process can be physical or information based, require a single unit for development or many. Generally this methodology is applied in the development of systems of systems (Blanchard and Fabrycky, 2010). The aim here is not to promote a specific model or developmental process but to show the hybrid process followed, in an effort to show system designers how the design decisions influenced the work, and by doing so expedite future development projects. The work presented here also focuses on the user specific requirements and testing of the system, following a central theme of user-centred systems design, ultimately to produce a usable system - this will assist in making the system commercially viable.

All the methodologies followed in this thesis can clearly be separated into three distinct categories. These are hardware system development (Classical Systems Engineering), software systems development (Software Engineering and HCI) and user centred analysis methods.

Engineering management principles are key in the system development process. As shown by Blanchard and Fabrycky (2010), engineering management forms the basis of development and analysis of a system. This generic systems development process is shown in Figure 3.0.1. This generic model places the emphasis on requirement management and decision management aspects of engineering management.

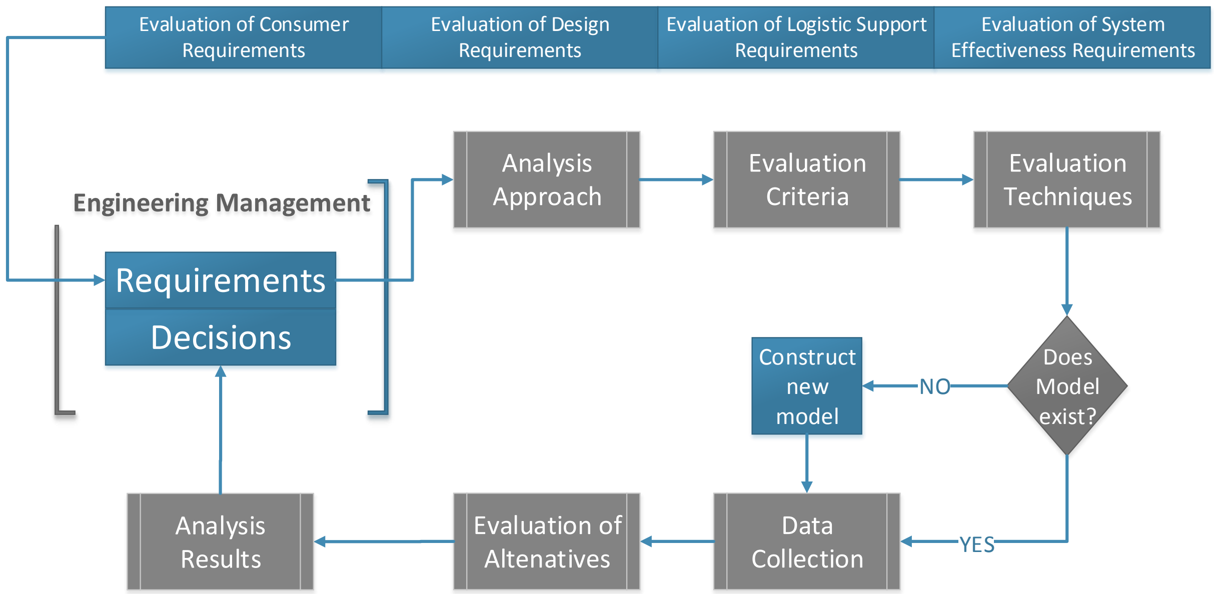


Figure 3.0.1: Generic systems analysis process

Engineering management forms the core of the SE process in that all decisions made throughout the process are inherently engineering management decisions.

3.1 Systems Development Methodology

The SE approach is a holistic approach that sets a framework of development that considers the entire life cycle of a product. The focus of systems engineering is to provide a structured way of accomplishing a complex goal. System engineers focus on development of a system of systems, although this may not always be the case. Intuitively one can see that for the case of developing a telerobotic system for testing various user-centric performance measures the system engineering approach can be applied. This approach allows for iterative development and allows the developer freedom to make design decisions not only based on performance measures but also incorporate human factors analysis and alternative generation as methods for decision making.

Technical performance verification and validation forms a central part of this methodology which plays an influential role in user-centric evaluation techniques (Haskins *et al.*, 2007; DAU, 2012). For teleoperation systems, the system's performance is a function of the performance of its subsystems and has a direct effect on the attainable user performance. The system's performance may be adversely affected by a subsystem that is not to specification or relies on complex input modalities or signal processing.

Teleoperation systems development is a complex task, with many interdependent systems and subsystems that must be taken into account. As SE lends itself to structure and drive developmental processes as a meaningful and coherent whole, it was selected as the preferred methodology.

For this reason a SE approach was adopted for the hardware systems development and integration phase of the project. Further a User-centred System Design (UCSD) approach was adopted, while utilising a research goal orientated development strategy for the systems development process. Research goals were used as driving factors for the system's requirements analysis and evaluation. The scope of this thesis focuses on the significance of the vision mode in the presence of haptic feedback to assess the impact on user performance and to guide future research at the university. To this end a telerobotic system was developed and evaluated. This thesis builds on the work of Mr. J. Pretorius which designed and implemented the initial phase for the telerobotic system Pretorius (2012). The second phase of system development establishes clear performance goals. It also aims to address the shortfalls identified in the initial phase of the development project. The initial system was fairly successful although no clear system requirements and functional analysis was presented. A top-level scenario analysis was conducted to ensure that the system is sufficient for use in the proposed user-centric performance analysis for the visio-haptic correlation tests, the system was found to be too unstable for use. As a result the system had to be redeveloped.

There are many aspects to teleoperation, these range from control systems, mechanical and electronic design, communications technology, information systems, interface design, technical development (systems design), human-computer interaction and more importantly human-machine (robot) interaction. The work presented here focuses around the last four aspects.

The development of kinematic models and complex control systems are outside the scope of this thesis. No in-depth analysis will be performed and only user centric data will be considered for this system in its current state, some generalisations would be possible.

3.2 Systems Engineering

As in most complex development projects, the SE approach would be the intuitive / obvious choice. The System Engineering approach is an interdisciplinary development framework which structures all the phases of system development in a concise and easily understandable way. Haskins *et al.* (2006) define SE in terms of *ISO/IEC 15288: 2002(E) - Systems engineering - System life cycle processes* as:

> "*Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.*"
>
> -*Haskins* et al. *(2006)*

Most development processes can be expedited using ISO/IEC 15288, which provides a comprehensive life cycle process description that contextualises the systems development process within an enterprise as shown in Figure 3.2.1. This is useful to identify the areas on which the Engineering Manager must focus to complete a developmental project successfully. The figure also shows the different management aspects of system development required to develop a successful system, thus showing the implications of the systems life-cycle approach on SE. From an EM perspective the process followed in this project can be classified as:

- Assessment
- Configuration Management
- Decision-making

Further the Technical processes outlined are, as shown in Figure 3.2.1:

- Stakeholder Requirements Definition
- Requirements Analysis
- Architectural Design
- Implementation
- Integration
- Verification
- Validation

The work presented in this project can be classified under decision making in the realm of product development, which in turn lies within systems engineering and the management of the systems life cycle. This pragmatic approach aims to show how to bridge the gap for ease of funding using the project as a proof of concept commercial systems can be used to assemble a generic teleoperation system.

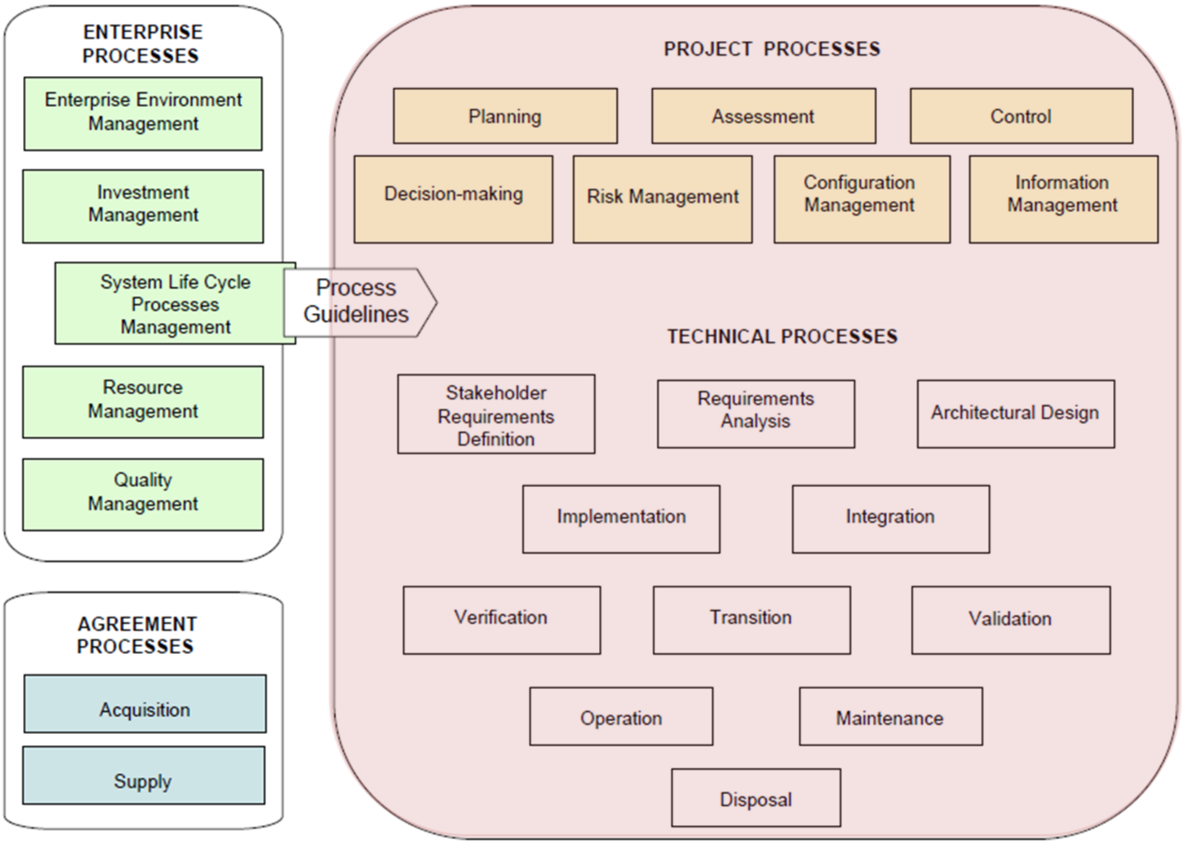


Figure 3.2.1: Systems Life Cycle Overview, adapted from ISO/IEC (2002)

Further to clearly contextualise the current research project within specific knowledge areas. One will first have to first consider the areas of knowledge that will have to be incorporated in order to develop a successful system. Firstly the main system in question must be considered, *Telerobotic Systems*, secondly, how these systems will interface with the user, HCI. Thirdly one will have to consider the use of the system within a scenario where, the system will have to interact with people / users, HRI. The knowledge areas under which this project falls can be seen in Figure 3.2.2. Knowledge from these fields will be required throughout system development process, thus it is important that a developer must be well versed in these fields.

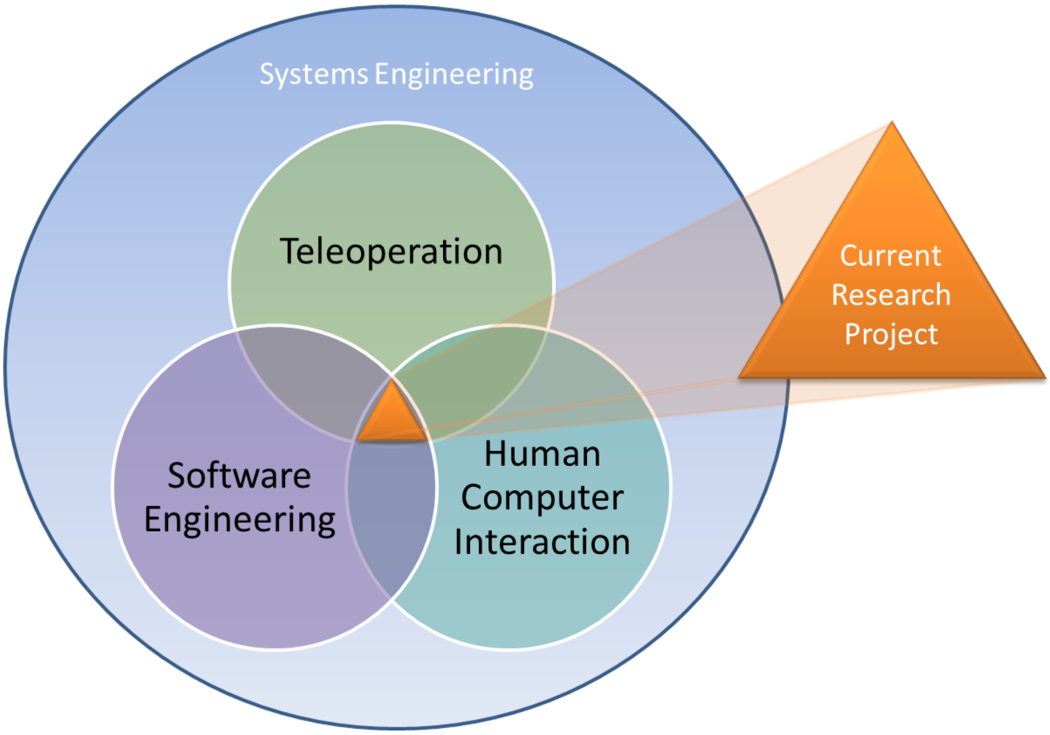


Figure 3.2.2: Project Contextualisation

Detailed development of (complex) telerobotic systems will be discussed with a focus on the incorporation of the various knowledge areas within a SE framework. This approach forms part of a larger methodology that allows for structured systems development of complex systems. Through the completion of this project, the author wishes to provide a short guide for telerobotic systems development. The system will be discussed in terms of usability and user performance, along with technical validation of the hardware components; it may act as a guide for future testing and development projects.

The SE process may be expanded and contextualised with respect to Engineering Management as shown in Figure 3.2.3. Engineering management forms the core of the SE process in that all decisions made throughout the process are inherently engineering management decisions. In order to produce commercialisable and relevant products many functions within the overall process may be categorised as supportive steps and others as core functions.

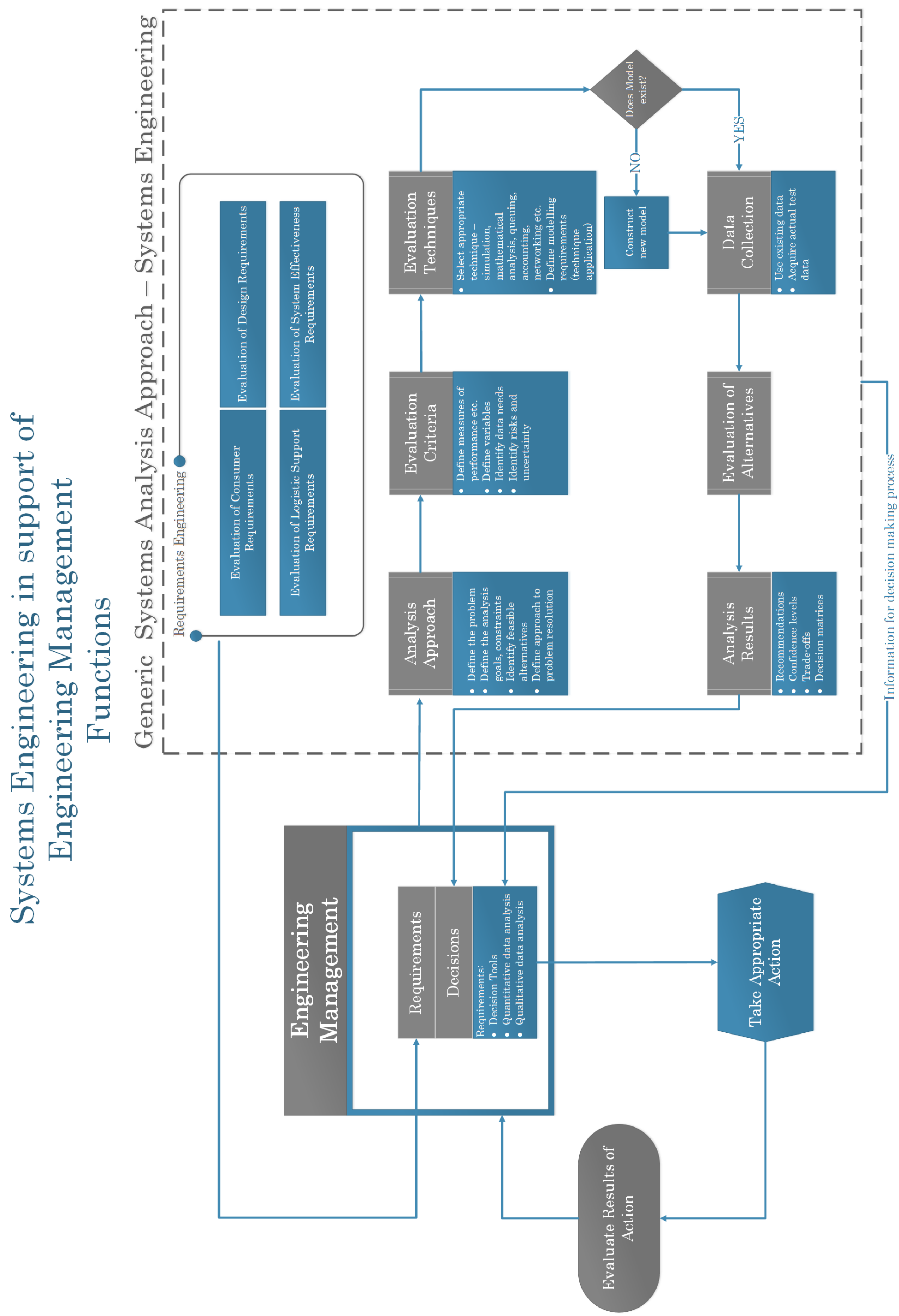


Figure 3.2.3: Contextualised Systems Engineering Process w.r.t. Engineering Management, adapted from Blanchard and Fabrycky (2010)

3.2.1 Available Developmental Models

When developing a complex system such as a telerobotic system, it is important to familiarise oneself with the developmental models and strategies that are available. The chosen developmental approach structures the phases of the development process as well as the stages of evaluation, verification and validation. For this reason a developmental process model must be identified to structure the developmental process.

Many developmental models exist today, each with its own set of pros and cons. Each model is more suited for certain types of projects, the model choice is also influenced by the structure of the company, the resources available (constrained or unconstrained), as well as other considerations such as time or financial constraints. Developer preference also plays a role, as some developers may not be familiar or experienced in the developmental methodology of a given model.

Systems development as well as software development is built squarely on a system engineering foundation. The processes and methodologies involved in software development are nearly identical to that of system engineering. According to Visser and J.G. (2012) there are only three main methodologies when approaching software development, they are:

- Linear
- Iterative
- Incremental

As can be seen these developmental models follow closely from software engineering's System engineering roots. The system engineering approach also lends itself to developing software. This can be shown in the many various development approaches developed specifically for software engineering such as RAD and Crystal to name a but a few.

An addition two methodologies that can be followed has been discussed in the *Centers for medicare & medicare services (CMS)*'s publication "Selecting a development approach" Centers for Medicare & Medicaid Services (2005). Here the CMS outlines Spiral and Rapid Application Development (RAD), which allow for iterative and faster development.

Pallot *et al.* (2010) showed that the proactive user involvement in the development process increased the usability and user acceptance of the system. It is important to note the difference between *useful* and *usable*. A useful system is a system that can complete a task successfully, whereas a usable system makes it easy to be able to complete the task. This may result in developers having to have to make a decision about the trade-off of technical performance and usability.

Many generic systems engineering models exist today, the most common models' pros and cons are shown in Table 3.2.1. The models are shown graphically in Appendix F. Considering the complexities of any developmental project. It is intuitive to deduce that some models are more applicable or naturally tailored to certain developmental projects than others. The chosen or most applicable may not fulfil all the requirements of the project. This is where the developer's knowledge plays a role in adapting or combining one or more of the developmental models.

In order to accomplish this, developers must consider the developmental units. For example for the given project the two main developmental units are: Hardware (electro-mechanical) system development and Telerobotic control and interfacing software development, the top tier developmental component break down are shown in Figure 3.2.4.

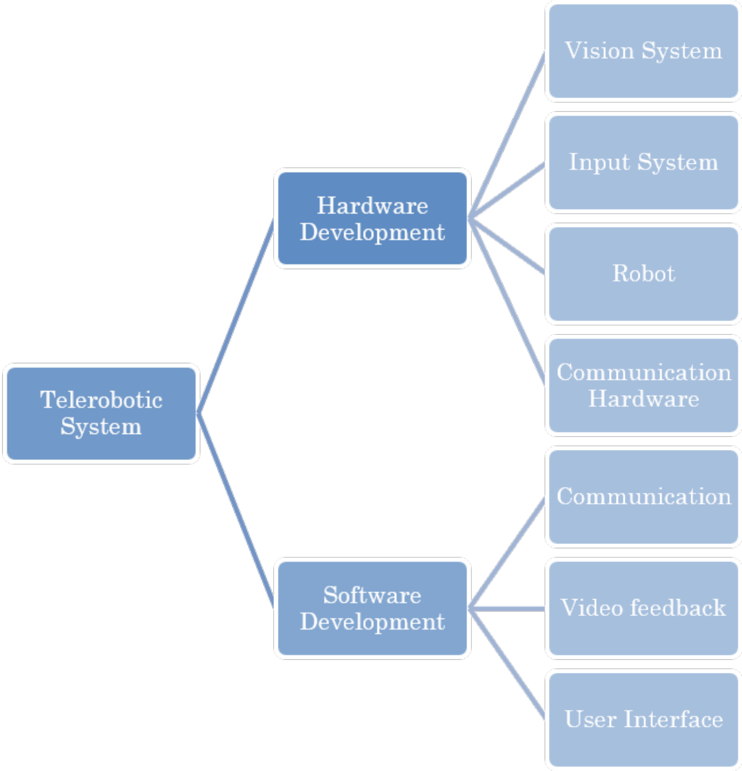


Figure 3.2.4: High level developmental component breakdown

The model or hybrid model that is selected must be able to adjust and adapt to continuous changes in requirements as throughout future development cycles the user and system requirements will have to change due to external pressures. Even during each iteration of improvement there will inevitably be changes to customer / user requirements for the system as shortfalls in usability, functionality etc. are identified. The model should provide tools to ease the generation of user requirements in a timely and cost-effective manner. The model should be implementable in small developmental teams. The developmental process should also preferably be non-intensive on resources.

In order to give a holistic overview of the models available to developers, the major defining characteristics of common developmental models are discussed in Table 3.2.1.

Table 3.2.1: Summary Developmental Models, adapted from Zhao (2012)

Approach	Pros	Cons
Waterfall Model	• Measurable progress of system development • Minimal resources required • Strict controls • Ideal for inexperienced development project teams	• Problem detection usually only in test phase • Slow • Costly • Cumbersome and inflexible
Incremental	• Able to utilise knowledge gained in early increments, which may build developmental inertia • Mitigates architectural and integration risks early • Allows for continuous delivery of a incrementally more complete project	• More complex problems are usually postponed in order to show early success • Well defined interfaces are required in order to avoid integration problems • Has to the potential to exhibit a lack of consideration for OVERALL technical requirements and business problem
Spiral Model	• Increases risk avoidance • Allows for a hybrid approach, and utilising many other sub-developmental models • Highly customizable	• Requires skilled project manager • Highly complex • Very limited re-usability

table continues on next page...

Approach	Pros	Cons
Prototyping	• Improves user participation in all steps of developmental cycle • Allows for early objective clarification • Flexible • Allows for more complete product to be developed by identifying missing/unclear functionality • Allows for quick implementation of functional units	• Relaxed control and approval process • High risk of uncontrolled scope creep or significantly changing requirements • Documentation often neglected • Iterative nature results in more costly projects or longer time to delivery • Dependent on strong cohesion within the development team
V-Model	• Simple and easy to use • Requirements may change at any stage • Specific Deliverables at every stage of the process	• Primarily for use in small projects • Very rigid • Problems detected late • Not for use in short-term projects

These models can also be nested in that for a given system, i.e. the developmental model for a sub-system may differ from the others all while the system follows another larger developmental process model. This hybrid approach will be discussed in Section 3.4, to give an overview of how this type of strategy was applied. These models do however not consider the user needs throughout the process and as a result the UCSD was developed.

3.2.2 User-centred Systems Design (UCSD)

UCSD was developed by the University of San Diego, with a keen focus on the system and the people that use the system. The end user is the main driving force for decisions and as such user participation is key. User participation is present in all phases of the design process and not only in the verification and validation phases.

User-centered system design focuses around task analysis and the user that will use the system. The system requirements also focus around user preference. User testing also plays a role in proactive alternative generation and evaluation by users.

When developing a system or software, developers have to continually consider the needs of your users. A software development project (as is required by the current project) usually includes some form of Graphical User Interface (GUI) that is used to interact with the user. Developing a user interface that can be understood and used effectively by a user with no prior knowledge of the software is challenging. For this reason UCSD may be used as powerful tool to increase usability.

Similarly when developing a system that is user focussed or the user is an integral non-removable part, designers and developers should strongly consider UCSD as the development process model. UCSD is systems design process which conforms to two main principles. The first is that the user needs must be central to the entire developmental effort, and must be uncompromising in its effort to fulfil their needs. The second is that the process is iterative in its approach and that a optimised solution may take many iterations to find, Smith-Atakan (2006).

Further Smith-Atakan (2006) shows that the approach relies on formative evaluation as well as design iteration. Formative evaluation is aimed at finding and exploring key factors that either promote or work in against development goals and requirements. These include but are not limited to the task space (what tasks are the system designed for), user skill, and user needs/requirements. There is also a focus on empirical measurement and testing; the focus shifts to learnability and testing of prototypes during the evaluation cycle.

The iterative process addresses many of the short comings of non-iterative processes, such as waterfall, concurrent and V-models. Iteration allows developers to gather information to address shortfalls in the current design and make changes to the requirement statement governing that function or element.

The design methodology followed to develop the user interface for the teleoperation testing system is based on the UCSD process model. The UCSD process is shown in Figure 3.2.5. This process is similar to most iterative systems design processes and the systems developer will most probably not reach a acceptable solution within the first few iterations. Due to its fundamentals in iterative design, UCSD can easily be incorporated with other developmental models, for example the spiral model.

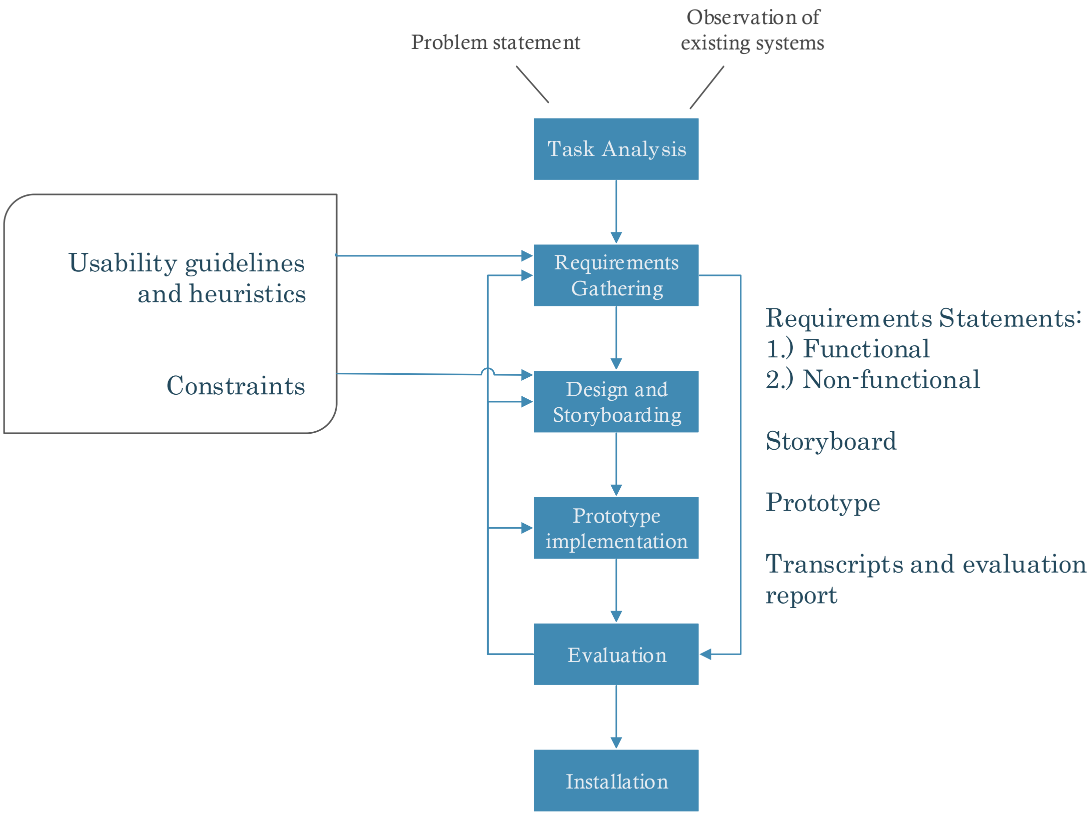


Figure 3.2.5: UCSD model, adapted from Smith-Atakan (2006)

Maguire (2001) argued the benefits of following a UCSD developmental process. Among them are:

- Increased productivity due to a focus on user focused tailored usability. Also allows users to focus on the task and not the tool used to complete it.
- Significant reduction in user error rates.
- Reduces the need for intensive training and support.
- Improved acceptance of the system or software.

Considering the focus on user requirement generation, task centred analysis and design in context of the benefits found using this approach, UCSD has become an attractive alternative to less progressive models in the fast paced development world.

3.3 Software Development

Software development is a key part of any modern computer controlled telerobotic system. The developer must posses the required skills in programming to understand the intricacies in data handling along with the insight to pre-empt any possible problems that might arise from hardware, software integration. It is also imperative for the developer to acknowledge his/her own short comings or inexperience as this will inevitably affect the quality of the final system.

As discussed in Section 3.2.1, many process models exist for systems / software development. Among these spiral, the V-model has had much traction within the industry, although no model is perfect for an specific development project, it is merely more suited to it.

In recent years the software development community has spearheaded the development of so called agile methods. This can ease development and respond to changes in requirements faster than more traditional methods, with a focus on the user. In response for the need for formalising the agile methodology within SE, the agile manifesto was drafted. The manifesto outlined the values of agile methods as follows (Fowler and Highsmith, 2001):

- Individuals and interactions are valued over processes and tools.
- Working software is valued over comprehensive documentation.
- Customer collaboration is valued over contract negotiation.
- Responding to change is valued over following a plan.

The values of agile development, proved attractive for the time-constrained development required for the current telerobotic system, as is the case with the current project. For the given project, the deliverables were destined for an internal customer. This customer formed part of the development team and as such clear and timely communication was possible. Further the aim was to produce software that would comply with all the functional requirements of the system for testing. As the project evolved the needs and requirements for the system had to change and changes to the software had to be implemented on the fly.

Two main competitor models have been identified, they are extreme programming (XP) and rapid application development (RAD) which will be explained in Section 3.3.1 and Section 3.3.2.

3.3.1 Rapid Application Development

RAD is an iterative approach that requires active user participation. It also breaks the projects up into smaller more manageable segments with flexible goals that helps with agile development. This approach is also characterised by the user of computerised development tools such as Graphical User Interface (GUI) builders (Centers for Medicare & Medicaid Services, 2005). The RAD model is shown in Figure 3.3.1 and it consists of three phases, namely *Initiate*, *Develop*, and *Deploy*.

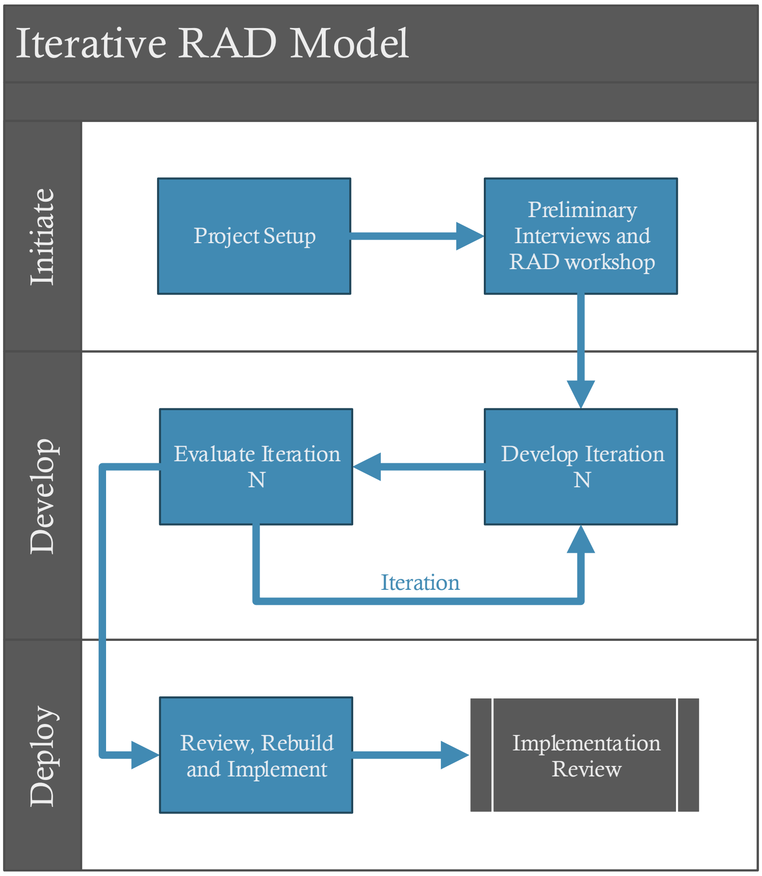


Figure 3.3.1: Iterative RAD Model, adapted from ProfIT Labs Ltd. (2013)

The initiate phase is dominated by the preliminary interviews. These interviews serve as the base for all future development of the project. Client interviews are used to set baseline functionality and specify customer requirements in terms of scenarios or use cases. These requirements will inevitably change as the customer requirements become clearer or change through the development process.

The Development phase consists mostly of iterative prototyping. Here limitations are overcome and teething problems are identified and removed through the use of functional code testing. User / client participation is key to delivering successful software. Here many user changes to requirements force redevelopment. After testing and evaluation, the software is reviewed and scrutinised before implementation.

A good example of a similar implementation scheme is beta testing for large scale roll-out software such as an operating system. Finally the implementation is reviewed and delivered.

RAD is an attractive agile process due to the following characteristics:

- Provides the ability to rapidly change system design as demanded by users/clients.
- Generally produces a dramatic savings in time, money, and human effort.
- Concentrates on essential system elements from user viewpoint.
- The operational version of an application is available much earlier than with Waterfall, Incremental, or Spiral frameworks.
- Produces software that more tightly satisfies user requirements and system specifications.

As a result RAD development is widely used in the industry, although it has fallen out of favour as processes such as XP and Crystal gain traction.

3.3.2 Extreme Programming (XP)

Another agile method considered was that of Extreme Programming (XP). The use of the word *extreme* implies that the implementation of this method for software development is unorthodox. Indeed some developers may argue that the method is not structured enough and that the lack of comprehensive documentation may hinder later maintained efforts. Although this development method has proved to save time and cost while delivering usable software faster.

The XP methodology makes use of the following principles (Kioskea, 2013):

- Development teams work directly with the customer in very short development cycles.
- Early software delivery, with short delivery intervals for review.
- Coding is a team effort and more than one person may code on the same code segment at a time.
- Best practise is enforced during coding; for example, cleaning of code etc.
- Progress of the project is measured regularly as to update the project plan.

These principles allow for functioning software to be delivered fast which satisfies customer needs early. A key principle is that of early delivery, where the client is actively participating in the development process. This allows for fast responses to changes in client requirements.

The simple XP model is shown in Figure 3.3.2. As can be seen, XP is an iterative process that delivers small releases to the customer for testing. At the core of XP requirements analysis lays user stories. These stories are usually no more that three sentences long and describes on a functional level what the software must be able to do. The "how" of the software can be gathered by developer (technical user) stories, which are similar to user stories but describe how the software must interact within the software itself.

The development team can then use their technical knowledge to breakdown the stories into requirements for the software.

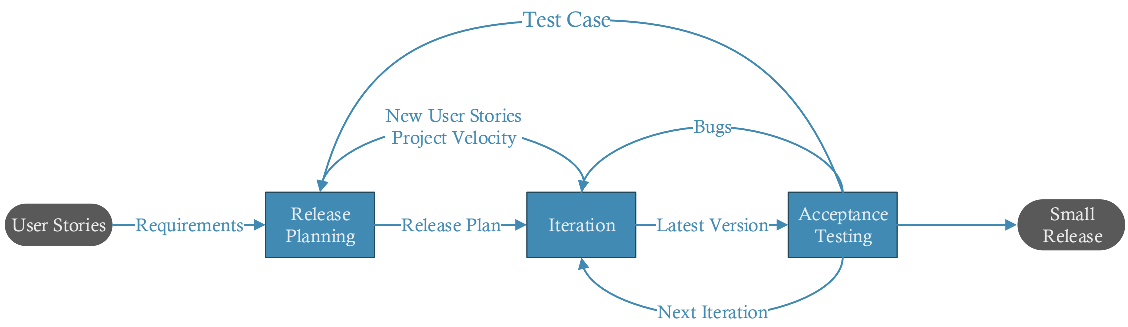


Figure 3.3.2: Simple XP Model, adapted from SERENA (2007)

In order to fully comprehend XP one must consider the work of Paulk (2001). Here the process at the core of the methodology is distilled into 7 principles along with their respective implementations in practise, shown in Table 3.3.1.

Table 3.3.1: Extreme Programming properties, adapted from Paulk (2001)

Common sense	**XP extreme**	**XP implementation practice**
Code reviews	Frequent code review	Pair programming
Testing	Continuous testing with customer involvement	Unit testing, functional testing
Design	Design at the core of daily business for all involved	Refactoring
Simplicity	Simplest design that supports the system's design must be implemented	The simplest thing that could possibly work for current functionality requirements
Architecture	Continuous collective effort to refine the architecture	Metaphor, user stories
Integration testing	Integrate and test several times a day	Continuous integration
Short iterations	Make iterations extremely short	Planning for near future deployment and implementation

Maurer and Martel (2002) showed that XP delivers productivity gains for developers and that it delivers higher quality software as compared to more traditional methods. Further the use of functional acceptance tests ensures that customers are satisfied with the delivered software. The ease of implementation, cost and time benefits for developers and the fact that higher quality software is delivered faster to the client ensures XP's place as one of the leading agile models.

3.3.3 Interface Development and Evaluation

Software interface development has always been problematic, as there is no standard human to design for. Accounting for user preference in all its forms and permutations is an infeasible exercise. In order to overcome this limitation, interface designers may incorporate customisation features. Interface customisation was not a viable option as time constraints will not allow for stable development practices. Also when viewed in light of research goals the customisation might skew the results obtained, as little change between the experimental environment was allowed to keep the environmental variables constant.

Human computer interaction is a field of study focussed on the effective use of interfaces presented by computers. Many tools have been presented, most focus around heuristic evaluation practises Smith-Atakan (2006); Nielsen (1994*a*,*b*). Many well documented heuristic approaches have been presented. Among them are: Norman's seven principles and Schneiderman's eight rules. The components of these usability heuristics are shown in Table 3.3.2.

Table 3.3.2: Heuristic principle comparison

	Norman's 7 Principles	Schneiderman's 8 Golden Rules
1	Use both knowledge in the world and knowledge in the head.	Strive for consistency.
2	Simplify the structure of tasks.	Enable frequent users to use short-cuts.
3	Make things visible.	Offer informative feedback.
4	Get the mappings right.	Design dialogs to yield closure.
5	Exploit the power of constraints, both natural and artificial.	Offer error prevention and simple error handling.
6	Design for error.	Permit easy reversal of actions.
7	When all else fails, standardize.	Support internal locus of control.
8	-	Reduce short-term memory load.

In order for a user interface to be usable, it must conform to the majority of these principles. The design of the user interface will incorporate these heuristic principles. A usable interface also promotes acceptance of the software, thus it is advantageous to spend time to evaluate user requirements that address these principles. The majority of these principles will be included in the GUI design in the development phase of the Control dashboard for the telerobotic system.

3.4 Selected Developmental Strategy

Combining the knowledge from Section 3.2.2 and Section 3.2.1, a hybrid developmental process was devised. As users are at the core of direct-control telerobotics an overall User-centred System Design (UCSD) approach was followed, with a focus on producing a usable system. For this reason evaluation and user requirements focus on user-centric performance measures (human-factors).

Hardware system development and configuration management was expedited by limitations imposed on the system, for this reason many of the subsystems were predetermined. For this reason a V-model approach could be easily be followed, this is due to the small nature of the residual hardware system. The V-model was also implemented with UCSD considerations.

Software development utilised a combination of Extreme Programming (XP) (Agile) development strategy with UCSD considerations. This process model was selected mainly due to its flexibility and short delivery times.

3.5 User Performance Testing

Note: The performance testing was conducted post- system development.

When developing a user-centric system, i.e. a system in which overall system performance is directly linked to user performance; human factors are an important consideration. Human factors within a system do not have direct performance indicators, for performance assessment heuristic approaches has to be adopted.

Performance indicators are also mostly task specific and are influenced by what the developer or tester deems to be indicators of performance. Also the performance indicators directly link to the human factors to be tested. For the given system the human factors considered are primarily focussed on vision components and input modes, this will be explained fully in the following sections.

The limited set of human-factors were extracted from Chen *et al.* (2007), which completed a comprehensive study in the field of human performance issues with a focus on telerobotics was conducted, this comprehensive review discusses topics from more than 150 papers. The main issues influencing human performance with respect to telerobotic task completion rates and other performance measures are as follows:

- Limited Field of View (FOV)
- Orientation
- Camera Viewpoint
- Degraded Depth Perception
- Degraded Video Image
- Time Delay
- Operator Motion

Further user testing scenarios were used along with base task descriptions, i.e. moving objects from varying heights, moving to a target, navigating through the environment, for requirements generation.

3.5.1 Vision System Evaluation Procedures

The vision system plays a crucial role in the development of a teleoperation system. The current system relies heavily on fixed camera angles and poor image quality with little or no control over the viewing mode, i.e. course PTZ control and fixed camera resolution. The newly implemented system allows for both physical manipulation of viewing angles and flexible video resolution selection. The vast majority of viewing modes have also been incorporated into a control website that can be accessed both internally and externally.

This section aims to explore the correlation between user performance and subjective elements of the specific telerobotic system. The tests will be conducted with and without haptic control to also investigate whether or not there are any significant performance differences. A summary of the tests scenarios that will be tested is given in Table 3.5.1.

System Setup:
The system consists of two cameras. One mounted in a fixed position above the work area, the other mounted on a robotic arm. The fixed (perspective) camera is a Pan/Tilt/-Zoom (PTZ) camera, giving the user added functionality that allows for customisable and flexible viewing of the operation field. This camera also has a build in network server, allowing for direct connection and streaming over any internet browser from any location. The fixed camera requires external hardware to interface with the network as explained in detail in Section 4.2.1.2.

The perspective camera system consists of a pro-sumer High Definition (HD) camera that has 3D capability. This camera was chosen as it allows for the most cost effective solution to both allow 3D and 2D streaming via a server.

3.5.2 User-centric Testing Procedure

The tests have been adapted from various standard usability tests for teleoperated robotic systems. The main performance measures of the system will be tested on a local network only where the communication speed of the communication channels can be regulated and kept constant with minimal or no congestion thus simulating a dedicated channel. There are three tests that will be conducted. All these tests have been developed to assess user performance in both the two and three dimensional operating fields. All the tests will be repeated in the scenarios specified by Table 3.5.1.

Table 3.5.1: Vision System Testing Scenarios

Vision Mode	Input Mode	Full-field	Perspective
2D	Haptic	x	x
	Joystick	x	x
3D	Haptic	x	x
	Joystick	x	x

Planar Operational Accuracy:
Many teleoperation procedures require planar tasks such as moving small parts or sub-assemblies from one location in the field of operation to another. The test has been designed to follow a multi-level planar select approach, the user will have free control of x and y planar movement using the haptic or joystick input modes. The level of the z-plane is selected using input from the host control computer's keyboard, i.e. the z-plane is set, prior to experimentation.

The test is set up using a safety mat, a porous sponge material that is approximately 4cm thick. The safety limit will be placed 1.5cm below the top surface of the safety mat as to avoid damage to the robot and the test equipment. The safety level can be programmed into the robot using either the pendant (locally) or the host-control computer (remotely). This will be done at both the master and slave workbenches.

For the planar experiment the users will be asked to complete a simple maze. The maze will be presented with way-points indicated by small 1,5 x 1,5cm squares. The users are instructed to complete the maze as fast as possible without hitting or wondering outside the walls or limits of the maze. The user must pass through each consecutive way-point before being allowed to move on to the next one. If the user "hits" a wall or wonders outside the confines of the maze, it is recorded as a collision. One would expect to see an increase in collisions for a decrease in time to completion.

The maze is also equipped with a capacitive touch, flexible surface that tracks the motion of the contact point between the robot arm and the maze surface. This data may be used to find causes of delays or analyse the trajectory path of the contact point.

The data that will be gathered is quantitative:

- Time to completion for given maze
- Total time taken to complete given maze
- Number of starts
- Number of collisions
- Intensity of collisions
- Recoverable failures
- Unrecoverable failures

The planar test will repeated for each viewing case using haptic input, then again without haptic input; utilising direct (joystick) control. This aims to show if there is any significance in user performance and how viewing mode and haptic control interacts for planar actions, the experimental design is shown in Section 3.7.

3D Operational Tests:

This experiment is designed in such a way that it will test both the users' ability to perform a pick and place manoeuvre in a small 3D environment, but also test the accuracy with which the user can complete the tasks and what the trade-off will be between time to completion and accuracy.

The user will be asked to move a set of brightly coloured rings from one platform to another. Each platform will be equipped with a target, similar to those used in target shooting. The task will require a user to complete a set of sub-tasks:

1. Position the 90° hooking tool to a position to hook the ring.
2. Manoeuvre the tool to using a combination of rotations to be able to hook the tool.
3. Lift and move the ring to the new position.
4. Place the ring as accurately on the raised target platform.

The task ends as soon as the user switches off the robot servo controller. The user aims at scoring the highest possible overall total for the repeated test under varying conditions. The accuracy is then gauged by both a user and constant observer. The results of the accuracy score will give an indication of the user bias or preference for a given combination of input and vision mode.

The rings position will be noted after each completion, a score will be given according to the maximum radius of overlap, i.e. the higher the score the worse the performance. The same performance data will be used as in Section 3.5.2: *Planar Operational Accuracy*, with the addition of the following:

- Accuracy score (1 - almost falling off to 5 - bullseye)

3.5.3 Subjective Analysis

Also a questionnaire will be conducted upon completion of each test. The questionnaires focus on the weighted difficulty of the tasks under the various scenarios. This will be used to correlate actual system performance to perceived difficulty and performance. The user will be asked three questions to assess the users' perception of the task. They are:

- Rate the complexity of the task (1 - not complex at all to 5 - very complex)
- Rate the overall difficulty of the task (1 - very easy to 5 - very difficult)
- In your opinion how accurate did you complete the task (1 - Not at all to 5 - very accurately)

This data will be used to set up correlation matrices to show what user opinion of the system is and how difficult tasks are perceived to be under various viewing and control modes.

3.6 Research Protocol - Contextualising Protocol with Respect to Research Aims

As operators are a key component of telerobotic systems, human factors play a crucial role in both the design and implementation phases of such projects. The User-centred design methodology naturally lends itself to improving usability of a system by focussing on the user within the system, throughout the entire development process. As a result several relevant UCSD principals will be used to develop and test the system.

The evaluation of human-factors plays a crucial role in decision making for design of telerobotic systems. The results from the evaluation may also be used to decide on preferred operation modes for telerobotic system, be it for future experimentation or further development. The aim of the case-study into human-factors for bilateral telerobotics is to determine the relationship between user performance (for the *current* implemented telerobotic system) with and without haptic feedback and under varying vision conditions, for small-field-of-operation (SFO) work volumes. This will assist in guiding future low-cost, feasible research projects, as well as help set guidelines for testing schema.

The *current* project does not claim to be a complete study but rather serve as a guide for future development and research at Stellenbosch University, Department of Industrial Engineering, in the field of telerobotics. The evaluation was three fold (as shown in Figure 3.6.1), the first study was system verification orientated. The technical specification verification process is a key part of all system development approaches, as it is measures how well a system matches up to the original ideal system envisioned by the designers.

The study was also an empirical comparative study to show the change in system performance for the *initial* telerobotic system compared to the *current* system after crucial communication and design upgrades were implemented.

The evaluation does not only focus on time and force based performance measures but also strives to assess if the *current* system is a one-to-one system. If a telerobotic system is one-to-one, the output mimics the input exactly with no noticeable delay. It is possible for systems to not be one-to-one, this may be attributed to delays in execution or filtering execution errors and may compound with undesirable consequences.

A development goal that was identified in the conceptual stages is the fact that for spatial awareness and memory to be employed the system must have, as close as possible, one-to-one mapping. If this cannot be achieved, the system performance will be adversely affected. The comparative study will also test the new posture based haptic control system with a newly developed Proportional, Integral (PI) controller, the design and testing of the PI controller will be discussed in Section 3.8.1.2.

The second and third studies were user focused, they were simultaneously completed, and will focus mainly on qualitative and subjective analysis of the system, more specifically the human-factor driven aspects of the system and user preferences.

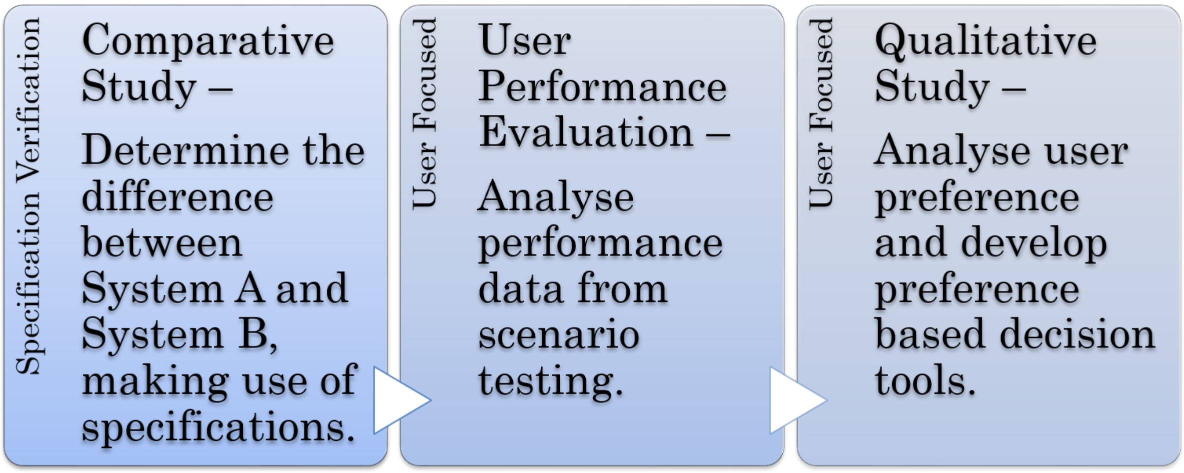


Figure 3.6.1: Testing Procedure

For the given analysis the unit of variance is the user him / herself as the system performance for each sub-system and the system as a whole is fixed, i.e. the variance due to system parameters are seen as negligible as system parameters will be exactly the same for all users. Although this is the case, there is no way to directly separate the variation introduced by the system and the variation of the user. This can be reasoned logically as well, because the system cannot operate without a user. Thus the variance of the system in the use condition is the combination of the variance of the system and the user.

Random, naive users where identified for the tests, all have good or corrected binocular vision. User data was collected for basic teleoperation tasks (the task design process will be discussed in Section 3.10), these where only studied for small-field-of-operation work volumes. Some of the tests were designed to test the limits of operation, for example, conducting a 3D spacial test using only 2D vision.

The first of the user focused studies aims to assess user performance under various vision and control modes. The aim of this section is to assess the impact of the visio-haptic influence for the user. This shows the trade-off and performance differences that can be used to discriminate between some vision modes when or if haptic control is implemented. A statistical hypothesis test such as an Analysis of variance (ANOVA) with a Tukey honestly significant difference (HSD) will first be completed, further a Wilcoxon rank-sum test and Friedman test with post-hoc analysis will be used to analyse the data and show if there is a significant difference in user performance under varying conditions.

All users tend to be visually dominant, as was shown by experiments conducted by Pavani *et al.* (2000). The experiments centre around what is known as a distracter test, for this kind of test a user is placed in a position where they are to manipulate cubes in hand, but is presented by visual stimulus. The visual stimulus is a pair of rubber hands that are placed in a predefined position. After some time the user when asked in which position their own hand is, will point toward the position of one of the rubber hands (Hecht and Reiner, 2009), showing that users are visually dominant. This needs to be taken into account in assessing the success rate of a task when the output cannot be seen or felt directly.

Furthermore subjective analysis is an important part of usability studies for a product or service.Subjective data is important in assessing why some results were achieved or can easily show the level of acceptance of a specific element of the system. To this end the third study was performed; user perception of difficulty was assessed by making use of questionnaires that relate to the various tasks. Subjective data forms an integral part of usability engineering as the usability of a product or system rests upon the user's experience with said service or product.

3.6.1 Experimental Summary

The experiments will be conducted as follows: Firstly a set of user evaluation tasks will be conducted under various vision modes for a joystick input modality. Then the same set of tasks will then be performed using haptic feedback control. The results will be compared in order to assess the performance differences under different control modes. The tasks have been designed to test basic motion that can be expected in small field operations. These test operational accuracy as well as defining a platform to test more complex haptic tasks. This also shows the differences between user performance under various vision modes for a given input modality.

The key user-centric performance measures in teleoperations are difficult to define as researchers try to quantify a subjective qualitative experience. Performance measures can be broken down into two categories; they are spatial and time-based measures. Examples of spatial measures are orientation (degrees), physical position. While examples of time-based measures are time-to-completion (TTC) and total time taken. Other empirical user-centric performance measures include: number of attempts, number of collisions and severity of collisions. These all contribute to a user performance index for the system.

Time based measures relate strongly to the spatial measures. Time based studies are dependent on the inherent latency of data transmission that is always present in any telerobotic systems.

User perception of task difficulty can be difficult to quantify as there is no hard and fast method for compensating for user preference. Subjective data such as user perception of difficulty plays a crucial role in developing telerobotic systems; as insight into why some contrary results may be found may be clarified.

3.7 Design of Experiments

When considering user-centred systems design as the main development process, the factors of interest will inherently not only be technical specifications but also be user focussed. User specific factors also known as human-factors are the system factors that are associated with human senses and human interaction with the rest of the system ,(International Ergonomics Association, 2013).

As human-factors play an crucial role in the evaluation of a system and as intrinsically the overall performance of the system. Human factor evaluation can also be a measure of the usability of the system, and may be used to identify shortcomings or performance hampering elements. Furthermore it has proved to be useful in finding elements of a system that do not add to overall performance but may be expensive to implement. In contrast it may also show the elements that do not add performance but the removal thereof may negatively affect overall performance.

The experimental design process is fairly simple once all the contributing elements have been identified. For this, the following elements must first be identified and defined:

- Identify the elements of interest for the system.
- Define the questions that the experiment must help clarify.
- Identify factors that may have an effect on system element performance within the system (**independent** variable(s), continuous or discrete, controllable or uncontrollable).
- Identify factor levels that are appropriate for the given experiment.
- Identify the performance metrics (**dependent** variable(s)).

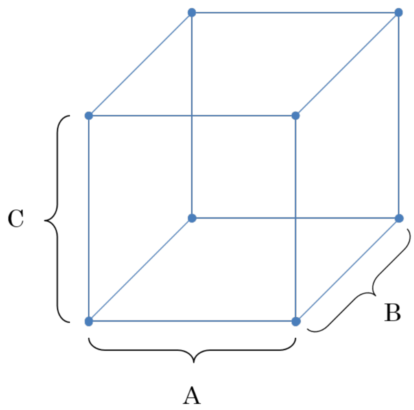
From the Chapter 1 the human factors must help identify if there is any effect in user-centric performance measures for changes in common human-factors. The data required to make the heuristic and quantitative analysis has been identified from Section 3.5.

Each factor has two distinct levels, for this reason a 2^k full factorial design was chosen for the given evaluation of human-factors. Each combination of factor scenario or experimental trial for a specific combination of factors, or treatment, is kept constant. As the factor levels will be chosen by the experimenter the experiments follow the fixed effect model.

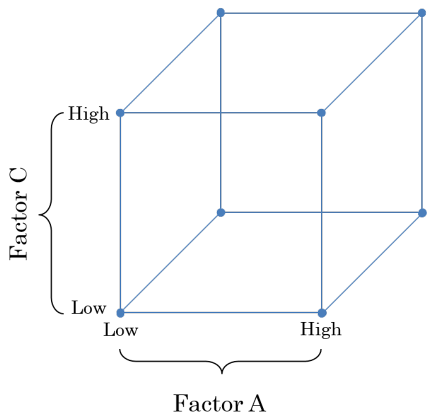
For the current evaluation, the three main human-factors for usability and intrinsic user performance were identified as *input mode*, *viewing angle*, and *viewing mode*. The three human-factors cover aspects of vision, physical skill, and cognitive ability. Situational awareness is the user's ability to assess and make decisions based on the limited information supplied by the feedback devices, thus the user's SA will be challenged or differ for changes in *viewing angle* and *viewing mode*, while changes in *input mode* will test both physical skill and cognitive ability.

The experimental design also aimed at minimising potential uncontrollable biases by randomly assigning the order of a given treatment for all users, this is known as *randomization*. It also serves to minimise the effects of extraneous factors that might be present and that the experimenter is unaware of.

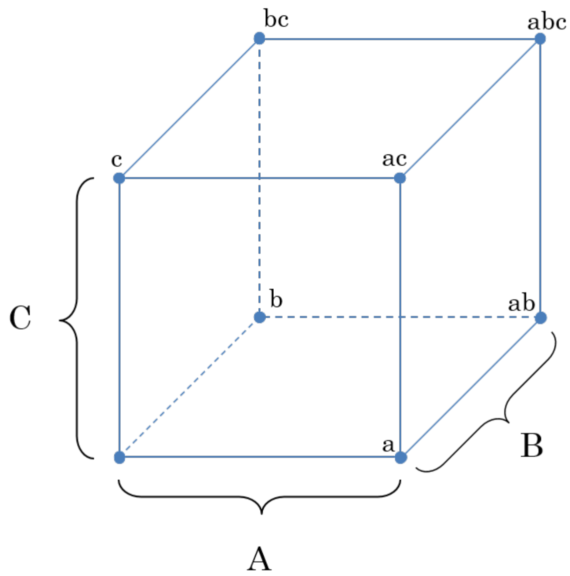
For a balanced 2^3, each participant will have to undergo each of the 8 possible treatments. The response for each of the treatment combination can be represented geometrically as a cube. The general case for a 2^3 design is shown in Figure 3.7.1, for factors A, B, and C.



(a) Generic Geometric 2^3 Factorial Design



(b) 2^3 Example levels



(c) Geometric representation of 2^3 factorial design with treatment combinations

Figure 3.7.1: 2^3 Factorial Geometric Representation

Further, for analysis purposes it will be required that the Effects of each factor A,B, and C be calculated, along with the interaction effects AB,AC,BC,and ABC. It is useful to construct a contrast table as shown in Table 3.7.1 for further analysis, given in standard order or otherwise known as Yates' order. The treatment combinations are shown geometrically in Figure 3.7.1c.

Table 3.7.1: Obtaining Effects for Factors A, B, C, and interactions AB, AC, and ABC for 2^3 Design, adapted from Gitlow *et al.* (2005)

	Contrast						
Notation	A	B	C	AB	AC	BC	ABC
(1)	-	-	-	+	+	+	-
a	+	-	-	-	-	+	+
b	-	+	-	-	+	-	+
c	-	-	+	+	-	-	+
ab	+	+	-	+	-	-	-
ac	+	-	+	-	+	-	-
abc	+	+	+	+	+	+	+

The main effects can be calculated using equations 3.7.1 - 3.7.3, (Gitlow *et al.*, 2005).

$$A = \frac{1}{4}(-(1) + a - b + ab - c + ac - bc + abc) \tag{3.7.1}$$

$$B = \frac{1}{4}(-(1) - a + b + ab - c - ac + bc + abc) \tag{3.7.2}$$

$$C = \frac{1}{4}(-(1) - a - b - ab + c + ac + bc + abc) \tag{3.7.3}$$

The interaction effects may be calculated using equations 3.7.4 - 3.7.6, and 3.7.7.

$$AB = \frac{1}{4}((1) - a - b + ab + c - ac - bc + abc) \tag{3.7.4}$$

$$AC = \frac{1}{4}((1) - a + b - ab - c + ac - bc + abc) \tag{3.7.5}$$

$$BC = \frac{1}{4}((1) + a - b - ab - c - ac + bc + abc) \tag{3.7.6}$$

$$ABC = \frac{1}{4}(-(1) + a + b - ab + c - ac - bc + abc) \tag{3.7.7}$$

The effect sizes are useful in drawing conclusions based on the relative effect size for varying levels of a given factor. For example if A is calculated using equation 3.7.1 an the result is large with a factor 5, then we can say that for that factor the response at the "high" level is 5 times greater than that at the "lower" level.

The effect sizes are useful in determining some of the Sum of squares values in the ANOVA hypothesis test which is explained in Section 3.7.1.

3.7.1 ANOVA Explained

The ANOVA is a statistical hypothesis test. The most valuable conclusion from this test is to find if there is a significant difference in the group means between two treatments, or levels of a given factor. The attributes that make this hypothesis test attractive is its ability to simultaneously test many factors with 'n' levels. The ANOVA is also robust against violations of the normality assumption.

The aim of the analysis is to shed light on the relationship between the response variable and the chosen independent variable(s). In this case, the three main human factors that have been identified as key factors in telerobotics. For this reason we would like to compare the mean Time-to-completion (TTC) for each treatment. It would be safe to assume that there may be an degree of interaction between factors.

ANOVA is a fairly simple yet robust hypothesis test. The ANOVA is used to analyse the differences in the group means of related data. An ANOVA is particularly useful in experimental scenarios where the experimenter is required to work with empirical data.

There are two main ANOVA models available. The first is the fixed effect model (FEM) while the other is known as the random effects model (REM). The main difference between the two models is that FEM is interested in the experimenters chosen parameters only and that the conclusions drawn from them only applied to the given factors and cannot be generalised. On the other hand REM utilises a random sample from a large population and conclusions can be generalised for the whole population of treatments.

The ANOVA makes the following assumptions:

- The data is normally distributed, some robustness against violations but cannot handle severe outliers.
- The standard deviation of each group is approximately equal.

The ANOVA compares the relation between two sources of variation in the data and compares there relative sizes (Pruim, 1999). For this it is useful to use the decomposition of the total variance of the observations, as shown by Hosek and Erin (2011).

Nomenclature:

SS	- Sum of all the squared effects for each factor / treatment
k	- Degrees of freedom
MS	- Mean Square, defined for each factor as SS/k
MSE	- Mean Squared Error, collective variance for all the observations within each level
F	- F-statistic, MS_{factor}/MSE
I	- Number of levels in factor A
n_i	- Number of samples at level i for a given factor
$\bar{y}_i$	- Average factor response for level i
$\bar{\bar{y}}$	- Grand mean

In order to construct the ANOVA table, the sum of squares is required. The sum of squares decomposition is given by equation 3.7.8.

$$SS_{total} = SS_{between\_treatments} + SS_{withinstreatments} \tag{3.7.8}$$

For the main effects SS_{main} is calculated using equation 3.7.9 and 3.7.10 :

$$SS_{main} = \sum_{i=1}^{I} n_i(\overline{y_i} - \overline{\overline{y}})^2 \tag{3.7.9}$$

$$SS_{main} = n_1(\overline{y_1} - \overline{\overline{y}})^2 + n_2(\overline{y_2} - \overline{\overline{y}})^2 \tag{3.7.10}$$

Furthermore the interaction SS can be calculated as shown in equation 3.7.12.

$$SS_{\text{i nteraction}} = \sum_{j=1}^{J} \sum_{i=1}^{I} n_{ij}(\text{ij}^{\text{th}} \text{ level effect}) \tag{3.7.11}$$

$$SS_{\text{i nteraction}} = \sum_{j=1}^{J} \sum_{i=1}^{I} n_{ij}(\overline{y_{ij}} - \overline{y_i} - \overline{y_j} - \overline{\overline{y}})^2 \tag{3.7.12}$$

For a balanced three factor experimental design with 2 levels per factor each, the degrees of freedom for any of the main effects will be 1, from $I - 1$. In a similar way the degrees of freedom for the interactions can be calculated, e.g. for the interaction between A and B with $I - 1$ and $J - 1$ as respective degrees of freedom. The interaction will have $(I - 1)(J - 1)$ degrees of freedom.

Further the MSE can be calculated using equation 3.7.13, MSE is the pooled variability within each level, thus from equation 3.7.8 we can see that if we subtract all other sum of squares from SS_{total}, we are left with the MSE.

$$MSE = SS_E/k_E \tag{3.7.13}$$

The MS is also required for each factor and interaction and may be calculated from equation 3.7.14, where factor A was used as an example and I is the number of degrees of freedom for factor A.

$$MSE = SS_A/(I - 1) \tag{3.7.14}$$

From this the F-statistic may be calculated:

$$F_{factor} = MS_{factor}/MSE \tag{3.7.15}$$

From this we can construct the ANOVA table. The generic table shown in table 3.7.2, which shows the collection of F statistics that can be used to find the p value, using a F table for a chosen α level, usually 5%. The p-value will show if there is any significant differences, it comes highly suggested that the test is followed by a post-hoc analysis, namely the Tukey HSD test to confirm that there is truly a significant difference.

By using the ANOVA one can determine which factors play a significant role in changing the response variable(s). It does this by identifying the factor combinations that result in a difference in the mean of the response variable, which is not due to chance or random variation. It is also important to note that it is recommended that a Tukey HSD post-hoc analysis be completed, considering the size of the samples available to cross check in case there was a violation of one or more of the assumptions for the ANOVA.

Table 3.7.2: Generic ANOVA table for 3-factorial design

Factor / Interaction	Degrees of Freedom	Sum of Squares (SS)	MS	F
A	I-1	$SS_A = \sum_{i=1}^{I} n_i(\overline{y_i} - \overline{\overline{y}})^2$	$SS_A/(I-1)$	MS_A/MSE
B	J-1	$SS_B = \sum_{j=1}^{I} n_j(\overline{y_j} - \overline{\overline{y}})^2$	$SS_B/(J-1)$	MS_B/MSE
C	K-1	$SS_C = \sum_{k=1}^{I} n_k(\overline{y_k} - \overline{\overline{y}})^2$	$SS_C/(K-1)$	MS_C/MSE
AB	(I-1)(J-1)	$SS_{AB} = \sum_{j=1}^{J} \sum_{i=1}^{I} n_{ij}(\overline{y_{ij}} - \overline{y_i} - \overline{y_j} - \overline{\overline{y}})^2$	$SS_{AB}/(I-1)(J-1)$	MS_{AB}/MSE
AC	(I-1)(K-1)	$SS_{AC} = \sum_{k=1}^{J} \sum_{i=1}^{I} n_{ik}(\overline{y_{ik}} - \overline{y_i} - \overline{y_k} - \overline{\overline{y}})^2$	$SS_{AC}/(I-1)(K-1)$	MS_{AC}/MSE
BC	(J-1)(K-1)	$SS_{BC} = \sum_{k=1}^{J} \sum_{j=1}^{I} n_{jk}(\overline{y_{jk}} - \overline{y_j} - \overline{y_k} - \overline{\overline{y}})^2$	$SS_{BC}/(J-1)(K-1)$	MS_{BC}/MSE

3.7.2 Tukey HSD

Tukey Honestly Significant Difference (HSD) test is is also known as the Tukey-Kramer method and is commonly used as a post-hoc test for an ANOVA hypothesis test. The test is a multiple comparison test that compares all possible pairs of means and is based on the studentised range distribution (Lowry, 1999).

The test assumes that the data is normally distributed. The test is based on equation 3.7.16, where μ_A and μ_B is the means of the treatments which are being compared. Further SE is the standard error for the given data.

$$q_s = \frac{\mu_A - \mu_B}{SE} \tag{3.7.16}$$

If the q_s test statistic is greater than the critical value from the standardised range distribution tables for the given α level, then there is a truly significant difference in the data.

3.7.3 Non-parametric Tests

When working with small sample sizes it becomes advantageous to work with non-parametric tests. The reasoning behind this is that if there is little or no information about the distribution of the data, the data might seem to be normally distributed but in truth is not. As the sample sizes that are available for user testing is considered small, the need for a statistical test that makes no assumption about the distribution of the data becomes apparent. Non-parametric tests are also more robust against outlying observations that can otherwise be problematic, although this does come at a loss of power (Whitley and Ball, 2002).

Smith (2010), showed that non-parametric tests can be used for continuous data, although this comes at a loss of power. The sensitivity to the differences in means is diminished. For this reason we can consider the test(s) to be conservative in nature. Two analysis methods where identified, the Wilcoxon signed rank sum test and the Friedman test

3.7.3.1 Wilcoxon Signed Rank Sum Test

The Wilcoxon signed rank sum (WSRS) test was chosen as it shared most of the properties and assumptions common to non-parametric hypothesis tests. Some important assumptions are that it does not assume normality of the data set and is resistant to outliers. The WSRss test is essentially a a t-test for paired samples.

The null hypothesis is that the difference in population means are 0 (zero), put differently the treatment effect is 0 (zero). The WSRS differs from the Wilcoxon ranked sum test in that test also allows for inference about the whole population (Pearson Education, 2011). The formal procedure for the Wilcoxon Rank Sum test is as follows:

Hypothesis:

- H_0 : There is no difference between the means of the given sample pairs.
- H_1 : The difference between the means of the sample pairs are not 0.

Nomenclature:

N_0	: Number of paired observation sets
N_t	: Number of observed pairs not resulting in a difference of 0
$x_{1,i}$	: Treatment 1, observation i
$x_{2,i}$	: Treatment 2, observation i
R_i	: Rank number i
σ_w	: Standard deviation of the Wilcoxon signed rank sum test statistic W
μ_W	: Mean of the test statistic W
$Z_{crit,N4,\alpha}$	: Critical Z value from table, for given number of samples N_t and alpha level α

First we obtain the ranked data scores, this is done by assigning a rank $R_i = 1$ for the lowest scored data and incrementing for each consecutive score. If two or more scores are tied, then they are assigned the average of their rank range.

The method can best be described at the hand of a simple example. To enable ease of use we construct a table, an example table is shown in Table 3.7.3. From this example data set it is clear that $N_0 = 6$ and that the considered final data set is $N_t = 5$. Further we have a score tie situation, the range is between two entries that span the 2 and 3 position in order of rank, thus they get assigned the average value of 2,5 for the rest of the computation.

Table 3.7.3: Example Wilcoxon signed rank sum computation table

i	$x_{1,i}$	$x_{2,i}$	$\lvert x_{2,i} - x_{1,i}\rvert$	sgn	R_i	$sgn \cdot R_i$
1	3	6	3	+	2,5	2,5
2	7	7	0	0	0	0
3	9	2	7	-	4	-4
4	8	11	3	+	2,5	2,5
5	4	3	1	-	1	-1
6	1	15	14	+	5	5

This computational table can then be used to determine the Wilcoxon test statistic W, and can be computed using equation 3.7.17. Further it is convenient to determine the standard deviation of the test statistic σ_W which is given by equation 3.7.18.

$$W = \left| \sum_{i=1}^{N_t} \left[\text{sgn}\left(x_{2,i} - x_{1,i}\right) \cdot R_i \right] \right| \tag{3.7.17}$$

$$\sigma_W = \sqrt{\frac{N_t(N_t+1)(2N_t+1)}{6}} \tag{3.7.18}$$

Next we calculate the Z value for the test statistic using equation 2.6.1.

$$\mu_W = \frac{N_t(N_t + 1)}{4} \tag{3.7.19}$$

For the current project the sample sizes will be small, ten people or less. According to Lowry (2011) it is important to use the Z values from a table for sample sizes smaller than 10. If the Z value is larger than the $Z_{crit,N4,\alpha}$, we reject the null hypothesis. This shows that the null hypothesis is not rejected. As a result it is now known that there is a significant difference between the means, at the selected alpha level.

For the current research project, the following assumptions hold as required by the work presented by Lowry (2011):

1. The paired values of $X_{1,i}$ and $X_{2,i}$ are randomly and independently drawn (i.e., each pair is drawn independently of all other pairs).
2. The dependent variable is intrinsically continuous, capable in principle, if not in practice, of producing measures carried out to the n^{th} decimal place.
3. The measures of $X_{1,i}$ and $X_{2,i}$ have the properties of at least an ordinal scale of measurement, so that it is meaningful to speak of "greater than," "less than," and "equal to." .

The randomly and independently drawn criteria are covered by the randomisation applied in the initial execution of the experiments. Further the data is continuous and all the performance parameters where measured using the same equipment that accurately records time etc. to the n^{th} decimal and is ordinal in nature.

3.7.3.2 Friedman Test

Another parametric test that is frequently used for analysis in 2^k is the Friedman Test. Although seldom used for scenarios where $k < 3$ it can still be shown to be applicable. Although also being conservative, the use of post-hoc analysis makes this a robust hypothesis test. The test was developed for use in economics but in recent years has been shown to be successful in a multitude of applications. The Friedman test is similar to a parametric repeated measures ANOVA.

The reasoning behind being able to use a repeated measures test is that the user is seen as part of the system and that the combined effect of user within system forms part of the total system performance. Thus using 8 users within the system can be seen as 8 measures of the system performance, although this comes at a cost of not being able to make any inference of the performance differences for a given user demographic.

The Friedman test is based on the following hypothesis.

- H_0: The distributions (whatever they are) are the same across repeated measures.
- H_1: The distributions across repeated measures are different.

The test statistic Q for the Friedman test may be approximated as a Chi-squared distribution. For small sample sizes it is important to note that the value of χ^2_{crit} must be read off from a table with degrees of freedom equal to (Number of observations - 1).

Nomenclature:

n	= Number of participants (observations)
k	= Number of treatments
Q	= Friedman test statistic
R_j	= Sum of the ranked scores in each column, ranks by treatment
SS_{bg}	= Sum of Squares, between groups
n	= Number of observations per treatment
T_i	= Total rank sum of treatment i
T_{all}	= Total of all the rank sums, total of all the treatment totals

The formalised procedure for the Friedman test is as follows, Sheldon *et al.* (1996):
First identify the factor and its various levels that must be investigated, then identify the various treatments that are under investigation, as the investigation will focus on system performance and not specific user but a specific combination of factors. Thus the aggregated group means are calculated and used to rank the data, an example is shown in Table 3.7.4. Ties are dealt with in a similar way as in the Wilcoxon (SRS) test discussed in Section 3.7.3.1.

For the example the original data is shown in the left section of Table 3.7.4, with the equivalent ranked data set shown in the right hand section. As can be seen the data is ranked for each user (marked observation) and ranked from the smallest to the largest across treatments. The mean and sum of each treatment will be used in further calculations; these values are calculated using the ranked data set.

Table 3.7.4: Friedman example table, for aggregate group means

	Original Measures				Ranked Measures			
Treatment / **observation**	1	2	3	4	1	2	3	4
1	5	2	3	5	3,5	1	2	3,5
2	8	5	4	4	4	3	1,5	1,5
3	9	3,6	1	8	4	2	1	3
4	10	1	7	0	4	2	3	1
5	11,5	8	6	1,2	4	3	2	1
6	12	9	5,5	9,8	4	2	1	3
7	16	15,2	18	5,7	3	2	4	1
8	7,6	6	4	6,5	4	2	1	3
9	16,9	7	9	3	4	2	3	1
10	1,1	9	3	4	1	4	2	3
Mean	**9,5**	**6,5**	**4,75**	**4,5**	**4**	**2**	**2**	**2,25**

For each treatment the mean of the ranked data, as well as the sum of the ranks are calculated. The next step will be to combine the data into the aggregate group mean table, shown in Table 3.7.5. This table will be used for the rest of the calculations.

Table 3.7.5: Friedman example aggregate group mean table

	Treatment 1	Treatment 2	Treatment 3	Treatment 4	All
Number of Observations	10	10	10	10	40
Total	35,5	23	20,5	21	100
Mean	4	2	2	2,25	2,125

Next the tests statistic Q may be calculated. This may be done using, Friedman (1937) which showed that Q may be calculated using equation 3.7.20. This is then compared to the χ^2_{crit} value from the table for $(k-1)$ degrees of freedom and the desired alpha level.

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3N(k+1) \tag{3.7.20}$$

Equation 3.7.20 is the contracted form the alternative equation of equations 3.7.22 and 3.7.21.

$$SS_{bg} = \frac{\sum_{i=1}^{i=k} (T_i)^2}{n} - \frac{T_{all}}{nk} \tag{3.7.21}$$

$$Q = \frac{12SS_{bg}}{nk(k+1)} \tag{3.7.22}$$

For the testing scenario presented in the combined user-system performance for the developed telerobotic system, the difference in treatments will amount to the differences in factor levels. To identify the factors and whether or not they have an effect on overall performance only one factor will be varied for a fixed combination of the remaining two factors.

3.8 Technical Verification

Technical verification is a key part of any SE process. Each model of the SE process has some form of technical verification process.

When considered in context of the UCSD systems engineering approach followed in this project as explained in Section 3.2.2. Verification is a process component followed throughout the system development process; this can be applied in both software development and systems hardware development. As was shown in the Chapter 4, verification can be tested on many different levels.

The experimental verification of system hardware performance will focus on the force response and force-following characteristics of the system. Force balance is used as the control variable and to drive the kinetic model used for motion command generation, thus it is very important to verify this.

3.8.1 Haptic Response

Haptic response is a combination of both accuracy (force response) and time response. For the current system, there is a trade-off between speed of response (rise time) and the accuracy with which the force can be reflected (error). The system is also very sensitive to impulse inputs, this is a hardware limitation of the F/T sensors and the control system implemented.

Haptic response will be tested in two different scenarios, so called elastic environments and rigid environments. This forms part of a combination of Section 3.8.1.1 and Section 3.8.1.2. The response will be tested in a pre-load modality. This implies that a desired force/load is applied prior to system activation, which can be likened to a step input for general control systems.

3.8.1.1 Motion Control

The system goals dictate that small-field-of-operation should be considered; further the system model was adapted to resemble an over-damped "stiff" system. This increase robustness to sudden changes in both input and response force resolution vectors. The magnitude of the response can be tuned to some degree using the build in "K_p" value field in the control console. This console allows for tuning of all the PI controller variables. Although a default set has been designed and tuned for the current system. This will not be available to end users.

3.8.1.2 Proportional, Integrator Controller Testing

The proportional controller's development was explained in full in Chapter 4. This section will focus on the tuning procedure that will be followed to optimise the response of the system for both soft-body (elastic) impact and rigid body impact. When "tuning" a PID controller one uses the design controller parameters as a starting point. These values will seldom be exact as some assumptions are necessary when calculating the transfer functions and the order of an unknown system.

Kp was first varied to establish a sufficiently fast response to the maximum expected input force (20N). An error band was set at 0.5N, within this band the integral portion would be added as to negate the integral windup that was observed in for the system.

3.8.2 Directional Accuracy - Comparison

Testing was conducted by J.Pretorius in Pretorius (2012), which asserted the three dimensional force accuracy of the system. In order to show whether or not the system performs better than the initial system, the 3D force accuracy needs to be assessed. This is also a response based measure, which will be a discretionary result, i.e. whether or not the result is truly better or worse will be at the discretion of the researcher.

3.9 Soft Body Impact and Pre-load Testing

Softbody environments have elastically deformable interfaces, i.e. the environment is made of soft materials, that give way and deforms. The testing was considered in two modes, although the second can be considered more determining a system characteristic rather than usable data.

3.9.1 Static Pre-load Test

The static pre-load test the SDA10 robot will be moved to a position above a sponge that is of uniform density, giving back a uniform pressure for a given compression force. This test was designed to test the system response to soft body impact recovery.

The soft body impact recovery is seen as necessary as there exists many sensitive surfaces that might be further damaged if the robot retracts too quickly. The robot will be moved manually so that the tool compresses the sponge. The position of the tool head will be determined by the reflexive force from the sponge, the response will be tested at four levels of pre-load. These are 1N, 2N, 4N and 14N respectively. The system response for the given preloads will be measured and compared for any trends. Further the steady state 0N time response will be measured to determine the baseline response of the system.

3.9.2 Dynamic Pre-load Test

The inverse of the previous test will be conducted by pre-loading the input side force torque sensor. The system will be put in a home position with the tip of the tool 40mm above the soft body; this position will be referred to as the home position.

The input side will be loaded with the same test weights of 1N, 2N, 4N and 14N. The haptic joystick function will then be engaged. The dynamic response of the system will then be recorded along with overshoot at varying speeds and the time required till equilibrium is reached.

3.10 Task Design

Task design, although assumed to be simple, is one of the aspects of a usability study which is easily underestimated (Perfetti, 2010). Task design must be considered in the planning phases of the usability study as it should incorporate the following:

1. What are user goals for the product/service?
2. Which potential problems do you want to investigate?
3. What data will be needed in order to evaluate the properties you which to investigate?
4. What tasks will commonly be executed?
5. Special use cases of the product and how will the product react?

The task design process allows for much interpretation. This is one of the activities that gives the researcher much freedom, but also sensitive to misguided testing. The errors usually creep in early during the process, when goals of the research are not clearly defined. Further erroneous data collection might also cause delays and or misinterpretation of the results.

When considering the task design for the current system, the goals where clearly defined. System development and usability goals are aligned for the purposes of the study. The system goals were defined as:

- *To study the effect of vision mode in the presence of haptic and non-haptic input modes, in order to evaluate the change in user-performance for streamlining the telerobotic system development process.*
- *Collect user preference data to assist in the development of a decision matrix for both testing and development.*

As for point number 2, there is no problem that would want to be investigated for the development process; instead a trade-off would like to be generated. In other words the trade-off between vision mode and input mode would have to be investigated in terms of user-performance, although user performance cannot be directly measured, as there is no standard model for user performance. User performance will be affected by the situation, mental state and other situational variables. For this reason, we look to heuristic approaches. The result is an inference of user performance.

The heuristic performance measures which have been identified as most relevant to user-performance and which have been identified in Table 3.10.1.

Table 3.10.1: Heuristic Performance Metrics

Dimension of Performance:	Unit
Task completion time	Seconds
Number of system errors	Integer Value
Number of recoveries	Integer Value
Number of collisions	Integer Value
Severity of collisions	Ranking Value 1 - 5

These measures can easily be recorded during experimentation, using automated logging processes. Some will have to be recorded manually as the measure or rank will be a subjective impression as is the case for the "Severity of collision" measure.

For the last stage of task development, steps 4 and 5 where combined. Firstly we consider step 5 in context of how the system will be operated. Each case can be considered as a special use case. There will be some commonality between operations, as the system has a fixed instruction set. This gives rise to the activities in step 4. The process of how the activities can be identified can be observed in everyday life. The activities most commonly associated with working at a bench (as is the case for small-field-of-operation) are as follows:

- Gripping / Picking up objects in order to move them.
- Moving objects in a 2D plane.
- Moving objects in a 3D environment, from one height to another.

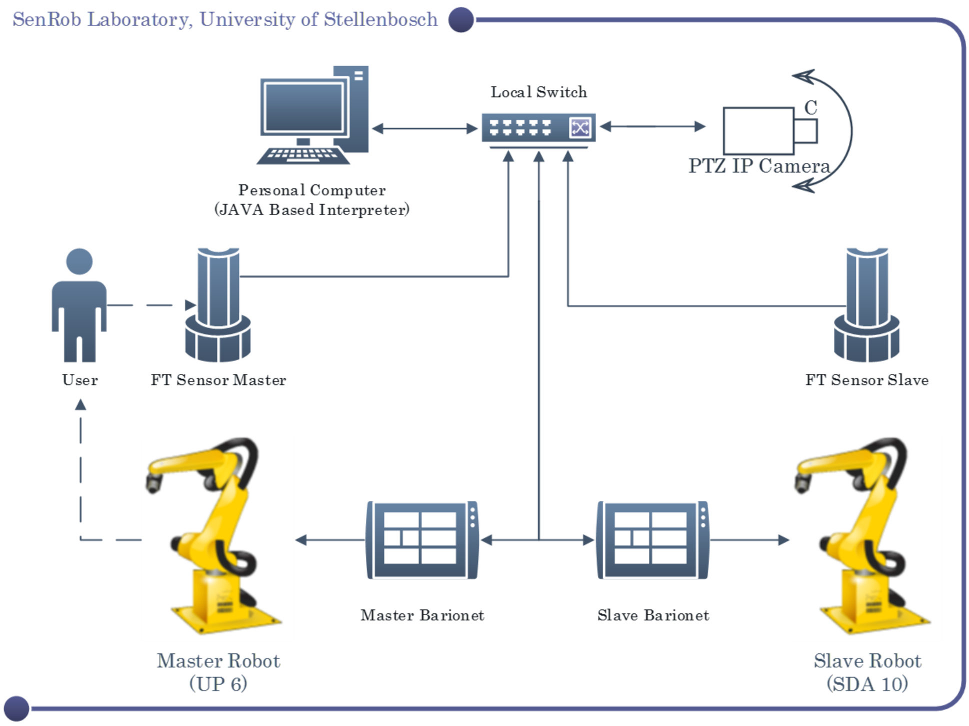
With this knowledge the experiments and experimental tasks described in the rest of this chapter have been designed. These tasks will collect the required data to assess the changes in user performance with changes in both vision- and input mode.

3.11 Experimental Setup

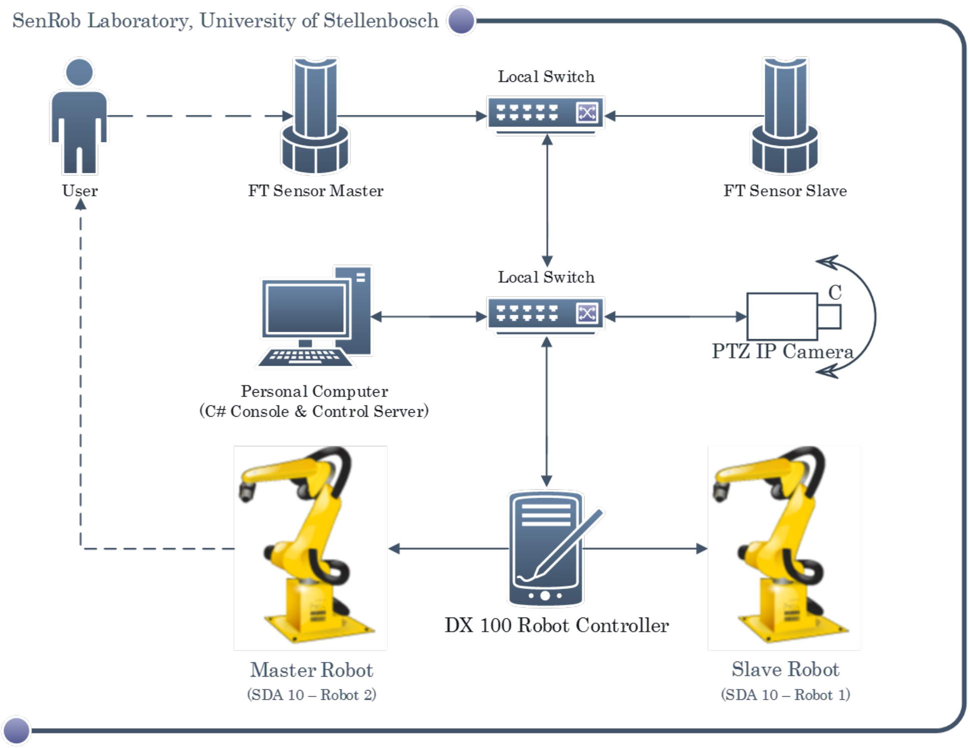
The original implementation was discussed in Section 4.1.1. An telerobotic system was implemented on a closed one gigabit (1Gb) local network using CAT-5 RJ45 communication cable in a crossover configuration, technical configuration available in Appendix G.

This enabled the ability to control and maintain a controlled communications network. The network is used for the network traffic for control signals along with high density multimedia such as video and audio. Figure 3.11.1 shows both complete experimental network diagrams. Although in the initial implementation UDP had to be used and in the current system TCP/IP is used except for the F/T Sensors.

First we consider the robots used, a Motoman SDA10 with DX100 controller was supplied, this is a 15 axis anthropomorphic robot. The SDA10 is a collection of 3 smaller robots, two identical 7-axis robots and a third external axis that acts as a torso. The robot is classified as anthropomorphic as is closely represents the degrees of freedom of a person, which is advantageous for natural manipulation in telerobotics. It is necessary to clarify that the system communication is classified as bilateral as there is two-way real-time communication. But the robotic control scheme is classified as unilateral as only one arm (robot) is manipulated on the SDA10 at a time, although the time required to queue a command for the a specific robot is negligible.



(a) Original Implementation



(b) Current Implementation

Figure 3.11.1: Network Diagrams for experimental setups

3.12 Comparative Study

A comparative study will be conducted in order to establish the effectiveness of the implemented changes to the control architecture (redesigned haptic model - incorporates orientation) as well as the software and hardware improvements that have been implemented. It has been shown by T. B. Sheridan that vision delay and execution delay severely reduces user performance as it forces a user to adopt a move-and-wait input style.

The pose model was reconsidered and re-developed to incorporate orientation modelling about a point. This brings the model closer to a natural 6 axis system which is analogous to the human arm-wrist-hand paradigm. The previous model as developed by Pretorius (2012) was found to be inadequate and outdated as it relied solely on an absolute reference frame co-ordinate shift to achieve motion within the work volume and focused merely on pose and did not include orientation.

Further comparative testing was conducted for technical performance measures and will be discussed fully in Section 5.1.2.

Chapter 4

System Development

4.1 Telerobotics at the University

This section guides the reader through the architecture and design of the existing telerobotic system developed in the initial implementation of the telerobotic system development by J. Pretorius. This also shows why some shortcomings / limitations need to be addressed and to reiterate the importance of major design decisions and system features.

The current project is the second iteration in the design and implementation process of the telerobotics development program. A continuous improvement approach has been adopted with respect to overall system development. For research purposes, this approach is desired as it can respond to changes in technology and user needs as they arise, to help develop relevant technologies or systems.

Further the approach was found to be appropriate as the iterative nature of fast system development, as there is limited time for each development cycle, presently set at 18 months. As new technologies become available and further development options are explored the system may be improved. Valuable lessons can be learned and limitations of telerobotic systems can be explored. In order to fully understand the improvements introduced by the second iteration, one first must have an appreciation of the work that has gone before it.

The discussion will refer to two main sections of work:

- **Initial System** - The work completed in the previous (*first*) implementation.
- **Current System** - The work currently under development / completed.

To fully comprehend the scope of work, one must consider both projects independently. Although the telerobotic system as a whole is the end product, each followed a distinct developmental process and evaluation. The initial system was focussed on developing a telerobotic system that proved that the utilisation of industrial robots is a viable alternative to custom developed hardware systems. The current system focuses on a user-centered design approach, with a focus on usability of the improved system as well as the redevelopment of selected sub-systems and the addition of new sub-systems.

4.1.1 Initial System Background

The ground work for the current project was laid over a number of years. The original focus of the research group was the development of new and improved robotic systems using standard software and hardware components. This idea later evolved into the current research focus of telerobotics or remote robotics under the supervision of Dr. A. F. van der Merwe.

Telerobotics has been the focus of many researchers since the early 1950's; Sheridan (1989) reviews a complete history of the various topics covered by researchers in the early years. The long term goal of the research being conducted at the university is to establish a real-time, transparent teleoperation test-bench which fully immerses the operator in the remote environment. The aim is to allow the user to perform dexterous tasks in the macro, SFO and micro-manipulation environments.

Historically one can consider the general unilateral telerobotic model. As shown in Figure 4.1.1 the model for unilateral control requires a master ($INPUT$) that controls a slave ($OUTPUT$) and is connected by a communication channel of sufficient bandwidth as discussed in Section 2.2. Here a uni-directional communication is shown, this communication architecture is characteristic of early implementations.

Figure 4.1.1: Master - Slave one-way communication

A conceptual prototype system was tested in a pilot project conducted by Opperman (2009). The system allowed for the loading and execution of pre-programmed robot jobs using a web interface. This system employed the first implementation of online video feedback from the remote environment. Further feedback was given by the interface website which allowed users to query joint positions and check the status of the robot. For this application the master was a desktop PC and the slave was the Motoman UP-6 Industrial robot with an XRC controller (for technical information see Appendix C). Here the use of a Barix Barionet as a local-host-controller and transducer for local communication was first introduced. The function of the Barix Barionet will be explained in more detail in Section 4.1.3.3.

After some time the focus of the research moved away from pure robotics for manufacturing and flexible production and the need for a fully developed telerobotic system became more apparent. This prompted the development of rudimentary online control for manipulation of a mounted Internet Protocol (IP) camera. The system was developed using a PTZ network camera for use on the local network. The network camera also known as an IP camera (shown in Figure 4.1.2), enables a live stream to be accessed using multiple video streaming formats using any PC on the network. This streaming capability is key for arbitrary node access of the video stream. Video streaming can be rerouted to numerous streaming services and clients making the video feeds more accessible when monitoring is required. Inevitably there will be delay present in the final online implementation. One design restriction placed on further development was that video streams and communication systems must be implemented on a local isolated network which will allow for control over communication variables and access to data.

To this end IP cameras were used for all future development. IP cameras are uniquely suited to be used for telerobotics. They are flexible in the sense that the data stream can be captured locally, as is the case in the current research. Or the stream can be access remotely or be sent to a third party service, although all these have their own set of drawbacks and limitations.

Figure 4.1.2: Vivotec PTZ camera used for initial implementation

In order to understand the telerobotic system's design process and evaluation thereof, one first needs an understanding of the **initial implementation** of the telerobotic system. The rest of this chapter is dedicated to defining the **initial system** on both the hardware and software levels.

4.1.2 Initial Bilateral Implementation

The re-purposing of standard industrial Robots was encouraged by project goals, industrial robotic systems are standardised systems that can be bought off-the-shelf and pose formidable accuracy and repeatability capabilities. Industrial robotics have been implemented in telerobotics before (Ferretti *et al.*, 2009),(Li *et al.*, 2011). From a life cycle cost perspective, this also reduces life-cycle cost and the liabilities that come with maintenance and training of operators. These systems also have the added benefit of having maintenance and support structures in place globally, which allows for third party service contracts and maintenance, further reducing risk for the system operator.

As a result industrial robots are attractive alternatives to custom specialised robotic developed for task specific applications when developing a telerobotic system. The use of industrial robots also negated the need for development of robotic actuators. Software integration was also eased as the robotic systems are supplied with software development tools for stand-alone application development.

Access to low level functionality has been made possible by flexible API and SDK such as the Yaskawa Motoman MotoCOM package that allows for a multitude of programming languages to interact with Motoman robot controllers (YASAKAWA Motoman, 2011). This allows for independent software development and promotes modularity of software and hardware design, which was shown by Sheridan (1989) to be a crucial component when developing a system for testing teleoperation.

4.1.3 Hardware

The Initial implementation relies on a centralised interpreter in the form of a Personal Computer (PC). The complete initial system layout is shown in Figure 4.1.3. The interpreter was developed in the java programming language. The software was developed in one layer.

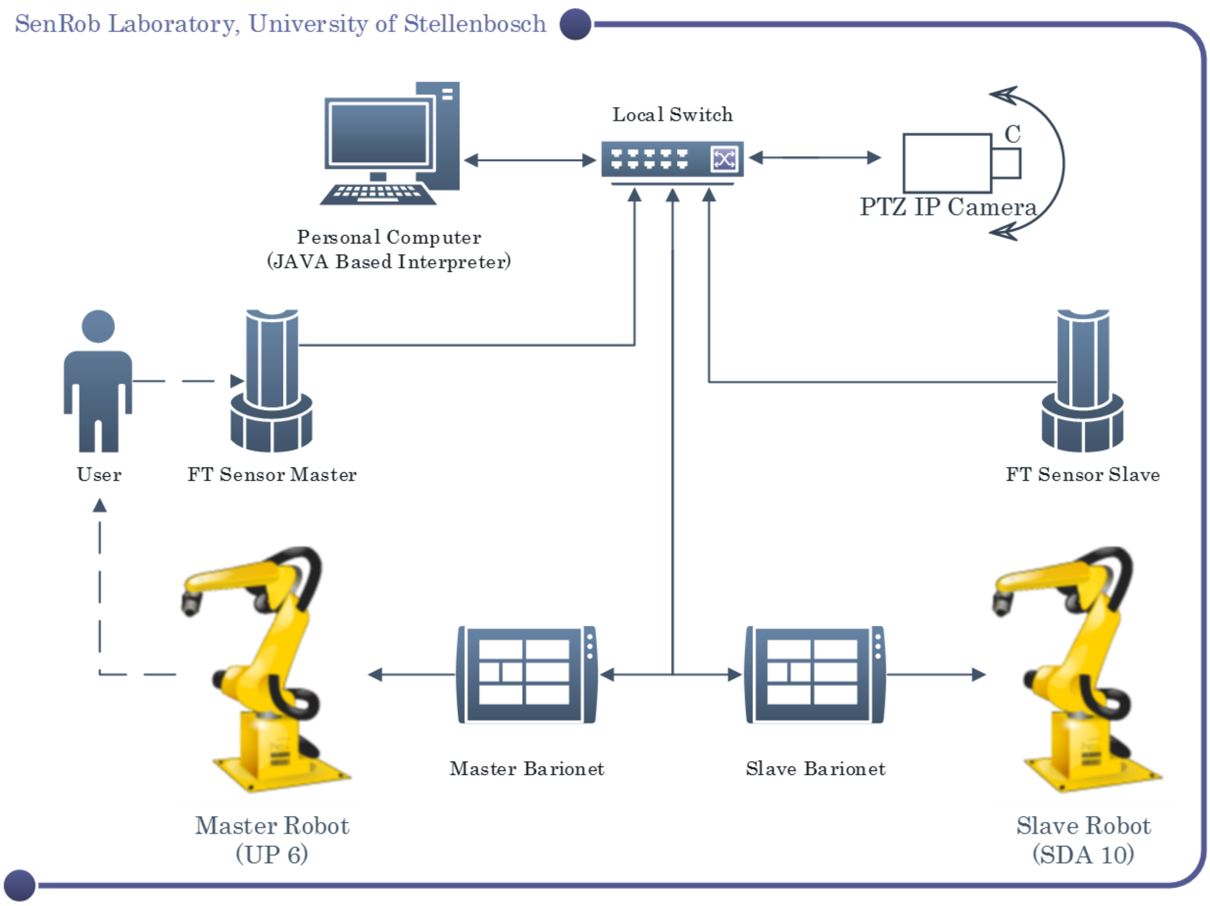


Figure 4.1.3: Initial System Layout, adapted from Pretorius (2012)

4.1.3.1 Hardware - Robots

Robots come in many different forms and configurations. Robots range from simple line following robots to complex personal care robots. When most people think of a robot in existence today, they think of industrial robots, in particular articulated robots. In general robot actuators are classed according to their main joint configurations. Six types are commonly used in industry. They are Articulated, Cartesian, Cylindrical, Polar, SCARA, and Delta configuration robots to name only a few (RobotWorx, 2013), some examples are shown in Figure 4.1.4. A generic robotic system consists of three components. They are the robot actuator(s), robot controller and programming pendant respectively.

(a) Cartesian (b) Articulated (6 DOF) (c) SCARA

Figure 4.1.4: Examples of Robot Types, (Bucknell University, 2012; robotmatrix.org; robotmatrix..org, 2012)

Robot controllers house the Central Processing Unit (CPU), switch gear and logic controller cards that control servo-motors on the robot. Robot controllers are the brains behind the brawn, Controllers have the ability to calculate position, velocity and acceleration for all the servo-motors, joints and peripherals such as tools or grippers. The controllers also house software and hardware components of the robot-specific control system. These advanced control systems allow for precision movement which compensates for wear of mechanical components as well as calculate the appropriate velocity and acceleration profiles for a given payload as not to over stress the mechanical components of the robotic system.

Commonly known as robotic arms the robotic arm is a collection of levers and actuators. Actuation is usually provided in the form of servo motors, each servo motor in turn drives a gearbox that corrects the output either increasing or decreasing torque or speed. The configuration of joints define the type of robot, the six named above are just some examples of robots, each type can be applied more effectively than others depending on the application.

Other than a few control buttons the controller has no user interface. The user interface is presented by the programming pendent (Figure 4.1.5b) and is the operator's primary interface with the controller. The pendent allows for various modes of operation of which the most commonly used is the *"play"* function in an industrial setting, the differences in the various commands are shown in Table 4.1.1. The teach function allows for manual manipulation of the robot joints at various speeds. Each position can be saved and the motion type between consecutive coordinates can be specified and saved. This allows for fast *"on-the-fly"* programming and changes to the master program or the current job.

(a) DX100 Controller (b) Programming Pendant

Figure 4.1.5: Basic Components of a Robotic System

Table 4.1.1: Command Modes Explained

Command Mode	**Description**
Teach	Allows for manual manipulation of joint position in various frames of reference via the pendent. This command mode is used to construct a robot job manually. The robot is moved to a position and the motion type is saved and stored in a job.
Play	Allows preprogrammed jobs to be executed as programmed.
Remote	Allows reading and writing of the controller variables, jobs and various controller parameters via the remote function. Communication can be via Ethernet or RS232.

For the initial implementation, the master input robot is a Motoman UP6, an articulated robot with 6 Degrees of Freedom (DOF) with a carrying capacity of 6 kilograms (See Appendix C for technical data). The UP6 is controlled by an Motoman XRC controller similar to the DX100 controller shown in 4.1.5a. In the initial implementation the controller was only able to communicate via serial (RS232) communication. In order to enable distributed communication, the use of local web-host controllers was deemed necessary. The web-host controllers allow for flexible communication solutions, which were implemented using a set of Barix Barionets. The Barionet and its function will be explained in more detail in the following section.

On the slave side a Motoman SDA10 bilateral articulated robot with 15 DOF and a payload carrying capacity of 10 kilograms was used (see Appendix C for technical information). The SDA10 (shown in Figure 4.1.6.) has two arms, each with 7 DOF and an external axis which acts as a torso. The SDA10 uses the latest generation DX100 controller (see Appendix C for technical information). The DX100 controller has the added advantage of direct network integration capabilities and an on board web-server that allows real-time variable monitoring and control-data reading, while maintaining a stable platform for the robot's internal processes.

Figure 4.1.6: Yaskawa Motoman SDA10D,YASAKAWA Motoman (2012)

This implementation poses a challenge in that feedback cannot be given using encoder or torque values returned from the robots for the control system as implemented in related research (Glover *et al.*, 2009). For this reason, the system was modeled as weightless, in order to reduce the problem to that of a simple rob model. It was later found that this assumption cannot hold due to safety concerns. If the input system is to be modelled as a simple rob only, safety of the operator and work volume will have to be set aside for performance requirements to be met. This assumption also had the negative effect that the speeds of the robots will have to be much higher than what can be considered **safe** within the confines of the testing scheme. High speeds also caused system instability, as will be explained in Subsection 4.1.5. As a result the kinematic model was reconsidered and a new control system was developed, as will be explained in Section 3.8.1.2.

4.1.3.2 Hardware - Communication

Communication channels are the backbone of any teleoperation system. Without modern high bandwidth communication teleoperation as we know it today would not be possible in its current form. The advent of the internet and with a free architecture that allows for fast and smart routing of communication allowed for research in "*networked robotics*", while being fully scalable.

The initial system can be considered under the definition of a networked robotic system as applied to teleoperation (Sanfeliu *et al.*, 2008). Further as it has no reasonable level of autonomy or decision making ability, although not all decisions are made by the user explicitly thus it also conforms to the definition of supervisory control, albeit in a relatively loose manner. For the given supervisory control application the role of the user is that of a supervisor. Not all the decisions on speed and trajectory will be made locally and independent of the hardware, some filter based command correction will have to be applied as explained in Section 5.1.1.2.

> *Tele-operated, where human supervisors send commands and receive feedback via the network.*
>
> - *(Sanfeliu* et al.*, 2008)*

According to the work completed by Pretorius (2012), the communication system that was implemented for the initial system consisted of two independent communication loops. These were the video communication and control loops that carried all the high density multimedia independent of the control signals. The nominal bandwidth requirement (bandwidth overhead) is mainly associated with high demand components such as video and audio, while comparatively the bandwidth for command signals, can be considered as negligible. The communication loop for the control signals was bi-lateral communication, while the video / audio loop was unilateral.

For data security reasons, the system development was limited to an isolated Local Area Network (LAN). The same limitation is set for the current system that has been developed. The LAN is comparable to a Wide Area Network (WAN) in that the function and execution systems are the same. Although many of the inherent problems associated with Service Providers and infrastructure suppliers are avoided such as quality of service issues and limitation on data throughput as a result of fair-use policies. The systems were also developed on a LAN to give explicit and complete control over nominal transmission speeds and to monitor and control network traffic, which will affect the communication delays.

Network congestion is one of main causes of variable delay on communication systems. Network congestion and communication hardware add delays in control signals, these delays are random and varies significantly. These varying delays may cause erroneous control data to be generated, thus the delays must be controlled as far as possible as design and implementation of a advanced controller is outside the scope of this thesis.

Further the initial system utilised RS232 communication between the network interface hardware (Barionet) and the robot controllers. The RS232 standard allows for multiple speeds of data transfer, as shown in Table 4.1.2. The XRC controller can communicate via RS232 using one of two speeds, the first is 9 600 bit/s the other 19 200 bit/s, the communication system was implemented using the 9,6 kbit/s standard as it was found that the 19,2kbit/s transmission speed resulted in a far greater packet loss when tested with the larger communication system.

Table 4.1.2: RS232 Baud Rate Specification

Baud Rate (bit/s)
19 200
9 600
4 800
2 400

4.1.3.3 Hardware - Barionet

The Barix Barionet controller houses a simple web-server that can be accessed via Ethernet. Similar to a Programmable Logic Controller (PLC), the Barionet has digital and analogue inputs and outputs. The Barix Barionet is shown in Figure 4.1.7.

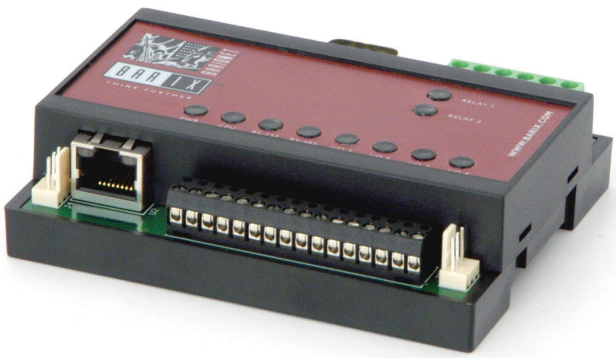


Figure 4.1.7: Barix Barionet

The Barix Barionet is one of the key components which have been used in many subsequent projects' newly localised intelligent control, in the form of a Barix Barionet. The local-controller was initially used to overcome a communication hardware and protocol incompatibility. The data that was sent to the controller had to be in the RS232 standard due to the XRC controller not having any alternative communication capability, while the operator was at an arbitrary location. Limitations of Serial communication (RS232) did not allow for large scale implementation. The local-controller was chosen as it can accept a RJ45 Ethernet connection as input and can function utilising both the User Datagram Protocol (UDP) and Transmission Control Protocol/Internet Protocol (TCP/IP) standards.

UDP as the protocol of choice for telerobotic applications (Resceanu and Niculescu, 2008; Resceanu and niculescu, 2007), as it does not require a handshake, as is the case with TCP/IP. This results in faster communication although it is much less secure. TCP/IP is more secure than UDP because packets are acknowledged, thus reducing or eliminating packet loss all together, but has the drawback that it is much slower than UDP systems. The differences in communication protocol are shown in Figure 4.1.8.

The local-controller would have to have localised processing capability that would allow for further thinning of the communication overhead bandwidth requirements. The Barionet is able to accept UDP or TCP/IP and relay the signal via RS232. This allowed for arbitrary node network communications to be implemented. An interface was developed that would allow for the user defined 3D location data to be entered and executed remotely using the Barionet's on-board web-server capabilities. This also allowed for ease of development in that the website is generated using a standardised interface.

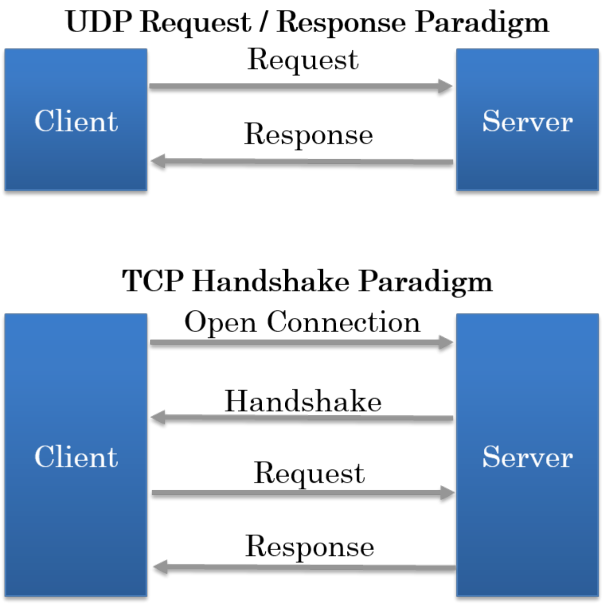


Figure 4.1.8: UDP and TCP Protocol Explained, Adapted from Sridhar (2011)

The programming language of the barionet (BCL Code) allows for easy processing and transcoding of input signals UDP to output serial communication (RS232). This data processing structure allows for event driven control to be implemented. As the name suggests event driven programming allows for a cyclic program to be interrupted to start execution of a subsegment of code as required.

The process requires some processing time and if many instructions where communicated the instructions would be queued. This in turn produced an buffer effect and introduced a execution delay in some cases. This buffer effect was not anticipated and had a negative effect on system performance when the initial testing regiment was completed. UDP although being the faster of the two protocols considered lacks the data security needed for the telerobotic system developed in the second iteration of the program. The use of UDP was allowed in the initial implementation in order to promote the fast development of a proof of concept.

4.1.3.4 Hardware - Haptic System

There are many modalities of haptic implementation. The generic name for these is a haptic display. Some are considered simple haptic displays such as a handle that gives varying degrees of force feedback, while twisting, (Brown, 1998). Many modalities such as touch feedback displays, levers, wire guided resistance frames or stringed haptic interfaces along with a plethora of wand manipulators exists for research purposes. Although these do have their limitations, some suffer from a lack of dexterity, limited feedback or feedback in only selected dimensions only. Others apply forces that are too strong or the device is too heavy. Further some are far to expensive to implement feasibly, (Williams II, 1998). Research focus moved to multiple degree of freedom systems in the early years and as a result commercially viable systems have been developed.

The need for multiple degree of freedom force feedback haptic displays prompted the development of universal systems that can be implemented in a wide field of applications. Some of the more popular systems are the PHANTOM 3 DOF system produced by SensAble technologies and the more powerful Delta Haptic device produced by VRLogic (2012).

The main developmental goals when considering haptics are, force tracking - the accuracy and magnitude of the force feedback and the rate at which these can be updated. The feedback scheme is also of importance. Some applications will be satisfied with vibratory feedback, where the magnitude of vibration is proportional to the severity of the impact/action suffered/taken. Further others will require only planar feedback, while the more advanced systems will require multiple DOF feedback.

Other modalities in haptic modalities are exo-skeletal systems that in two variations, the first is a rigid body structure that surrounds the user. This gives general feedback usually with point feedback, where the force is transferred to the user by straps or pads from the exo-skeleton. An alternative option is a articulated/pressure suite. The suite use shifts in fluid or inflation/deflation in conjunction with advanced materials to apply accurate forces on the user where it is required by the system.

4.1.3.5 Initial Haptic System

The haptic system was developed by Pretorius (2012). The system was developed using an ATI F/T Netbox sensors. Each sensor measures forces in the X, Y, Z axis along with the moment (Torque) about each given axis. The sensor outputs the analogue readings to a transducer Netbox, the Netbox captures the analogue signals from the F/T sensor and constructs digital data packets at up to 300 times a second. The data packets are broadcast using the UDP standard over any Ethernet.

The UDP broadcast was deemed suitable as the force / torque data is essential for control and requires no feedback to the sensor itself. The update rate of the system is fast enough that even if packets are lost, the next packet will update the control system, thus overcoming the expected problem of communication loss. An example of the data packet that is sent over the network is shown in Figure 4.1.9.

SOT	Fx	Fy	Fz	Tx	Ty	Tz	...	EOT

Figure 4.1.9: Example of FT Packet data

As can be seen, from Figure 4.1.9 the communication follows the standard packet handling protocol for UDP messages. The packet starts with a start of transmission (SOT) bit, which is followed by the transduced force and torque values and then by status information of the FT sensor, the transmission is then ended using the end of transmission (EOT) bit.

The force values generated by the software is sent in the format shown, the rest of the bytes are used as place holders in the initial and current applications.

4.1.3.6 Force Tracking and Articulation

Although force tracking was achieved and deemed satisfactory for the initial implementation, the system performance criteria did not fully describe the need for force tracking, especially force tracking response. Force tracking response is the time needed for the system to match the input (required) force. As the system was designed to use the force balance as the control variable, this is a crucial for system performance.

Force tracking is as crucial part in the teleoperation system as it is responsible for the most part for correcting errors in contact applications. As robots in general possess greater strength than their human counterparts the risk to the environment is much greater than when a human does the same operation. As a result the need for accurate force tracking in haptic implementations must be recognised as one of the more important criteria for system specification verification.

The initial system failed to achieve a usable response in that there was a response delay in excess of between 1.2 seconds and in some cases 6 seconds, depending on commands given. As a result delay compensated data was discussed in the final results. Although the response characteristics were found to be reasonable, the initial delay caused by the control system and slow hardware reiterated the need for a redesign or intensive improvements to the system.

4.1.4 Software - Java Based Interpreter

On a software level the system consisted of two distinct levels, as shown in Figure 4.1.10. The first is the local code that executes on the Barix Barionet itself. This level consists of two components. The first is the Barionet Barix Control Language (BCL) code which implements the operating system of the Barionet, defining variables and logic that is to be used. The other component is the Hypertext Markup Language (HTML) interface, the web-interface is generated using functions that imbed into BCL code.

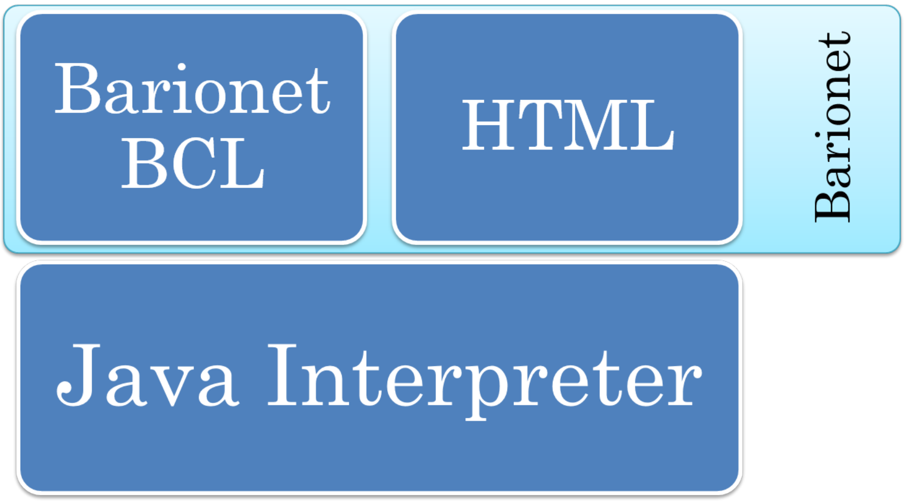


Figure 4.1.10: Basic Software Levels Explained

The second level is comprised of a Java based interpreter which acts as the user interface for the completed system and will be discussed in the following section.

The original system development resulted in a software based environment that was used to interpret input signals generated by the haptic telerobotic system over a local network. The system required some form of data capturing and decision making components. A software interface was developed using the JAVA development platform. This enabled the user to choose to store haptic data and gave the user the ability to observe the haptic data that was being sent and received.

Pretorius (2012) found that the BCL programming language along with the Barionets was not capable of achieving the computational accuracy that was desired. The move to the JAVA development platform was encouraged by the availability of sample code for interfacing with the F/T sensors. The Java development platform also has built-in libraries for incorporating video feeds, which was also required for user feedback.

The interface was used to display crucial force and torque data to indicate to users what the current state of the controller was. Further the interpreter collected force / torque data and interprets it by utilising of the basic direct controller. In turn the interpreter calculated relevant command data which is relayed to the respective Barionets. The implementation was found to be slow and unstable to any disturbance in the transmission of data.

4.1.5 Limitations of Previous Implementations

The initial system architecture follows a duplex communication design (Bilateral), including the local controllers and centres around a central interpreter personal computer, as shown in Figure 4.1.3. This system design allowed for full integration of both the signal conversion local-controllers as well as the peripheral sensors required for haptic feedback.

The initial system developed by Mr. J. Pretorius was developed for possible application in the field of tele-medicine, more specifically telesurgery. For these types of systems, high accuracy, repeatability and a high degree of safety must be guaranteed. The initial system is not a high fidelity system that can reliably execute tactile and kinaesthetic commands repeatedly. There is also a noticeable time delay present in the execution of the haptic commands. This is not acceptable when performing highly sensitive operations such as telesurgery.

Some hardware issues were identified, during nominal testing at the re-evaluation phase of this project. They pertain to the older of the two controllers, the Motoman XRC. The Controller also does not poses a network interface which will allow for fast Ethernet communication, which also prompted the use of the Barionet controllers. This will require a redesign of the communication system to incorporate an Ethernet-only design.

Some limiting factors were identified by J. Pretorius shown in Table 4.1.3, although not all have been analysed fully. The Hardware and software limitations must first be identified and analysed in order to find a usable solutions.

Table 4.1.3: Previously Identified Limitations by Pretorius (2012)

Dynamic Force Limitations:	
Response Time	Approximately 1.2 seconds
Slave Response to Sudden Impact	Slight overshoot from desired response, may also "overlook" sudden changes in force direction.
System Limitations:	
Controller Communication Errors	The system fails to communicate to controllers; this was attributed to old technology of one of the controllers namely the XRC.
Jittery Motion	The system executes jittery motion when moving; this was attributed to acceleration and deceleration of robot joints. Although it may also be attributed to slow processing and communication speeds.
Large Magnitude Inputs	A large input force results in an large displacement that can be hazardous as the robot is unresponsive to inputs while executing a command. The solution was to limit the maximum distance that a given motion command can be.

Another bottleneck for performance can be identified as the Barix Barionet. The transformation in communication protocol caused significant delays in the communication system. The effect of this delay was observed in a stop-go motion of the output robot. This has a negative effect on user confidence in the ability of the system to be used for small field of operation applications.

The initial interpreter structure resulted in a buffer being formed for the commands, where the user was unable to control the robot for the span of at least two (2) commands. Although following repeated testing, the system results showed that there is no set number of control commands that are buffered, suggesting that it is a timing dependent artefact of the interpreter and the physical system interaction. The communication backlog also resulted in a discrete robot control state, where the robot will continue to execute relative motion commands even when the user has stopped input.

Hardware limitations are of great concern when considering system performance. A limitation placed on the future development process is that the existing system be used as far as possible, serious system performance errors are to be identified and corrected as far as possible. This must all be done within a system design process that will allow a flexible, scalable and cost-effective solution.

The first hardware limitation identified in the initial implementation was the inherent controller command delay execution problem. The problem was identified in the command structure that allows for on-line control of the robots. The command structure requires that the controller firstly read the command into the command buffer, then the command is acknowledged and then it is executed. After execution the controller acknowledges that the command has been executed and that the buffer is free to receive the next command. This is for safety of the controller itself and is part of the INFORM programming language. The time delay between receive and acknowledge is command dependent, this is due to the size of the physical motion that is to be executed. System parameters also do not allow for interference of commands, this will result in a system error that has to be reset at the controller and cannot be done remotely, using the current controller and robot set.

Possible solutions include specifying new controllers. The high speed controllers that were identified are built to specification and allows for a 500Hz (interference) refresh rate. This allows for minor changes to the interpolation coordinates although this also requires specialised software. An example of such software is Kinematics-M will allow for advanced modelling and path planning algorithms. These controllers are built to order and as such are infeasible within the time frame available in for the current upgrade project, although future developers are encouraged to look into using such devices to allow for more accurate control.

4.1.6 Current Implementation Aims

The current system aims to be as close as possible to a six DOF articulated system. The proposed tool model set out by Pretorius (2012) in the initial implementation is that of a simple rod. This was found to be sufficient in the initial testing phase. The tool model helps to simplify the required modelling of the tool and allows the systems developer to form a concise base model upon which many different tool models can build. Although the model was found to be sufficient, there were some issues that were identified that needed to be corrected. Others only came to light after the fact and showed that the original analysis was incorrect.

The requirements for system development were deduced and from literature, the system specification requirements will be discussed in the following chapter. These values differ from that specified in the initial system development cycle as they were based on the system being totally weightless i.e. the system has no inertia and the movements are in real-time. As was found previously, hardware limitations make a *real-time* implementation nearly impossible at this time, although the reaction while in *joystick mode* is very close to real-time. This resulted in a loss of feeling and the decision was made to shift away from a weightless model. This weightless model was later replaced with a *stiff* model. The stiff model, when implemented, has a fixed offset and acts as a kinematic spring i.e. a spring absorbing some of the manipulation force. This modality allows the user to exercise more exact control over the movement of the robot as a result of this offset.

Developmental goals set out by J. Pretorius were that the force sensors used be able to sample the resultant force at a rate equal or greater than the human kinaesthetic perceptual band width of 20 - 30 Hz. Human sense of touch can be split into two very different types of contact sensing, simply put the pressure and tactile forces. The sense of pressure is also known as kinaesthetic sensing and the equivalent pressure refresh rate is much slower than that of the tactile response, as the tactile response is responsible for reactions to skin stimulus. The tactile network has an equivalent refresh rate in the region of 320 - 400 Hz. Pretorius (2012) also made the distinction between sensing band width and control bandwidth. The human nervous system works on a 8Hz refresh rate and as a result the human reaction bandwidth is much less than that of the equivalent sensing bandwidths, these are in the region of 1 - 5 Hz, but is dependent on the user's experience, and differs for periodic, internally generated and learned trajectories.

Two options for system development was explored, the first was to upgrade the initial telerobotic system. The next option was to use only one arm of the Motoman SDA10 for output and to use the other for input. This teleoperation system will be software based, with a centralised control dashboard that is located arbitrarily on the network. The second was to use a single controller to control a set of two robots with a external server.

Although the system aims at a natural interaction, intuitive input modality with as close to natural force reflection from the output feedback loop to the user input side. At the moment this is not possible. But many valuable lessons have been learned and can help guide future development.

The chosen system was developed using a ATI F/T Net sensor. This kit allows the user different options for integration with software and other hardware. The sensors allow for fast integration as it uses a Netbox that acts as a transducer for the incoming signals from the sensor unit itself. The signal is transducer to a information packet that gets transmitted via the UDP protocol on a LAN network. The supplied software development kit allows for packet listening and is capable of communicating to with many different software languages. This allows for fast flexible and agile software development and incorporation into a modular software architecture. Modular software architecture is a key part of the development strategy followed by the research group. The system will be improved on a continual basis and will help develop useful development strategies for future work.

4.2 Current system

As explained in Chapter 2, even a simple teleoperation system is complex due to the various fields of expertise required to develop such a system. The aim of the ongoing development project is to develop a low-cost modular, cost-effective and simple, usable telerobotic system that can be used as a test bench for core technologies. This will enable the development of decision tools for technology management, feasibility studies of such systems and will provide a stable base from which future research and development projects could be attempted.

The current system was developed within the limitations set in Chapter 1. A key limitation is that of the use of existing hardware as far as possible. This will inevitably impose limitations on any and all future development for which this limitation is upheld. This limitation is mainly due to the financial, development and training costs associated with advanced hardware or software implementations.

In order to expedite the development of the various sub-systems forming part of the final design. A system designer will first have to make some core design decisions. These decisions will determine the baseline configuration options, interface options and finally attainable system performance and is derived from the system requirements. The system performance will be the focus for evaluation efforts and aims to satisfy the requirements that were set.

For the current implementation it was decided that the system would focus on the simplest case of telerobotics. This use-case is direct user - environment interaction through the use of a modular telerobotic system. The basic system design is shown in Figure 4.2.1.

This design follows the basic Master-slave architecture. This architecture allows users to interact directly with the output environment. As can be seen from Figure 4.2.1, there are many sub-systems that must be developed in order for the system to qualify as a basic teleoperation system.

The feedback system which includes sensors, displays, kinematic modelling, and control systems, is the most important if the system is considered from the user's perspective. This sub-system feeds the user information about the tele-environment and allows the user to assess and react to that environment. The user can only complete a task as good as the information that is available to the user about the task.

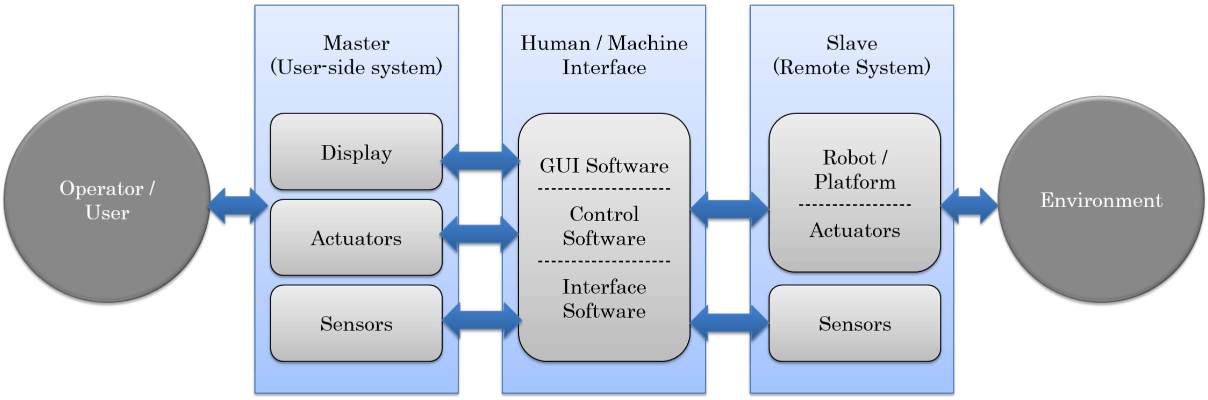


Figure 4.2.1: Basic System Development Framework

4.2.1 Vision System

User feedback is one of the most important aspects of any telerobotic system. This sub-system has many elements that must be considered, namely: audio, video and force feedback and client-side interfaces. This system is the direct link between the user, the system and the work (output) environment.

On a functional level the vision sub-system may be considered as a simple system. There are a plethora of options available to system designers. Among these are display technology, camera specification, 2D or 3D requirements and associated display technologies to name but a few.

This section aims to show the design decisions taken to accommodate the various research aims as development factors that will have to be taken into account throughout the design and development phase of the project.

The requirements for the vision system were defined as follows:

- Must be able to provide both 2D and 3D video feedback.
- Must offer two fields of view, one from the above and another from ergonomic perspective related to that of a user standing completing a task themselves.
- All combinations of views must be accessible independently or simultaneously.
- The display technology must be able to display both 2D and 3D stills and video.

For a successful telerobotic system to be feasible the degree of feedback needs to be sufficient. This implies that the user must be able to operate the robot in the tele-environment without the need for external assistance. The information presented to the operator must be of such nature that it gives positional awareness in the 3D environment or allows for an increase in Situational Awareness (SA) (Sonnenwald *et al.*, 2004). Also the operator must be able to assess the changes in the environment to make corrections to the execution procedure, faster and more accurately.

Since the beginnings of modern teleoperation (indirect observation), the primary feedback has been video as it presents the most natural feedback modality. The first implementation by Raymond Goertz in the early 1950's used CCTV cameras thus allowing the operator and tele-environment to be separated by an arbitrary distance. This was the first step in true teleoperations. The system also highlighted the problem that has been plaguing the research and development community for many years, the issue of latency. Latency is the time delay between input and output, modern switched network systems have randomly varying time delay and as such is very difficult to model. The logical conclusion is that the further the separation is between the two nodes on a given communication network, the more pronounced the latency. This in turn has an obvious negative effect on operator performance (Kaber *et al.*, 2000; Smith, 2008).

Video transmission is usually the most data intensive component of teleoperation systems. Control data can be streamlined and condensed using pre-processing, analogous to compression technologies; this may have the added drawback of increased latency depending on command complexity. Video transmission is mostly dependent on resolution as this is the multiplicative factor that determines the data requirements. Recent advents of improved sensor and display technologies has allowed for dense crisp images with higher contrast and reduced noise. These images are more pleasing on the eye and allow for high definition (HD) display.

Data requirement calculation for digital video transmission over a communication network is fairly simple. Uncompressed video is merely a set or stream of sequential still images. In order to calculate the uncompressed video data requirements we only require the following resolution, frame rate and whether the image is colour or not. The calculation in equation 4.2.1 shows the basic the required data rate (DR).

Data rate is a function of image properties, namely frame rate (FR), resolution (R) and Colour Properties (C). The colour property determines the number of bits required as shown by equation 4.2.2 is typically 1 or 4, this is due to how computers process colour. Computers interpret the colour as a set of values. If the image is Black and White (BW) the data required to generate it is only an intensity value, this is typically represented by a numerical value ranging between 0 and 255. For colour images this process is elaborated, to 4 values. The first three are the values for red, green and blue; the fourth is an intensity value, all these values range between 0 and 255.

$$DR = FR \times R \times C \tag{4.2.1}$$

$$C = \begin{cases} 1\ if\ Black\ and\ White \\ 4\ if\ Colour\ image \end{cases} \tag{4.2.2}$$

Standard video capture resolutions are shown in Table 4.2.1. For example, if the data required for a 15 frames per second black and white screen at 640x480 pixel (uncompressed) which is lower limit of natural viewing one would require 4.3Mbit throughput, where with 1920x1080 (HDTV) one would require 13Mbit. For a colour image at HDTV resolution one would require 52Mbit, which clearly shows the need for compression technologies to be employed.

Table 4.2.1: Standard Video: Size and Resolution

Size/ Format	**Pixels**
QQVGA	160x120
QVGA	320x240
VGA	640x480
HDTV	1280x720
1M	1280x960
1M	1280x1024
2M	1600x1200
HDTV	1920x1080
3M	2048x1536

Many compression formats for video exists today, each compression format can either be classified as ”lossy” or loss-less. Logically the lossy compression will result in less data that must be sent over the communication network, thus producing a lean overhead bandwidth requirement. The selective loss of data is part of the compression protocol for all ”lossy” compression technologies. There are many standard media codecs in use today the algorithms that determine the loss or compression protocol is called a codec. Of the protocols four have become standards in streaming media for network applications. They are namely:

- Motion JPEG (MJPEG)
- MPEG-1 or -2
- MPEG-4
- H.264

H.264 is a licensed technology, but comes standard with some hardware. This is a very desirable compression technology as it has low loss and high compression and decompression. Although some open standards such as MJPEG and MPEG 2 and 4 may also be considered.
Further there are many aspects of camera technologies that will have to be considered for system development. The most important of these have been summarised in Table 4.2.2.

The application of these concepts in requirements generation has a direct effect on system limitations. The decisions made for the requirements will also influence the cost of the system and as a result the feasibility of the implementation.

Table 4.2.2: Vision system concepts

Vision Component		**Definition**
Field of interest (FOI)	-	The area viewed that is of importance to the user, usually framed by the image.
Point of view (POV)	-	The attitude (position and orientation) of the camera, resembling the natural viewing angle of a human while standing and looking at the work volume.
Full field view (FFV)	-	An overhead view of the field of interest that covers the entire work volume and shows the relative position of the robot to the rest of the work cell.
Perspective	-	The appearance of viewed objects with regard to their relative position and distance from the viewer.
Depth of field (DOF)	-	The distance between the nearest and the furthest object that is in focus for a given image.
Viewing angle	-	The angle taken relative to the floor of the safety cell, which indicates the angle at which the work volume is being viewed by the user.
Field of view (FOV)	-	The extent of the observable world that is seen at any given moment.

4.2.1.1 Functional Analysis and Requirements Generation of Vision System

This study proposes to compare both POV and FF vision systems and how they impact the usability and task performance of the given haptic enabled telerobotic system for both 2D and 3D video feedback.

In terms of the visual components discussed in Table 4.2.2, the user requirements can be transformed into specifications. When considered within the development process discussed in Section 3.2, this falls within two categorical functions. Not only does it fall within requirements generation but can also be used in the alternatives generation phase.

First one must consider the developmental requirements of the research, which is to develop a scalable, cost effective, modular tele-robotic test-bench and the various research goals which have a direct impact on the requirements of the vision system. Secondly one must consider the various sub-subsystems that must be developed. This can be expedited making some design decisions early in the project.

The vision system can be broken down into three main elements, they are: the sensor (camera), required conversion hardware, and the display. The second - conversion hardware - might not always be necessary. The need for this hardware results from configuration management decisions that determine the interfaces, on both a hardware and software level. In the current developmental cycle, the decision to implement a software based telerobotic system, requires all video streams to be accessible over TCP/IP local area networks.

To ensure modularity of the system and ease of integration, some configuration management - specifically interface management - must be applied. To ensure modularity the system architecture was abstracted to that of a software based teleoperation system. Software control and integration will ensure that there is inherent system compatibility. For this the vision system must be accessible through software.

The vision system can be broken down into two main elements. The first is the sensor (camera) and the second is the display. The requirements and specification of these sub-subsystems will be discussed in the following sections.

4.2.1.2 Vision System Hardware Selection

The sensor components of the vision system are fairly simple to define. The sensor technologies in use today have been standardised as well as strict standards placed on output technologies.

The system requires that two main views be available at all times. As a result two cameras will be selected. The first will be the point-of-view camera. This camera was selected to function in both 2D and 3D viewing scenarios. The position was chosen to face the work surface at an included angle of 35 degrees. Evaluating the available cameras, the list was narrowed to three possibilities shown in Table 4.2.3. For reconfigurability, the Panasonic HDC-SD900 was selected as it has a detachable 3D lens (VW-CTL1). Although the camera can operate in 2D mode while the lens is attached, it does this at loss of zoom capability and resolution. The camera is also the most cost-effective of those identified and satisfies all the output, zoom, resolution and focus requirements. Further it possesses on-board H.264 compression hardware, thus reducing the streaming bandwidth requirement of the system.

Further one must note a property inherent to all 3D systems that will have an effect on user performance. The 3D recording hardware relies on the use of sophisticate optics that allow for the simultaneously recording of two images. The distance between the two lenses' optical centres is known as the interocular distance (IOD). IOD was shown to have an effect on operator performance for tasks in 3D environment, where a minimum IOD of 3 cm is required, Rosenberg (1993). For this reason, only cameras with IODs larger than 3cm was considered. The Panasonic HDC-SD900 with VW-CLT1 lens was selected, as shown in Figure 4.2.2.

Table 4.2.3: POV camera selection

		Camera Name	Sony HDR-TD30V	Panasonic HDC-SD900 with VW-CLT1 Lens	JVC GS-TD1
Evaluation Parameters		Estimated Price	R 10 000,00	R 9 700,00	R 15 000,00
	Interface	TCP/IP	×	×	×
		HDMI	✓	✓	✓
		2D/3D selective Output	✓	✓	×
	Recording and Streaming Mode	2D Recording	✓	✓	✓
		3D Recording	✓	✓	✓
	Capture Specification	Focus type	Full range - ∞	Full range - ∞	Full range - ∞
		Mountable	✓	✓	✓
		720p	✓	✓	✓
		1080p	✓	✓	✓
		Max Zoom level	10×	5× fixed when in 3D	5× fixed when in 3D
		Compression	MPEG 4, AVCHD	MPEG 4, AVCHD, H.264	MPEG 4, AVC/H.264

Figure 4.2.2: Panasonic Camera selected for vision system

Further a top-down overhead camera was required. Due to access limitations the camera would have to be operated from ground level. The need for a TCP/IP PTZ camera was identified. Unfortunately all the cameras available lacked the ability of 3D video streaming, although this type of camera has the advantage of having built in media server functionality and wallows direct access over TCP/IP. The video feed is also secured and requires authorisation to access.

This camera can also be controlled by making use of the on-board web server. The server grants access to higher functionality such as pan and tilt positioning as well as control over the zoom level and focus region, although the focus region is usually used as the default autofocus mode.

The camera selected is the Axis communications P5534 PTZ camera. This camera possesses H.264 compression along with MPEG 4 compression, further it does not require any external power as power is injected over the Ethernet and has an 18× optical zoom capability.

The vision system sensor configuration is shown in Figure 4.2.3. The two different field-of-views are also indicated. Please note that for 3D full-field experiments the Panasonic camera was moved to the full field position of the PTZ camera.

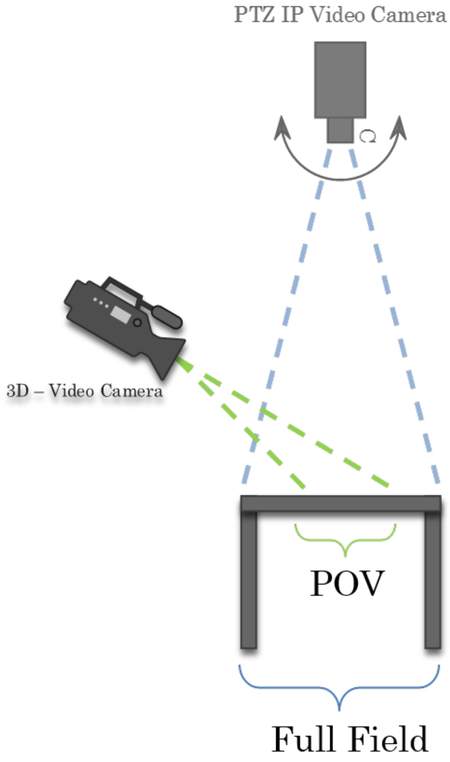


Figure 4.2.3: Vision system, camera layout with Field of view indicated

4.2.1.3 Display Technologies

Many options exist for display technologies today and they range from projection screens to Head Mounted Display (HMD) technologies as well as standard television technologies. The main developmental goal was to find a cost effective consumer level display that would be able to display both 2D and 3D images on demand. While being large enough to utilise most of the user's field of view as to increase the level of immersion of the user.

In line with the requirement of 2D and 3D display technology, one first must understand available 3D technologies. 3D video relies on the human brain to interpret the images shown to each eye. This is a product of evolution in that the each eye sees the world from another perspective as when compared to other. This allows humans to determine the distance to an object, its speed and the relative distances between the objects in view. 3D display technology also increases the bandwidth requirement of the system as two dependant images are now sent, instead of 1 per frame, which in turn will increase the reliance on compression.

As limits have been placed on the development of a new haptic feedback system, a complete redesign cannot be done to allow projection based displays, especially 3D projection as those used in the fully immersive CAVE system. The CAVE is a multi-projector visual environment, which utilises advanced filtering on polarised glasses to produce the illusion of 3D objects within the CAVE (CHRISTIE, 2012). The incorporation of such technology is not feasible as direct interaction with the robot will be difficult as the robot arm required for feedback will obscure some projections needed for the 3D functionality, furthermore it was not an economically feasible solution in light of other options.

HMD have become popular in research as it allows for a deep sense of immersion in the remote environment. HMD are also easy to use and understand, while it provides an ergonomic solution that feels natural to the user. 3D viewing is also easily achieved as each eye is exposed to a single video feed corresponding to the left and right views. The drawbacks of this type of technology are that most available hardware requires non-standard connections and software in order to operate. Furthermore the integration challenges will development time and thus further reduce feasibility for its use.

Television technology has come a long way in terms of 3D viewing. There are two main technologies in use for 3D viewing today. 3D technology can be broken down into two main categories they namely *active* and *passive* systems. The pros and cons used to determine the suitable display is shown in Table 4.2.4.

Active displays, utilises an active viewing glasses. These glasses utilises either blue-tooth or infra-red communication to synchronise with the television display. As the display displays the alternating views of the video stream, the viewing glasses functions as a synchronised shutter system. When the image pertaining to the left eye is displayed, the right eye is blocked out and vice-versa. This happens at such a high rate that the user is not cognitively aware of the process. The main drawbacks for this kind of technology are that the glasses require batteries and are thus heavier and more uncomfortable and do require charging. The shutter effect may also result in headaches in some cases.

Passive displays on the other hand do not require active shuttering to occur. The user is required to wear specialised polarised viewing glasses. This display technology has many configurations in which the images can be displayed but the most common is that sequential pixels are polarised differently, thus allowing both left and right eye images to be displayed at the same time and the viewing glasses to passively filter out the images. The main drawback of this kind of technology is that the user is required to be a certain distance away from the display and the angle at which the display is viewed will affect the 3D experience.

The solution that was chosen for the current system, is a passive 47 inch display. This was found to be a cost effective solution, which is easy to use and can take standard computer display inputs while providing the 2D and 3D capabilities required. The model of display is an LG 47LW6510; technical specification in Appendix G.

From a hardware layout perspective the IP camera will be mounted above the work surface, so that at a minimum it will have a full view of the work area. The data stream will be made available over the local network. The 3D camera on the other hand will be mounted to an height adjustable tripod, positioned so that it can be used to adjust the camera high above the work area as to assist in providing a 3D image of the work area.

The 3D camera can only output data via an HDMI connection. For this reason, the image data must first be streamed to appropriate capture and streaming hardware. As a remote side server will be used to channel communications and do processing, it was chosen to host the capture hardware and make the video stream available over the Ethernet.

The chosen hardware, must possess internal compression and make the capture stream available to the host. Further it must support AVCHD and MPEG 4 input standards and support the HDMI input standard at a minimum. Three options were identified, shown in Table 4.2.5, each with its own set of Pros and Cons. From experience PCI-Express cards outperform external devices that are within a comparable price range. External devices also generally require third party software to be able to integrate into developer applications. For this reason only the two PCI-E cards were considered.

From a development goal standpoint, the minimisation of the overhead bandwidth requirement is of utmost importance. Thus a lower capture frame rate would be advantageous, although user performance has been shown to deteriorate rapidly and task accuracy. According to Chen and Thropp (2007), user performance is acceptable above 15 frames per second (fps) should not be below 10 fps for a usable system.

Thus the frame rate band that will have a minimum effect of bandwidth but also be sufficient is between 15 - 30 fps. This narrowed selection options to the Black Magic Intensity Pro PCI-E card, as it full fills all the requirements set for the vision streaming system, namely, must possess adequate compression (supports H.264 and MPEG 4), Capture fps between 15 and 30fps, supports HD formats, and happens to be the most cost effective.

Table 4.2.4: Pros and Cons of television display technology

		Pros	**Cons**
Display Type	Plasma	Superior Brightness No backlight Progressive scanning Wide viewing angle Lower prices	Short life span Low contrast Heavy
	LCD / LED	Narrow to medium viewing angle High refresh rate capable Excellent contrast Progressive scanning available in large sizes Light weight	Backlit Loss of contrast at wide angles
3D display Type	Active	Higher resolution attainable Higher image quality	Flicker may be present Glasses must be charged Glasses are expensive
	Passive	Glasses are inexpensive Glasses are light weight	Lower image quality Small optimum viewing area

Table 4.2.5: Video Capture Card selection

Card Type	Estimated Cost	Connection Type	Connecting Standards	Supported Resolution	Supported Standards	Compression Standards	Capture Frame rates
Black Magic Intensity Pro	R 2 314,00	PCI-Express	HDMI DVI	720p / 720i 1080p / 1080i	PAL NTSC 720 HD 1080 HD	MPEG 4 H.264	24 25 30 60
BlackMagic UltraStudio SDI	R 4 594,00	USB 3.0	HDMI VGA 10 bit SD/HD	720p / 720i 1080p / 1080i	PAL NTSC 720 HD 1080 HD	MPEG 4 H.264	24 25 30 50 60
AVerMEdia DarkCrystal HD Capture	R 7 272,00	PCI-Express	HDMI VGA	720p / 720i 1080p / 1080i 640x480 - 1680x1050	PAL NTSC 720 HD 1080 HD	MPEG 2 H.264	50 55 60 72 75 85

4.2.2 Haptic System Development

The initial implementation of the haptic system was fairly successful. Although it should be clarified that haptic system developed focused primarily on kinaesthetic feedback, position and pressure response, while tactile response was neglected to some extent.

The hardware system implemented remains unchanged. The F/T Sensors are used as is with the supplied Netbox transducers. The Netboxes are operated in the UDP mode as before, this is to ensure that both the input and output side F/T signals remain broadcasting even if there is a problem with the server or controller. This is important to ensure safety of the user in case of a communication failure. The data collected may also serve for troubleshooting purposes.

The simple kinematic model from the previous implementation lacked the exact control needed for a high fidelity system and as a result will have to be reconsidered. The simple kinematic model was also not implemented to allow for rotation of the manipulator. For this reason the kinematic model and the resultant controller must be discussed.

4.2.2.1 Motion Derivation and Development of a Simple Kinematic Model

This section will focus on the modelling and experimental derivation of constants for the given robot system. The motion of both the input and output robots are to be mapped and matched to a simple kinematic model. This model expands on previous implementations in that the wrist angle will also be calculated and implemented in the final workable system.

The newly developed telerobotic system focuses around a software-based telerobotics implementation. This type of system shows promise for web-based telerobotics as software standards become faster and more reliable (Duff *et al.*, 2007). This type of system allows for development on a local network (LAN) that can later be expanded to a web-based platform (WAN) which is aligned with the development program's long term goals.

The implemented system can be modelled as a decoupled system, which can easily be deduced from the definition of a decoupled system (given below). If one considers each robot and the input mechanism separately with respect to force, there are three systems. Each only has knowledge of one another at their interfaces. Any change within the system does not affect the other. The only information exchanged over the boundary of the systems is forces and torques.

This is advantageous when considered in the context of software based teleoperation. The response to a given input and feedback set is generated by a software kinematic model. The final simple kinematic model is presented in Section 4.2.3.1, and is the theoretical response to a resultant force acting on a point in space.

Definition: *Decoupled Systems*
Two or more systems that are able to transact without being connected, or coupled. The systems do not interact with each other, and also one system usually has only a very limited knowledge of any other system, and that knowledge is usually limited to information about shared interfaces. A decoupled system allows changes to be made to any one system without having an effect on any other system. (Webopedia, 2012)

Similar arguments were made by Stark *et al.* (1987), in that weakly coupled or decoupled systems allow for simple direct and inverse kinematic calculations, as the case of the simple kinematic model. This also added to the degree of intuitiveness of the resultant system and proved to be advantageous as users can easily learn how to use the system.

The model is based on the rod in space, that crosses an interface plane. As shown in Figure 4.2.4 . The rod-plane intersection model is based on the user holding the rod like a pencil. The user's thumb and forefinger meet at the intersection of the plane and rod. This intersection point is the point around which forces and torque is applied.

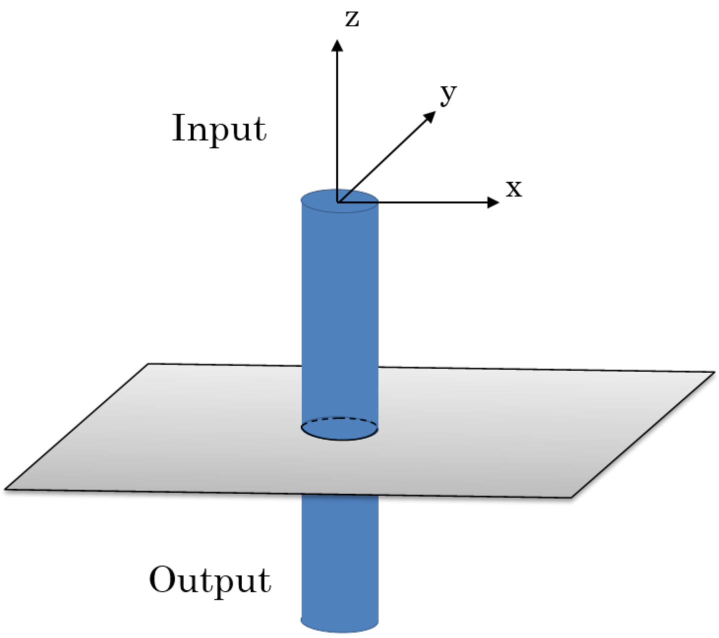


Figure 4.2.4: Rod, plane intersection model

4.2.2.2 Deriving Angle from Torque

Pose as interpreted from Pretorius (2012) relates to the position of a work tool in 3D space. This definition is rather ambiguous, thus for the purposes of this thesis, pose will be defined as the 3D Cartesian position and rotational position with respect to the robot base.

From basic dynamics one knows that when describing the motion of a system, two distinct sets of equations are applicable. These are the force-linear acceleration and torque-angular acceleration, which will be required to model the motion of the robot.

Nomenclature:

T	Torque	$N.m$
I	Mass Moment of Inertia	$kg.m^2/rad^2$
θ	Position Angle	rad
θ_0	Initial Position Angle	rad/s
$\dot{\theta}$	Angular Velocity	rad/s
$\dot{\theta}_0$	Initial Angular Velocity	rad/s
$\ddot{\theta}$	Angular Acceleration	rad/s^2
t	time	s

From basic dynamics, the general case for relating the moment (torque) about a point is given by equation 4.2.3. The relationship between applied torque, mass moment of inertia and angular acceleration is shown.

$$T = I \times \ddot{\theta} \tag{4.2.3}$$

Changing the subject of the formula to angular acceleration and integrating with time to find the angular velocity:

$$\int_0^t \ddot{\theta} dt = \dot{\theta} = \frac{T}{I}t + \dot{\theta}_0 \tag{4.2.4}$$

Integrating equation 4.2.4 again with respect to time will yield the equation for angle:

$$\int_0^t \dot{\theta} = \theta = \frac{T}{2I}t^2 + \dot{\theta}_0 t + \theta_0 \tag{4.2.5}$$

It can be shown that the resultant equations of orientation for a given 3-axis control volume is described by:

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{T_x}{I_x} \\ \frac{T_y}{I_y} \\ \frac{T_z}{I_z} \end{bmatrix} \frac{t^2}{2} + \begin{bmatrix} \dot{\theta}_{0x} \\ \dot{\theta}_{0y} \\ \dot{\theta}_{0z} \end{bmatrix} t + \begin{bmatrix} \theta_{0x} \\ \theta_{0y} \\ \theta_{0z} \end{bmatrix} \tag{4.2.6}$$

4.2.3 Deriving Displacement from Force

Nomenclature:

F	Force	N
m	Mass of input model rod	kg
a	Acceleration	$^m/_{s^2}$
s	Displacement	m
v	Velocity	$^m/_s$
t	time	s

For an arbitrary mass (m) to which a unbalanced resultant force F is applied it will experience an acceleration. From first principles one can deduce displacement from force, the basis is formed by Newton's second law of motion (equation 4.2.7).

$$F = ma \tag{4.2.7}$$

Integrating the equation and simplifying the velocity yields equation 4.2.8, by solving the equation at time equal to zero, it is found that for this system c_1 is also zero.

$$v = \frac{F}{m}t + c_1 \tag{4.2.8}$$

Integrating the above equation with respect to time again yields the solution shown by equation 4.2.9. The general solution for a 3-axis system is shown in equation 4.2.10.

$$s = \frac{F}{2m}t^2 + s_0 \tag{4.2.9}$$

$$\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \frac{t^2}{2m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} s_{x0} \\ s_{y0} \\ s_{z0} \end{bmatrix} \tag{4.2.10}$$

4.2.3.1 Kinematic Model Simplification and Discussion

This kinematic model can further be simplified within the framework of this implementation's specific hardware limitations. The robot command for movement is the IMOV command. This references the motion path relative to the previously executed command's final position. This means that for each command, the initial velocity of the robot used for the determination of the next rotation or displacement is always zero. Further the instruction set used generates a relative motion, which negates the need for the initial orientation of the robot arm. Further due to the command structure and data requirements of the controller the angle values need to be generated in degrees and not radians. The resultant simple kinematic model is shown by equations 4.2.11.

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{T_x}{I_x} \\ \frac{T_y}{I_y} \\ \frac{T_z}{I_z} \end{bmatrix} \frac{t^2}{2} \times \frac{180}{\pi} \tag{4.2.11}$$

As the movement is a relative movement, the displacement component of the kinematic model can similarly be simplified (shown in equation 4.2.11)

$$\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \frac{t^2}{2m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{4.2.12}$$

This simplified model also shows that for discrete time the displacement and rotation is linearly proportional to the applied force. In reality this is almost never true.

The master-slave pair must be modelled independently and a mapping process must be followed in order to map the input to the output. As J. Pretorius proposed in the first phase implementation the input side can be modelled as a thin rod, as this is the proposed input modality. This is done based on the assumption that the user requires a free-floating, weightless input configuration. For the current implementation the system will be operated in a stiff modality. This implies that the system is over-damped, which makes the system safer to operate for initial testing. As the input system is identical to the output robot - using one arm as input and the other as output - no cross mapping will have to be completed.

The kinematic models of the robots are proprietary information of the manufacturer. As a result the derivation of the models will have to be done experimentally, which is a complex task and the possibility of error generation is very high. If only the actuated robot arm is considered the seven axes each has an effective mass moment of inertia. The mass moment of inertia is dependent on the relative position of the other axis of the robot for a given motion. The derivation of the combined mass moment of inertia for a 7-axis robot will result in a 7 by 7 matrix of relative mass moments of inertia for every conceivable point in the work volume. Although an augmentation matrix can be calculated, the data available from the controller makes this nearly impossible.

The mapping process is user defined, with increments being specified by way of trial and error, this is done in a effort to increase the user perception of a fully ergonomic interface. System developer input will be used along with a small test pool, so that the input is perceived as natural as possible. This leads to the development of a matched system rather than a kinaesthetically accurate model. Initial testing revealed that this model is sufficient for user testing as a comprehensive control environment is outside the scope of this thesis. Although this model somewhat detracts from a natural input model as presented by J. Pretorius this model however can be feasibly implemented and tested and will have little bearing if there is a significant change in operator performance.

This model is crucial in the development of the software controller. The software controller is extremely important for the technical performance of the system. In other words the model is very important to the overall performance and usability of the system.

4.2.4 Development of a Control System

Upon careful consideration of the dynamics involved in executing the motion and the available input forces and torques, motion would have to be restricted if the need for an elaborate kinetic model is to be negated. In line with the current project limitations, the haptic system must remain as unchanged as possible. Thus the decision to further investigate and develop the software based interpreter and control module was taken.

4.2.4.1 Kinematic Model Simplification

In order to produce a near real-time control environment, the following simplifications could be allowed. This is system specific and is only applicable for the fast dedicated communication. If the system delay is variable as is the case in real-world situations some of these assumptions and simplifications may no longer be applicable.

The following assumptions where made:

- The focus is small-field-of-operation, thus only a small work volume will be considered.
- Only three torque values can be recorded thus it corresponds to three degrees of freedom which will be translated to a final workable angle increment.
- Time t is always a constant interval, the sampling rate of the F/T Sensor (10Hz). This was ensured by the control system in that the force sensors were only polled every 100ms.

The following is known about the robot-controller pair:

- The robot's internal control system compensates for variations in payload, resulting in a constant velocity profile over the entire work volume.
- The robot has a sequential execution strategy, thus the robot comes to rest before the next instruction is processed. Thus the initial angular velocity is always 0 rad/s and linear velocity is 0 m/s.
- The command structure of the robot follows a internally referenced relative position increment thus the increment and not the total angle and displacement is important thus the initial angle and displacement with reference to the model is always 0.

Taking into account these assumptions and the inherent robotic-controller command specifications, equation 4.2.6 can be reduced to yield the appropriate angle increment for each axis:

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{T_x}{I_x} \\ \frac{T_y}{I_y} \\ \frac{T_z}{I_z} \end{bmatrix} \frac{t^2}{2} \tag{4.2.13}$$

This model is applicable to an equivalent mass point and does not take into account the controller's internal control system. It is assumed that the robot will execute the generated command with an accuracy of 100 micro meters repeatable. The internal control system of the respective robots is responsible for constant velocity and accuracy of movement. This is achieved irrespective of payload up to a maximum of 10kg in the case of the SDA10. In return this shows that the internal control system compensates for any inertia effects. Further it can be argued that this shows that the internal control system compensates for the mass moment of inertia of every joint and limb, in order to give a specified accuracy when moving to a predefined point in space. The result is that the mass moment of inertia for a given axis from equation 4.2.13 is merely a scaling factor when considered in the context of the input system.

As the input system is decoupled from the output robot and the input is weakly coupled to the input robot, it can be argued that the mass moment of inertia used for the model should be that of the input tool. This was applied in the initial implementation of the telerobotic system and showed promise despite having performance issues due to time delay.

For the purpose of the research presented in this thesis, the scaling factor/ step size will be fixed, for all participants, and will be scaled linearly to the applied torque on the input joystick (thin rod).

4.2.4.2 Initial Control System Implementation

The original controller implemented by Pretorius (2012) is shown in Figure 4.2.5. Aptly named the intuitive controller, was based on a simple force balance principle. Further the distinction between active and passive control was described, *active control* was defined as a active (instantaneous) force balance approach, whereas *passive* described a more conservative hierarchical approach to the force balance. The active controller used the force difference as input on an instantaneous basis allowing the feedback torque and force to far exceed the input and allow inverted control scenario, where the slave is able to move the master. This was considered unsafe and thus the *passive* approach was followed. The passive approach does not allow inverted control, i.e. if the feedback force or torque exceeds that of the input the system comes to a controlled standstill.

This is not favourable for sensitive work volumes where sensitive surfaces are. The user did have the ability to convert the system back to an active system but under hard impact conditions the sudden force spike resulted in unstable motion control.

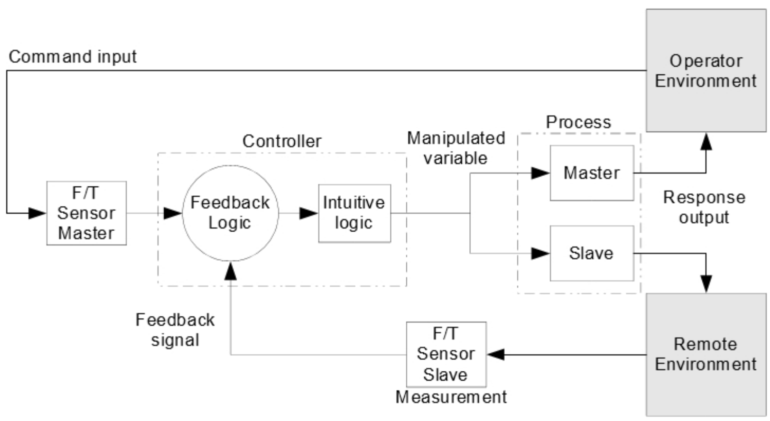


Figure 4.2.5: Initial Control Implementation from Pretorius (2012)

4.2.4.3 Newly Developed Force Controller

The current controller was designed as a simple PI controller. The basis of the active control principle (shown in equation 4.2.14) originally used is a force balance; the aim is now shifted to get F_{total} as close to zero. For F/T Sensor number 2, the axis are inverted so that the forces measured are positive in opposite direction when compared with sensor number 1, thus their addition calculates the difference.

$$F_{total} = F_1 + F_2 \tag{4.2.14}$$

Knowing that the original FT sensors and haptic interpreter system had to be used in the new implementation an effort was made post communication upgrade to improve the force error response. The force and torque values can be polled at any time, with a maximum refresh rate of 1000 Hz reliably. Thus force control had to be implemented as the position polling of the robot could result in further system delays and delay the entire control system. Force updates run on a separate network stream and do not require feedback from the relatively slow robot controllers.

4.2.4.4 Controller Design

As a force based controller had to be implemented, the force/torque values will have to be read in to a control buffer. The control buffer will have to be accessible to the entire control program.

It was decided that the new implementation would have to be an *active* response system. This type of system is inherently more unstable than the passive system discussed in Subsection 4.2.4.2. Thus a PI controller was designed to improve the force response. The force balance equation governing the final system is shown in the previous subsection can be rewritten in terms of force controller terms. For force balance, the input force values are the setpoint and the force measured at the output side is the feedback force. The difference between the setpoint and feedback forces is the error e (equation 4.2.15).

The first implementation of the controller assumed a fixed time difference $\delta t = constant$, thus a constant discrete value is selected based on the theoretical implementation. Although this is rarely the case as there are inherent variable time delay caused by network traffic.

$$|e| = F_{setpoint} - F_{feedback} \tag{4.2.15}$$

A simple PI-controller was designed and implemented for each force balance independently. The controller implemented is shown in Figure 4.2.6. The aim of the controller is to minimise the force error between the master-slave pair. The *Proportional* constant is responsible for most of the time response, the larger this constant the faster the system responds, and this also has the effect of causing a large overshoot. The *integral* constant is responsible for stabilising the system between the allowable error thresholds. Thus the augmented variable is displacement.

A PI controller was found to be sufficient. The proportional component of the controller is responsible for the “speed” of the response (displacement). As is the case with generic controller theory, the larger the proportional constant the more likely and with larger magnitude the overshoot of the response will be.

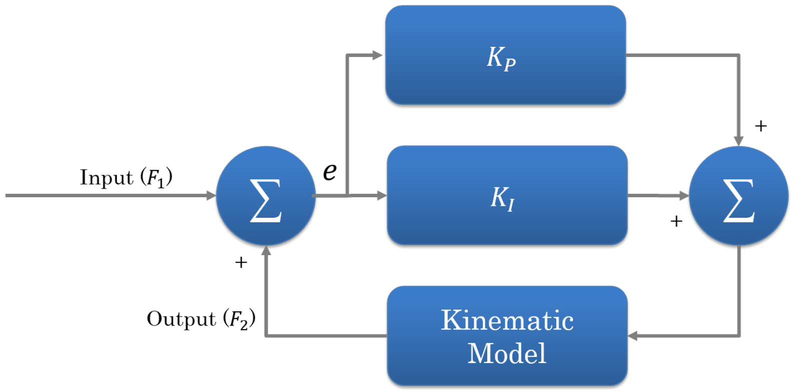


Figure 4.2.6: Basic PI controller implementation

Further in order to reduce the error one can implement an Integral controller, the integral component is responsible for reducing the error but may also make the system unstable. The output of the controller can be described by equation 4.2.16.

$$Output = K_P|e| + K_I \int |e|dt \tag{4.2.16}$$

Considering computer based integration and the information available to the controller, equation 4.2.16 takes the form of equation 4.2.17. This is the implementation solution that approximates an integral as the area under a curve. It can be showed that for this, each step size is the sampling time interval, denoted by δt. Thus the instantaneous output value of the integral component is equal to the previous sum of integral component output and the instantaneously calculated value, showed in equation 4.2.18.

$$Output = K_P|e| + K_{I-Previously-Calculated} + K_{I-current} \tag{4.2.17}$$

$$Output = K_P|e| + K_{I-Previously-Calculated} + K_I|e|\delta t \tag{4.2.18}$$

The new implementation was tested in a soft body impact configuration (Explained fully in Section 3.9), where the system was given a constant input and the output robot arm impacted a soft work volume, the arm was then brought to rest and the steady state response was observed. This process if repeated multiple times until a satisfactory response is achieved.

4.2.5 Software Development and Implementation

The software required for this project was developed using a hybrid development methodology. Overall an Extreme Programming (XP) approach was followed with the incorporation of UCSD elements. The project requirements were met by this combined model, as it was a low-cost approach which delivered high speed development and produced working software before the entire program as a whole was completed. The incorporation of UCSD elements specifically in the development of User interfaces and task scenario planning, proved to be advantageous in satisfying customer requirements. Further the approach was found to be ideal as the project had many independent sub-programs and was time constrained.

For the given development project, a top-down approach was followed, focussing primarily on user requirements of the software by a small two man development team. The core functionality of the software was derived from Pretorius (2012). An event driven system was established in the C-Sharp language, this was completed within the .NET framework.

Identical to the previous implementation, the F/T sensors broadcast force and torque data onto the local network using the UDP protocol, this in turn had to be converted to control signals using a newly developed kinematic model within software. The controller design was described in Section 4.2.4, including limitations and assumptions. While controller testing and final control variables for natural motion will be discussed in Section 5.1.1.

Communication to the robot controller is dictated by the robot controller. The robot controller can only use the TCP/IP protocol for live teach control, the communication specification is explained in full in Appendix D. Thus a TCP/IP control module with a complete instruction set as set out by the INFORM robot controller language had to be developed, applicable INFORM data structure specifications for communications are available in Appendix E. For the current implementation it was found that for improved stability and reduced communication lag, only the IMOV instructions should be used. The IMOV command is a move command that is executed relative to the previous position. Thus no previous positional data is required for movement command generation. This allowed for a simplified controller and kinematic model to be implemented.

Following the simple model set out in Section 3.3.2, Figure 3.3.2. The general architecture and layout of the software system is first planned, although this may change as the process continues. One of the developmental goals is to produce modular software. The modularity applies to the Kinematic model, Software based control system, F/T sensor communication among others. Further the video streaming and capture must be independent along with the robot communication and instruction set. This allows for change in the video streaming hardware and capture hardware without affecting the rest of the software, also the use of varying robots may be implemented if the instruction set is variable. The basic structure of the developed software is shown in Figure 4.2.7.

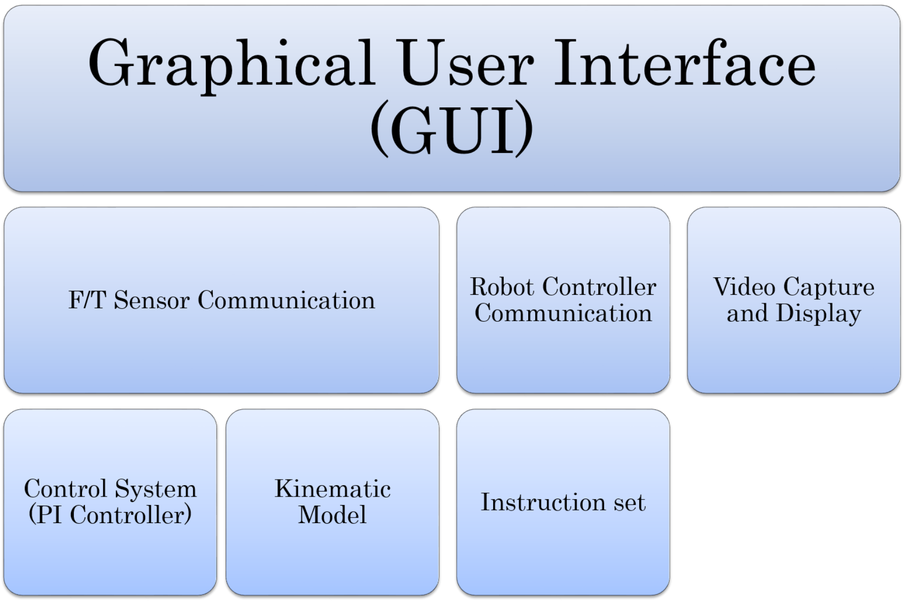


Figure 4.2.7: Software layers

The video feed from the IP camera was accessible using the streaming ports available on the local hosted streaming server that formed part of the camera. The software sent a stream request with authentication at program execution, enabling access to the stream from the local network. Further the feed from the on-board Black magic capture device, was available by accessing the fshow filter on the host computer.

Next the user interface will have to be considered in order to produce a usable interface that complies with the general heuristic principles for usable design of an interface, as briefly discussed in Section 3.3.3.

The simple UCSD model shown in Figure 4.2.8, was used to develop the interface in tandem with the XP development process. For software the UCSD model is based on work practise modelling, focussing on tasks and user classes for given tasks. The user-stories were used as functional developmental goals and to form required task descriptions that could be used for functional definitions which in turn could be used for functional specification. Further the user-stories were used in context of system testing scenarios (system goal driven scenarios).

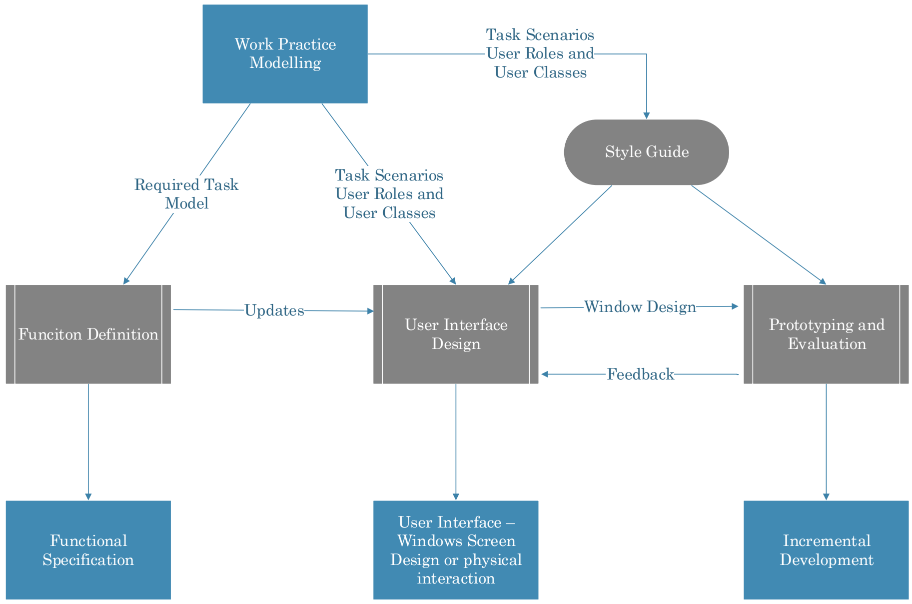


Figure 4.2.8: Simplified UCSD model for use in software development

For the current project no style guide was implemented in order to free developmental creativity when implementing code. Although user classes were used to generate two similar interfaces, the first is the developer view, this was implemented in a completely separate program. This version will allow developers to access direct control variables, such as software controller parameters and base speed settings as well as limiting directional execution to a selected axis only.

A limited version, without developer settings will be released to end-users. The interface was designed using the user stories, and technical functional requirements shown in Appendix H. After each functional component is implemented the internal-client tested and evaluated the implementation, if changes are required the new functional description replaced the old and another developmental iteration was executed.

The final dashboard was named SenCon, a mixture between SenRob robotics lab and Control Console. The final dashboard is fully discussed in Appendix I, along with functional areas and functional definitions, the dashboard can be seen in Figure 4.2.9.

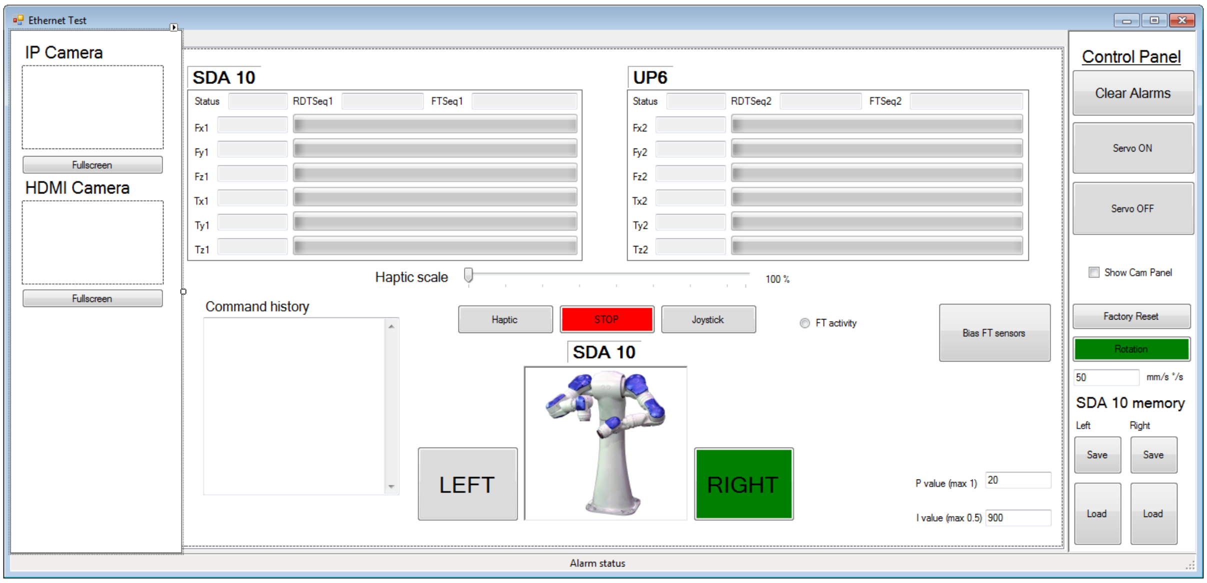


Figure 4.2.9: Console for Developers, Main Screen

The Dashboard consists of 4 tabs, the first is the main screen where major setup and feedback occurs, and this is the primary screen presented to the user during operation. The main screen consists of three panes. On the far left is the video feedback pane, which allows for control of video display. The middle section consists of 4 selectable tabs with varying functions and final pane on the far right is responsible for major control functionality.

The 4 tabs each pertain to a specific functional set or task completion regime. The first main screen, allows for selection and setting of primary functions. Some primary function include master arm selection, input mode selection and while in the developer version, setting of major control parameters. Further live status reports on the state of the force/-torque sensors are given to the user, this proves vital in understanding robot response and behaviour.

The second tab, holds the command window, shown in Figure 4.2.10. Here pre-programmed files can be selected and executed.

The third tab holds all the current detailed information on the robot. Information such as the status of the robot joints and movement parameters can be queried, changed and sent back to the controller. This allows for fine and specific control of the robot, along with higher level reporting functionality such as command streams, which can be useful during troubleshooting or traceability of actions, shown in Figure 4.2.11.

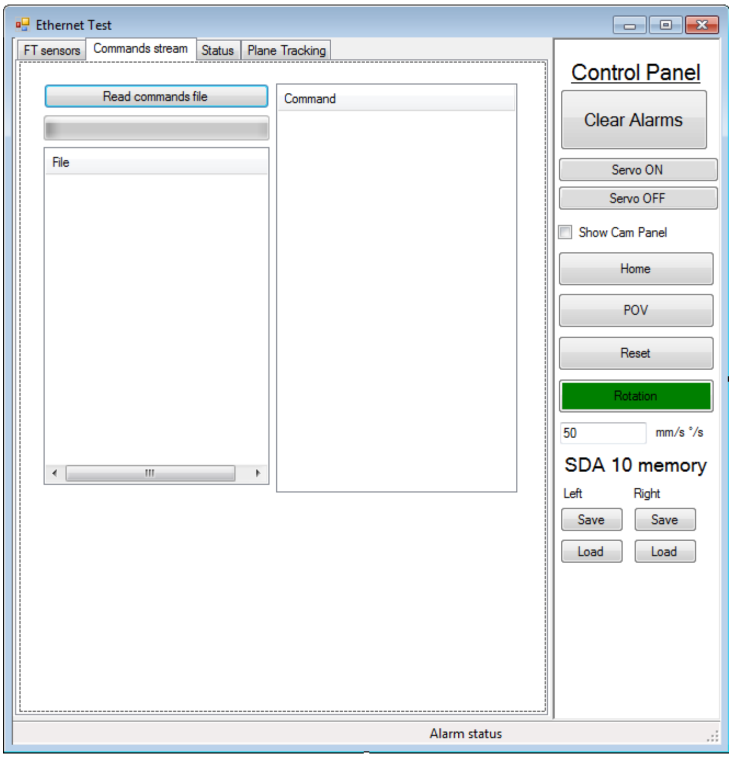


Figure 4.2.10: Command tab, file loading and execution

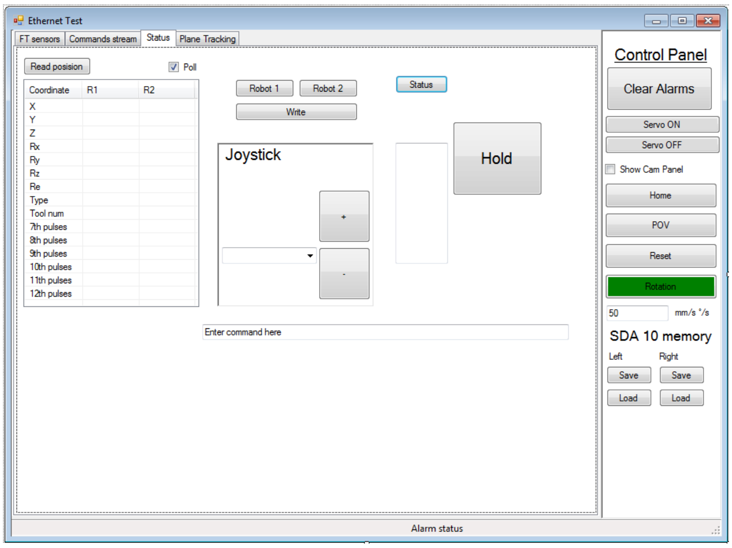


Figure 4.2.11: Status tab

The need for the next tab is a direct result of testing needs. Scenarios exist in which the robot motion will have to be restricted to a plane, for general use a click and drag mouse pane was implemented. This should only be used in non-sensitive testing environments. While the track pad is engaged the mouse event listener logs and interprets the signal and produces a pseudo displacement command. Further the user can use the master robot as a planar input in both haptic and joystick configurations. The tab is shown in Figure 4.2.12.

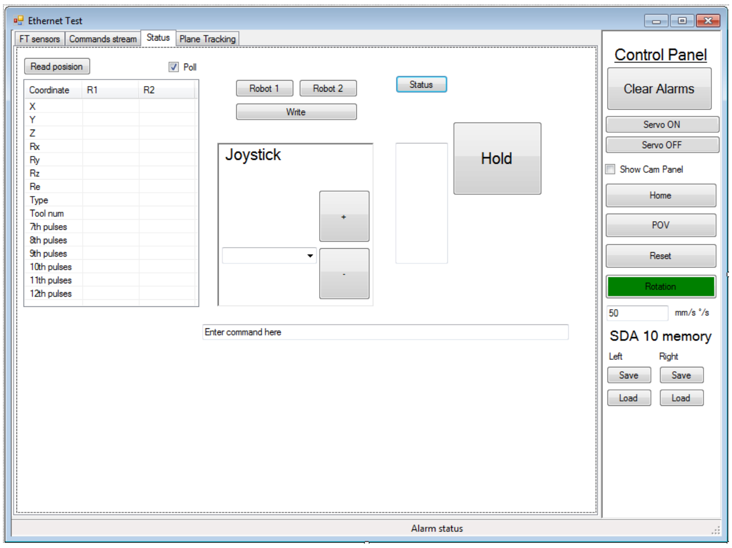


Figure 4.2.12: Status tab

The software proved to be successful, system developers could use the system to perform task specific performance testing with a test group. Further the system was able to assist in the completion of the micro machining project Read *et al.* (2013). The system also satisfies the heuristic principles for usability set in Section 3.3.3 by Nielson. After user testing the interface was found to be sufficient and satisfies all developmental goals.

The final step in the XP process is to refactor the software. The class diagram for the final software version is shown in Appendix H.

4.3 System Overview

All the sub-systems were brought together, combining all the electro-mechanical components, end effectors, input sensors and software components together. The input interface is comprised of two input components, the first the mechanical input can be seen in Figure 4.3.1.

The mechanical input is comprised of one arm of the SDA10 robot (master input), which one of the F/T sensors mounted to it, which in turn has an input rod (pencil substitute) mounted to the front of it. The rod design and use case was envisioned to be used as if writing with a pencil. The kinematic model which may provide further clarity on its design was discussed in Section 4.2.2.

Figure 4.3.1: Physical input design, haptic system

Further the user can interface with the software components by means of the software interface as discussed in Section 4.2.5. The software interface was displayed on a 24 inch touch display which was located in front of the user, at a position lower than the main 3D display which was used for visual and audio feedback to the user, an example of the use case is shown in Figure 4.3.2 and Figure 4.3.3

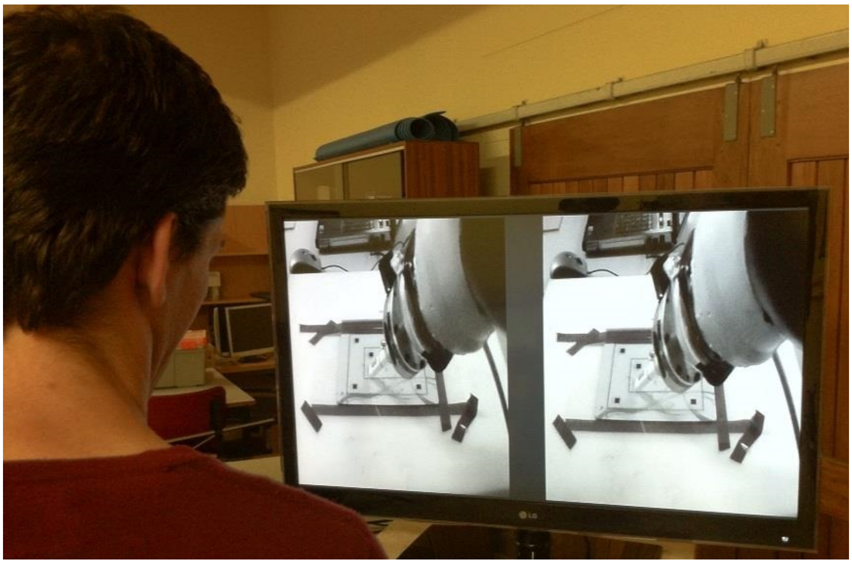


Figure 4.3.2: Testing of the visual feedback system

Figure 4.3.3: User interaction with physical and software input devices

For the experiments two specialised end-effectors where designed, both can be found in Appendix J. The first is an adjustable pen holder and this is used in maze completion (2D planar) experimentation. The other was designed for the ring placement (3D obstacle avoidance) experiment. As in most teleoperation scenarios the end-effector will have to be customized for each application.

Chapter 5

Results and Discussion

The results of the case study confirm what could be expected from literature. It is natural to assume that motion and kinaesthetic feedback will improve operator performance, the questions remain, by how much and how do the various factors influence each other.

Firstly, the system needs to be validated. Performance measures may be identified in two distinct categories. The first, is system specific performance measures. These are mainly whether or not the system satisfies technical specification. The specifications are originally described in Pretorius (2012) and reconsidered early in Chapter 4.

The second, user specific (user-centric) performance measures, is an indication of the usability of the system. Ease of use may be indicated using decreases in test times, less collisions, and collision mode - less severe. All these measures will be considered in conjunction with user feedback.

As was described in the Chapter 3, the system was not only evaluated using technical performance measures. The system was also evaluated using a hybrid heuristic approach, which will show the importance of some design aspects for the given user/test subject group and system. The analysis may be used to guide future development and show the importance of some design decisions.

5.1 Technical Verification

5.1.1 PI Controller Verification and Performance

Teleoperation systems call for advanced controller designs and control schemes (Passenberg *et al.*, 2010; Siciliano and Khatib, 2008*b*) which are outside the scope of the current project. The initial implementation used a simple direct feedback loop for control and relied on the speed of communication channel for system performance. This implementation also relied heavily on the speed of the older XRC robotic controller in order to achieve positional and force control. In this lies a fundamental flaw that lies at the core of one of the most important inherent problems with teleoperation systems, that of system latency. As was shown in Chapter 4 system latency is caused mainly by the communication latency, this may be attributed to components of feedback such as video or audio feeds. The control achieved was initially unsatisfactory; the implementation focussed on what could be achieved using the hardware available. To compensate a basic controller was designed and implemented by J. Pretorius in the previous iteration. Although improved force following and position-tracking was observed the system was still far from performing in such a way that a user will be comfortable using it, especially for high accuracy tasks. This is partly due to the extreme latency observed during live haptic control.

In order to improve upon the initial system, the current system required a more advanced controller, as the required level of wave particle controller design is outside the scope of the work presented in this project. The controller was omitted for the initial testing. Further as the internal robotic controller design was not available the decision was made to test initially without a software controller to establish a base line performance for the system.

Initial testing of the current system showed an extreme sensitivity to sudden sharp inputs from the input system. Sudden sharp inputs resulted in uncontrollable motion and feedback for the output-side system, which indicated that a controller will have to be designed and implemented. To make the controller more adjustable and to implement the controller as a modular component within the system, it was implemented in software. In general, control is implemented using velocity as the control variable for teleoperation systems. The velocity of the system cannot be polled from the robotic controller to be used as the control variable, thus force was chosen as the control variable.

As force-matching is important for applications such as micro machining and low contact-force pick and place tasks. The system was designed using force as the control variables. The system used a virtual kinematic model to resolve the input forces to the desired output position and force. The positional instruction set is generated using the force experienced by the input. As the positional instruction is the only commands that can be communicated to the robotic controller the virtual model is of utmost importance. The internal kinematic model used for the force to position conversion was derived from first principles and was shown in Chapter 4. In order to map the model some assumptions were made and have to be considered during future development.

In order to characterise the current system extensive testing was conducted and it was found that the system cannot be modelled as a first order system and that an advanced kinematic model will be required to generate a comprehensive instruction set. As a result it will increase both system control complexity and robotic controller load, thus reducing system speed and contributing to latency. For this reason the controller was designed using the IMOV instruction set, referencing the previous position and completing relative motion (IMOV is explained in Appendix E). This was done even though a comprehensive instruction and functional set of commands have been made available in software. This eased the use of force balance as the control variable. The PI controller was implemented at a level between the communication module of the interface and the kinematic model used to facilitate transducing from force to positional information as indicated by Figure 5.1.1.

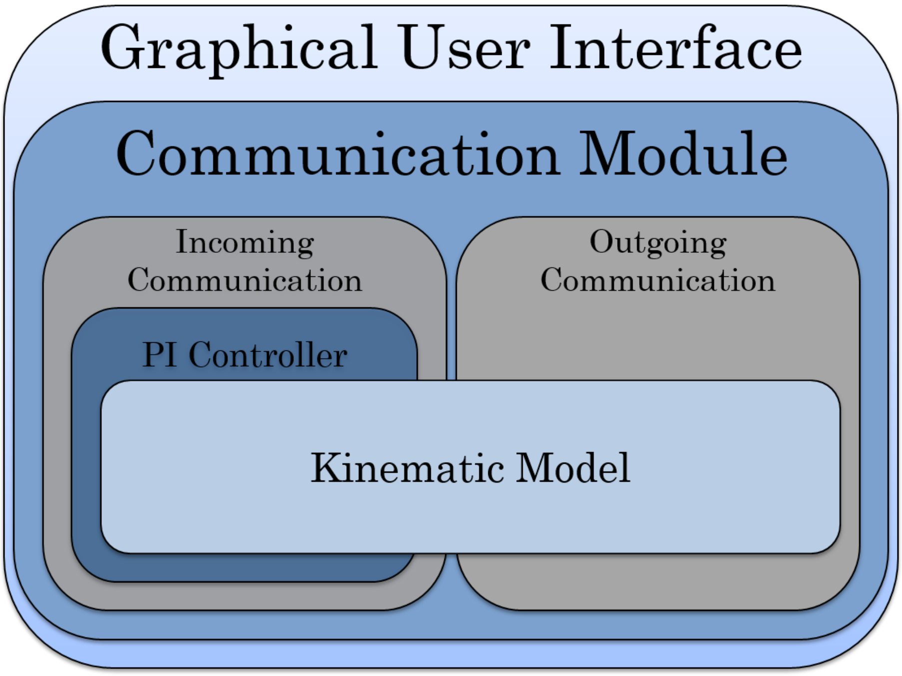


Figure 5.1.1: PI Controller position in software

General online tuning of the controller is required to get the desired output. The procedure was as follows; first the K_p (Constant) parameter was set to zero (0), then the K_i (integral parameter) value was varied to improve the rise time of the system.

The Developer must always be mindful of "integral windup" and instability of the system. The given system was first tested using constant input forces of 2N, 3N, 5N and 20N respectively. Both phenomena were observed during testing. To negate integral wind-up filtering was implemented as will be explained later in this chapter.

5.1.1.1 Integral (Ki) Tuning

From initial testing the system response was found to be unstable, with this a systemic error was observed which aggravated the instability. To stop the system form causing damage to itself the system was only tuned and the first 3 - 4 seconds was observed and considered. This was completed for large input forces only as these are known to cause an unstable response from the system. For the given instance the 19.65N force (20N) was used. As described in chapter 3, the system pre-loaded with the 2kg weight and allowed to settle. Then the system is activated in the joystick input modality, which did not allow the input-side to move in 3D space. The output-side was allowed free movement and was initiated from rest against a soft-body object.

As the PI controller uses force as the control variable one needs to consider the force error and its behaviour for varying K_i. The normal procedure for tuning the integral parameter of the controller is varying from a given initial value; from experience the initial value should be small. Due to the internal setup of the interpreter and the kinematic model, the initial value was chosen as $K_i = 100$. The behavioural change of the controller can be seen in Figure 5.1.2.

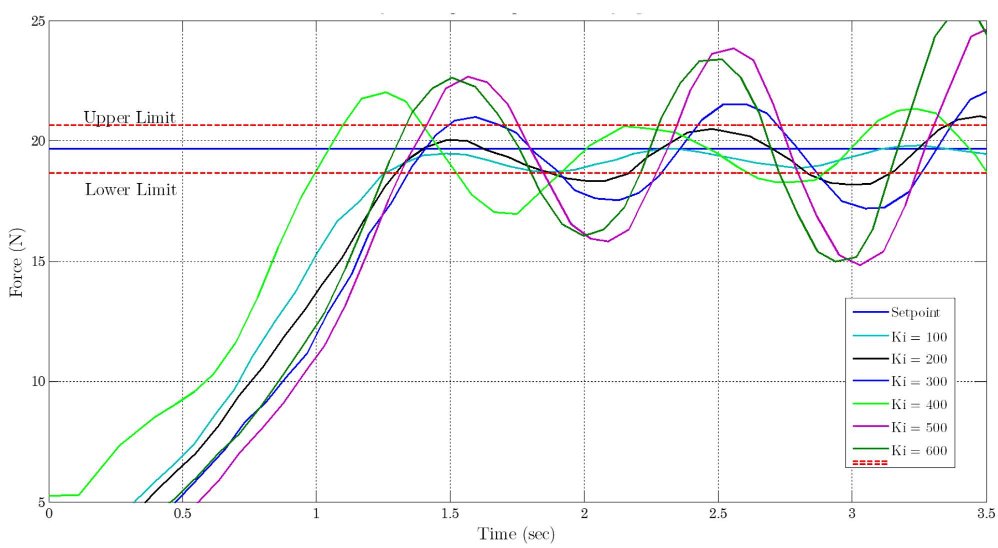


Figure 5.1.2: System force response for varying K_i and $K_p = 0$

From this it is easy to see that with an increase in K_i the system follows the expected increase in over shoot along with a decrease in rise time. The decrease in rise time observed is very small for the given values; this was attributed to the system speed limitation that was imposed for system safety. As the procedure is online, there was a constant risk of damage to the system if the tuning resulted in unstable behaviour. The system behaviour observed is consistent with unstable response.

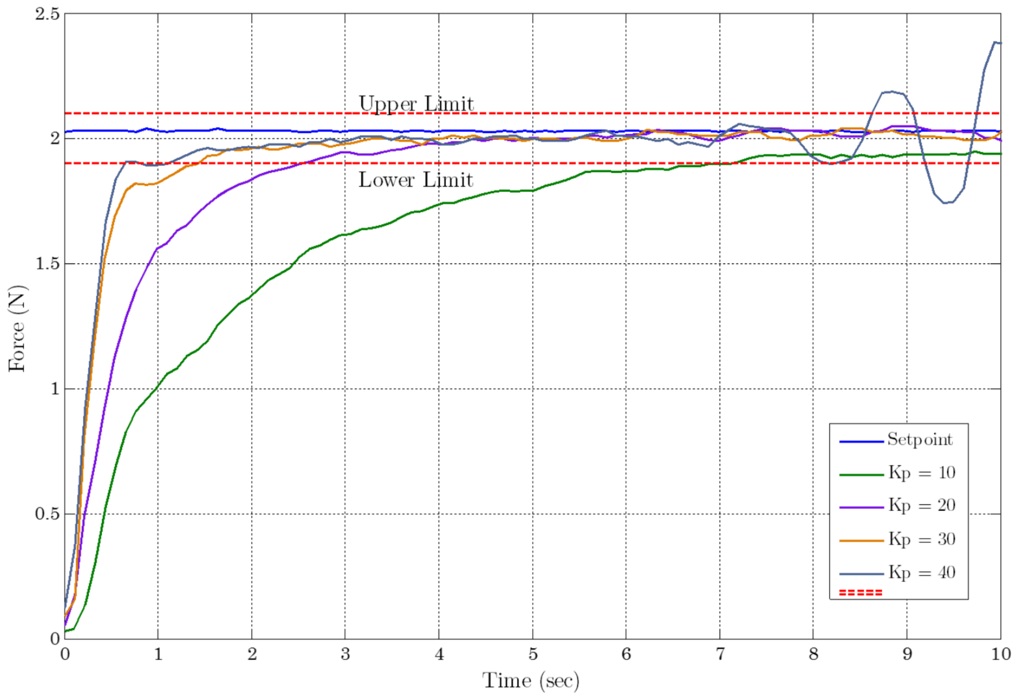
From this one can see that system stability degrades sharply between $K_i = 300$ and $K_i = 400$ and system overshoot becomes excessive. For $K_i = 100$ the system shows a tendency to undershoot and oscillate, while at the $K_i = 300$ level the system oscillates outside the 5% boundary design thresholds. After further tuning $K_i = 200$ was chosen as the value for K_i. This value showed a good balance of oscillation amplitude and rise time increase.

5.1.1.2 Proportional (Kp) Tuning

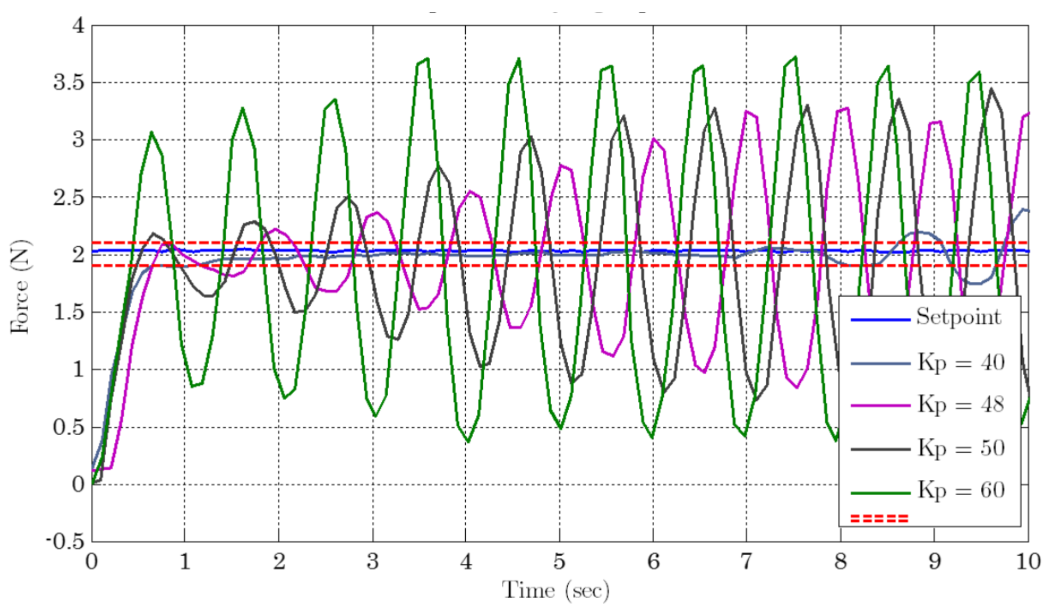
For the proportional parameter tuning the system K_i value was fixed to a value of 200 and the K_p parameter was varied. Due to the observed system instability, as shown in Figure 5.1.3b, it was decided that the system must implement some level of input filtering. In order to achieve this a filter was implemented in the kinematic model that the highest input force that can be observed by the system is 20N. Further the input sensitivity was limited so that noise on the sensors will have less effect on the system and to eliminate involuntary movement due to noise the minimum input force required for the system to take action was set at 1N. The latter value can be manipulated based on the system application. In most use cases the system will be used at a force level above 3N, the average input force for linear motion was in the region of 10N - 12N. By implication the lower limit has no effect on the normal operation of the system.

For system safety during tuning the system was tested at lower input level, as there is a correlation between system speed and the input force the system was tested at low input forces only. First we consider the most stable situation. As we see the rise time has decreased but the system will become more and more unstable as K_p increases. The stable region also showed a decrease for increased K_p, as can be seen in Figure 5.1.3a. This figure shows the relatively stable responses found by varying K_p. For low K_p the system showed a rise time of less than 1 second with a stable period of 7 seconds from contact, this is for an absolute constant input value. As a result the system design was changed to run on an event driven principal. If there is no change in the desired force for longer than 1.5 seconds the system will do nothing once that value is reached and will only actuate when the input or output side (feedback) is changed sufficiently.

From this it is clear that for increased K_p the system will become very unstable even at low input forces. To test this hypothesis and to assess system response at higher forces the system same tuning was done for an input force of 5N. The stable results are shown in Figure 5.1.4. As can be seen the force sensors showed a stable input force of 4.8N, and the system showed an decrease in rise time for the last psuedo stable value of K_p of 30. This shows the characteristic that for increase in input force the system's time for which it remains stable would increase.



(a) 2N response for $K_i = 200$ and varying K_p - pseudo-stable selection



(b) 2N response for $K_i = 200$ and varying K_p - unstable selection

Figure 5.1.3: System response for fixed Ki and varying Kp

This could be a compound characteristic of the kinematic model and filters. The fact that the larger the displacement command given to the robotic controller could in effect produce a larger velocity and as a result force feedback cannot be overlooked. Although detailed mapping is very complex, the control achieved was found to be sufficient in order to continue with the remainder of the testing regime.

It was also found that when user control of speed was instated, the lower the speed limit, the more stable the response observed. This showed that when very small movements were required the system speed could be adjusted to the required level and very accurate motion and force control could be achieved.

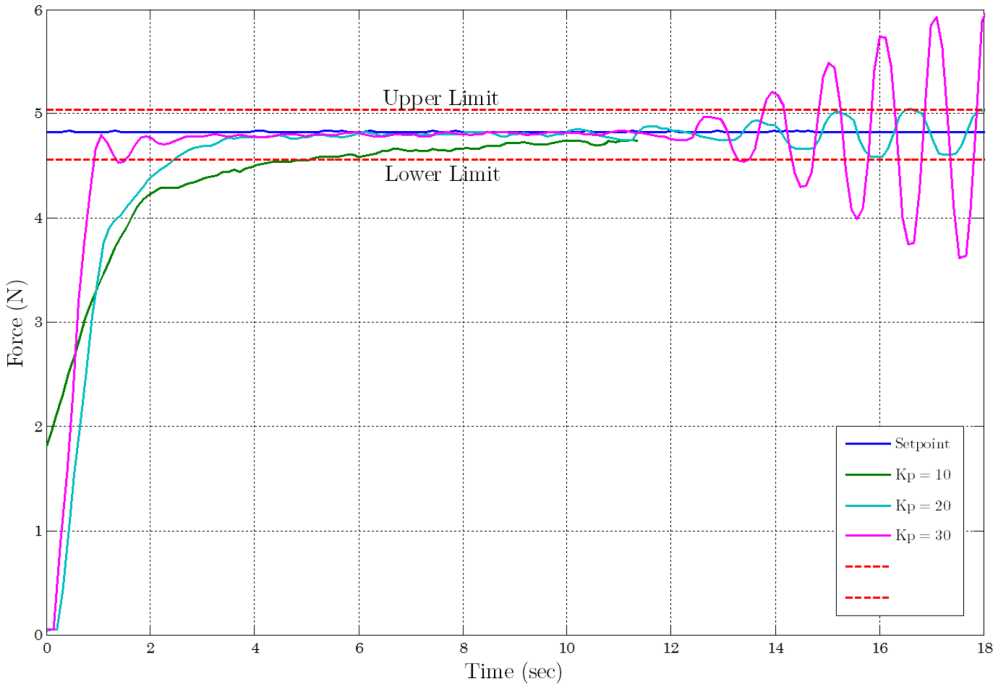
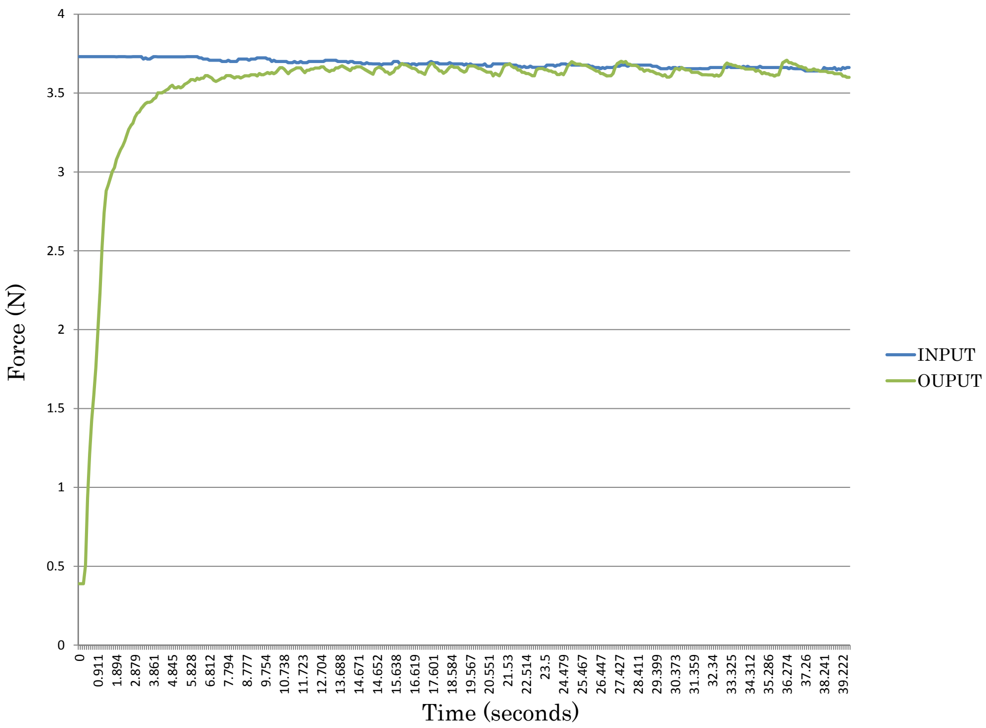


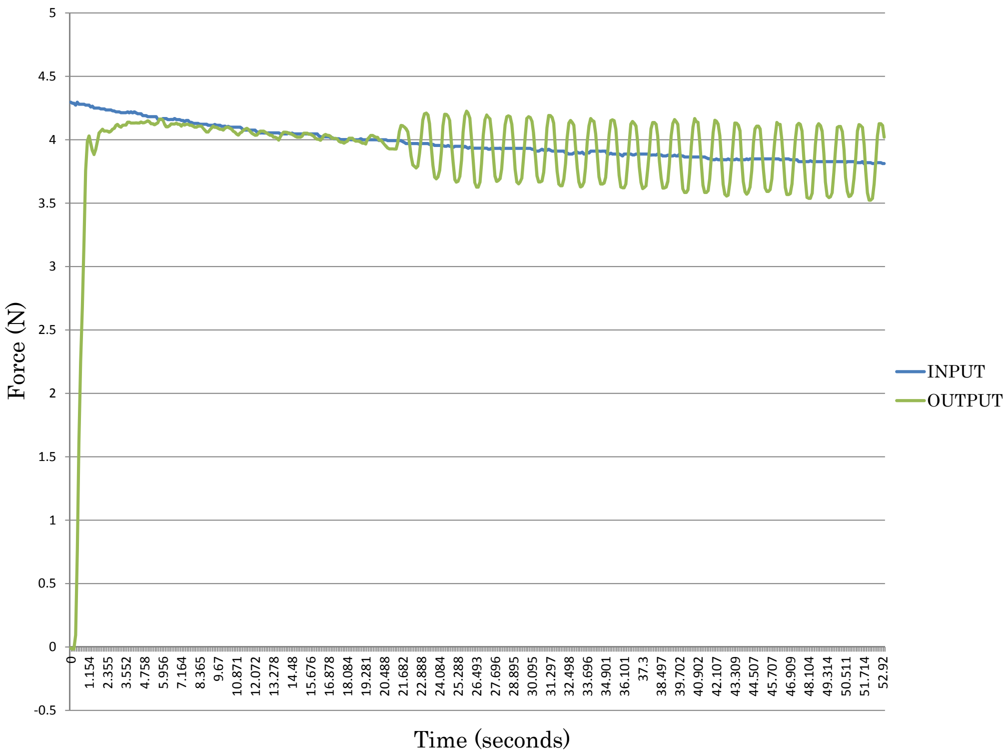
Figure 5.1.4: 4.8N response for $K_i = 200$ and varying K_p - unstable selection

The response change observed is shown in Figure 5.1.5a and Figure 5.1.5b clearly shows how the response is affected by speed restriction. The system showed that at 50% of the desired velocity the system was more stable at steady state and that there is an oscillation at steady state. This jagged response can be attributed to the response filtering, threshold control implemented, it was decided that this is sufficient for the given system. Although it is highly recommended that a more advanced control system be implemented that is more robust.

The system was tested for 2N - 15N and the same behaviour was observed. The speed was limited to a maximum of 30mm/s, although most operations were completed between 10 and 20 mm/s. The same response characteristic was observed for the 15N response characteristics. For this reason the speed was kept constant at 30mm/s for the user tests.



(a) 4.8N active filtering response 50% speed



(b) 4.8N active filtering response 100% speed

Figure 5.1.5: Comparative active filter response

5.1.2 Force Following

5.1.2.1 Force Following Time Response

The performance specification for the response was set at 500ms, as this is the level at which a user will become aware that there is latency in the system.

The system was tested at the lower spectrum of the expected input force, 2N. For this the system was preloaded to the desired 2N level and the system activated in the soft body impact modality. The observed response was recorded at 100% maximum speed of 30mm/s, shown in Figure 5.1.6. The system took 0.545 seconds to reach the 10% threshold and 0.647 seconds to reach the desired 5% input limit as this is an over-damped scenario the response took some time to climb to the required level. This is a vast improvement over the original system developed in Pretorius (2012), which had a system response of nearly 6 seconds.

Here system performance was limited by the controller design. As safety is paramount when using potentially dangerous equipment, it was decided not to pursue further improvement. As a result there is an accuracy limitation especially at low input forces.

Test users, did not complain about any latency issues and had no perception that there was an error in the force. It is strongly recommended that the controller and kinematic model be refined to reduce rise time and to enable more accurate force following.

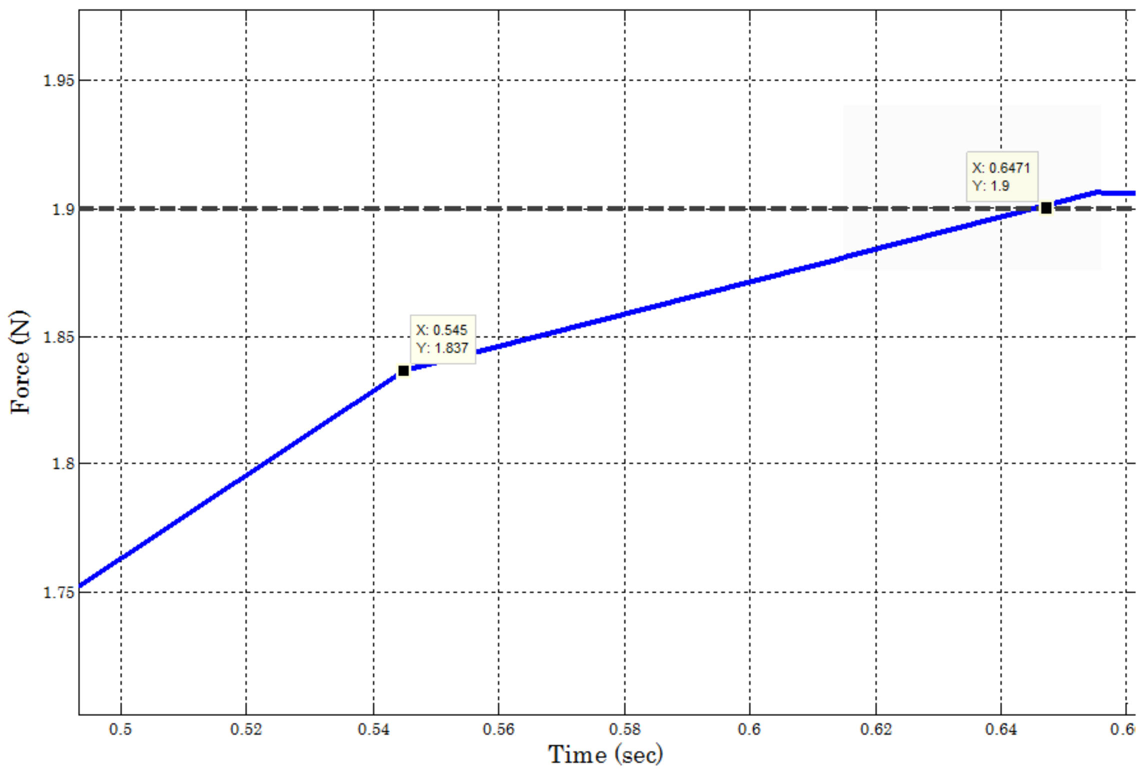


Figure 5.1.6: 2N Time Response

5.1.2.2 Force Angle Error

In order to assess the force angle, the force angle was decomposed into three force error vectors. This results in three force error angles, giving the orientational accuracy of the system. The angles were decomposed following their relation to F_x, F_y, F_z as α, β and ϵ respectively. The results are shown in Table 5.1.1, for a 90% confidence interval only 29, 55 and 68 samples needed to be taken for α, β, ϵ respectively.

The number of samples required was obtained using equation 5.1.1. Where T is the value of the t-distribution for the given confidence level, k is the confidence level, σ is the standard deviation of the sample and $\bar{x}$ is the mean of the sample.

$$n_{required} = T\sigma/k\bar{x} \tag{5.1.1}$$

Table 5.1.1: Force angle decomposition

	Average	σ	**Attainable accuracy at 90% confidence**
α	1.82	0.59	1.82 ±1.78 degrees
β	1.54	0.69	1.54 ±2.07 degrees
ϵ	1.67	0.83	1.67 ±2.48 degrees

As shown in the table one can see that the decomposed angle resulting from F_z is the most inaccurate with a maximum of 4.15 degrees off target. This is not ideal, especially at lower speed where greater accuracy will normally be required. As a result this requires more attention for future development, an example of where this will be very important is a pick and place task. The cause of the inaccuracy was initially thought to be the F/T sensors themselves. But this option was quickly discounted as the previously claimed results from J. Pretorius stated an angular accuracy of 0.07 degrees in steady state and angular error of over 15 degrees for dynamic loading.

Upon further investigation the 0.07 degrees overall accuracy could not be recreated using the initial system. The concept of force angle is also unclear from Pretorius (2012) and leaves the concept open for discussion. In conclusion, the error found is smaller for dynamic loading but is far greater at steady state. As the angular error plays an important role in force resolution, it is required to be as small as possible. The angular error, specification of 1 - 1.5 degrees was not attainable with the current implementation, although an improvement was shown in dynamic situation, with a maximum angular error being reduced by 3.5 times.

5.1.2.3 Force Magnitude Error

The steady state characteristics of the system will have to be verified. For this reason four cases for pre-load testing have been identified, each shows a modality in which the system might find itself on a regular basis. They are shown in Table 5.1.2:
Each of these conditions tested the dominance of the specific Force PI controller and the dynamic response to a force loading condition. Further the steady state response could

Table 5.1.2: Force loading conditions

Force	**Condition**
F_x	Force much larger than the other two components
F_y	Force much larger than the other two components
F_z	Force much larger than the other two components
F_x, F_y, F_z	Combination loading condition when all the forces are in the same order of magnitude, but have varying loads

be measured and stability of the system could be confirmed for a pre-load situation.

Force magnitude error is of utmost importance, the allowable error was based on a percentage of the input force. The allowable force error is 5% under and 5% over the desired force. The system showed some instability in its response. In an effort to negate instability filters were introduced on the input and output side of the kinematic model at the output side and on the input side before the PI controller.

Firstly one will have to consider static loading conditions. This is done in a effort to simulate settling when the system is required to keep a constant specific loading position. In an effort to do this, the system was preloaded and activated from a neutral position, i.e. no forces acting on the output sensor and one would expect a linear response.

Figure 5.1.7 shows how the complex force changes over time. From this graph one can also see some interesting characteristics, as the force torque sensors sample at a constant rate the force values are recorded at the same rate. For this reason the further the dots are apart, the larger the change in force and the further the motion, thus the closer the dots the smaller the displacement and the more accurate the motion.

As shown, this test only focuses on the magnitude error in 3 dimensions. When repeated there is a settling error that occurs. As shown in Table 5.1.3, the error maximum error for the example test run is 1.1N and the smallest is 0.34N.

Table 5.1.3: Force magnitude error for Test point 1

	Minimum (N)	**Maximum (N)**
F_x	0.8505	1.1805
F_y	0.34125	0.85825
F_z	0.455	0.711

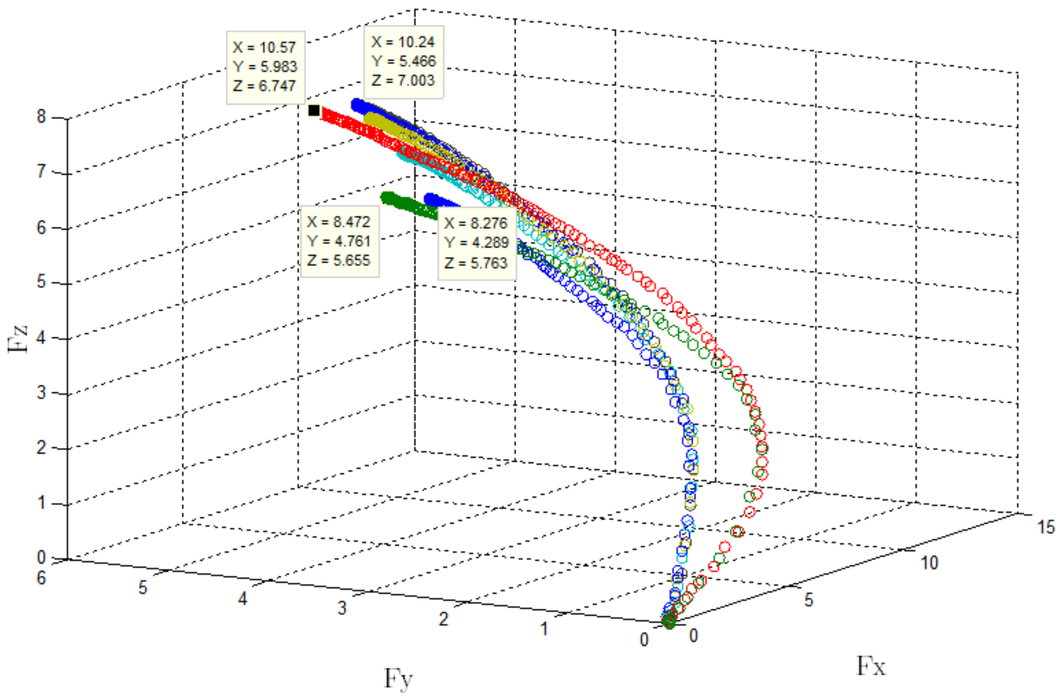


Figure 5.1.7: 3D Loading, Settling test

The experiment was completed for a total of 70 experimental runs, where all the loading conditions, for example F_x » F_y and F_z or $(F_z > F_y)$ » F_x. showed similar behaviour. Intuitively the error should tend to zero (0). After completion of the experiments, this was found to be the case. All components of the decomposed resultant force tend to zero. This is shown graphically in Figure 5.1.9.

Figure 5.1.7 also shows undesirable behaviour. As the force is used to generate positional data for movement commands, it is desired that there be no overshoot. Unfortunately as shown in Figure 5.1.8, there is a tendency to overshoot especially in the F_x - axis. Upon further testing it was found to be due to artefacting in the interaction between the robotic controller and the software controller, this is similar to the effects of both integral wind-up and rounding errors when large controller parameters are used. This originates if one of the forces is disproportionately large compared to the others, when the component of the movement vector is generated, it is much larger than the others and as a result causes an movement error that is only corrected when the forces are in the same order of magnitude. Analysis of the seventy (70) data sets showed that the steady state error for the system is 0.03 ±0.11N. This is a marginal improvement over the 0.123 ±0.034N previously attainable.

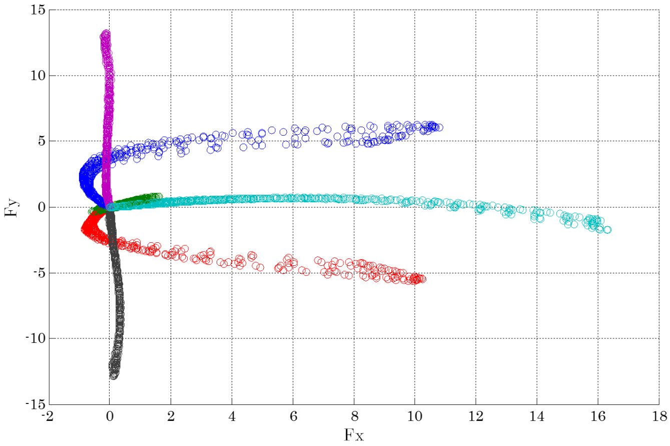


Figure 5.1.8: 3D Magnitude error behaviour Fx, Fy view

This was attributed to the fact that in this analysis in completed at steady state where the response becomes independent of time, especially as the mean values are used in the calculation. Under these conditions the major factors influencing the final error is the control system and the FT sensors. Put differently the sensor (sampler) and the input handler. The effect of the controller is minimised with the implemented filtering and the increase in accuracy of the control system for small movements. As a result of this the accuracy attainable is a direct result of the FT sensors themselves.

Further there was an observed drift in FT sensor output for a given fixed load, although very small this may become significant when working with small forces. This may be attributed to elastic deformation of the sensor.

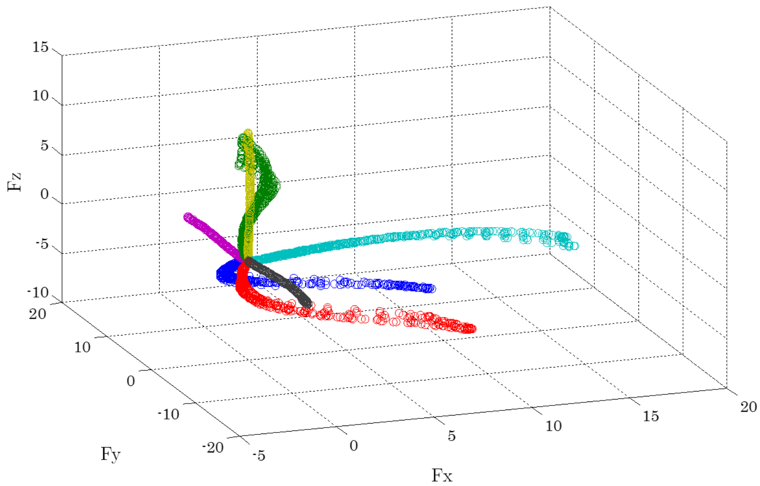


Figure 5.1.9: 3D Magnitude error behaviour

5.1.3 Summary of Technical Results

Although the system did not meet all the desired specifications, the system still has a high level of performance. The system also showed improvement, as can be expected in a second iteration project. A summary of the most important results are shown in Table 5.1.4, which show a current and first iteration over-view. Noticeably in system response rise time, the time required to mimic the master input. Further there was an improvement in the accuracy of the force following.

Table 5.1.4: Summary of Quantitative Results

Summary of Results					
	Symbol	Previous Implementation	New Implementation	Improvement
Response Rise Time	T	6 seconds	0.647 seconds	✓
Steady State Resultant Force Error	R	0.123 ±0.034N	0.03 ±0.11N	✓
Steady State Force Angles	α	0.07 degrees	1.82 ±1.78 degrees	-
	β	0.07 degrees	1.54 ±2.07 degrees	-
	ϵ	0.07 degrees	1.67 ±2.48 degrees	-

Although there was a degradation of force angle accuracy, the claimed value of 0.07 degrees was not specifically calculated in Pretorius (2012) and the procedure to do so was not outlined. As a result no assertion can be made as to if the result is truly poorer than in the first implementation.

The results show that the control system with filter has improved system control characteristics even for the relatively fast response time. For the given case study, the system was deemed sufficient and safe to move on to user testing.

5.2 User Centred - Case Study

5.2.1 Experimental Design Review

The user-centric design process called for a qualitative study to test important human factors for the system. The case study is part of the verification process for the design methodology that was set forth in the Chapter 3. For continuity a short overview of the experimental design is given here.

In short the experiment follows a 2^n full factorial design where each factor has two distinct levels which are shown in the design matrix below, Figure 5.2.1. The experimental design was designed such that each treatment was tested by 10 or 8 independent users. For this design the users were treated as a unit of variance.

For the first test (2D planar experimentation) the user group consisted of 10 people and for the second (3D obstacle experimentation) 8 participants were available.

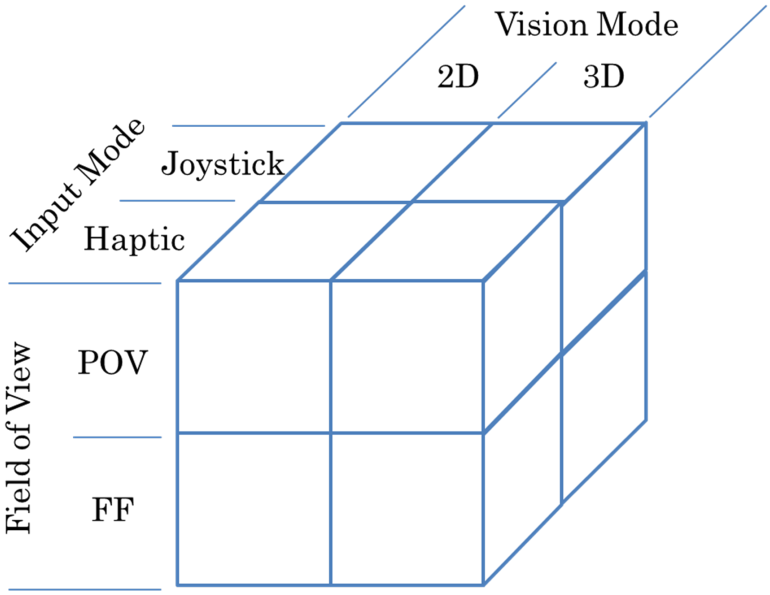


Figure 5.2.1: Experimental Design

The treatments were divided into several experimental runs. Following a balanced randomized design, the users were seen as a replication as they are a unit of variance. In order to meet the balance criteria, each of the combinations of factors would have to be completed at random by all 8 or 10 users in the sample group. The users were randomly assigned a treatment as is shown in Table 5.2.1. All users completed all treatments.

Table 5.2.1: Example experimental run sheet

n	video	FOV	Input Mode	User
22	3D	Full	Haptic	C
70	3D	Full	Haptic	I
39	2D	POV	Haptic	E
47	2D	POV	Haptic	F
73	2D	Full	Joystick	J
29	2D	Full	Haptic	D
45	2D	Full	Haptic	F
72	3D	POV	Haptic	I
28	3D	POV	Joystick	D
57	2D	Full	Joystick	H

When conducting studies involving users, many factors can be measured to gain an overall view of their performance. Here the aim is to assess the performance of the system. As the user is an unavoidable part of the system, one must consider the measures of performance as if the user is another component of the system itself. In light of this only the measures of performance that can be attributed to both the user and the system can be used to determine an overall view.

Further user preference data was collected after the completion of each experiment along by making use of a questionnaire along with the quantitative data indicated in Table 5.2.2 which quantified the system performance, with a focus on user enabled performance.

Table 5.2.2: Experimental Data collected for experiments

Performance measure	**Unit**	**2D - Planar Experiment**	**3D - Obstacle Experiment**
Number of collisions	Integer value	✓	✓
Collision Mode	Poke Graze Jab	✓	✓
Severity of Collision	Ranked value 0 - 5	✓	✓
Number of system errors	Integer value	✓	✓
Errors recovered	Integer value	✓	✓
Did not complete test	Integer value	✓	✓
User - perspective rating of task completion accuracy	Ranked value 0 - 5	×	✓
Pro - perspective rating of task completion accuracy	Ranked value 0 - 5	×	✓
Time to completion	Seconds and milliseconds	✓	✓

5.2.2 User-centric - Usability Testing

In the current project a user-centric performance analyses was conducted. For the purposes of the test users are seen as an unavoidable part of the system and were handled as a unit of variance within the context of the system. This was seen as the quantitative empirical study to assess the system as a whole and to see what effect the various (user-centric) factors have on overall system performance. All statistical analyses were completed with the open source software package R. The code used can be found in Appendix B.

Data was collected as described in Chapter 3. Users were asked to complete online questionnaires, which included both preference based and ranking questions. Refer to Appendix A for more details. The factors used in the factorial designed experiments are shown in Table 5.2.3, where the table includes all levels of each factor.

Table 5.2.3: Factor level description

Factor:	**Factor Levels:**	**Description:**
Input	Joystick	Tilt-only input.
	Haptic	Full (Haptic) kinaesthetic feedback.
Vision Mode	2D display	Conventional television display for a given field of view.
	3D display	3D enabled display (specialised television) for a given field of view.
Field of view	Point-of-view	Camera angle closely representing that of human vision.
	Full field	View of the entire field of view, a view from on high.

Although the test groups is small, consisting of only 10 and 8 people respectively, it will be sufficient as according to Nielsen (2000) only five test subjects are required and that the value added by extra participants will be minimal. A heuristic qualitative evaluation - usability study - is able to determine the major issues and will yield less results per person added. The information obtained from these insights may be applied in further development of user interfaces and system tools/control scheme.

Data was gathered for both sets of experiments for all users for each treatment. The user group was a representative sample of students. Demographic analysis showed the age spread from 21 - 29 was chosen as they have varying levels of experience with similar technologies and were readily available for the testing.

The sample groups aimed at having the same number of men and woman. The demographic analysis is shown in Table 5.2.4. For the second test (3D Obstacle experiment) only 8 participants were available.

Table 5.2.4: Demographic of the heuristic experiments

Experiment	**# men**	**# Woman**	**Total:**
2D Planar Test	6	4	10
3D Obstacle Test	4	4	9
	Age: 21-29	Age: 23 - 26	

The analysis of results was twofold. As this is a preliminary study, a small representative sample was chosen. The user data was analysed using two methods, as described in the Chapter 3. Firstly a qualitative analysis was completed, making use of the gathered data from the questionnaires. Secondly an empirical quantitative study was completed to assess the effects various experimental factors on system, and therefore user performance. The experiments were designed to emulate scenarios for general use cases.

For the quantitative analysis utilised time-to-completion (TTC) is seen as a heuristic performance measure as it takes into account multiple variances within the system for combinations of factors, showing the *overall* system performance. TTC is also commonly found in system performance measurement literature. As explained in Chapter 3, TTC is the system response variable for the different treatments, in some cases collision and accuracy data was also recorded. The data was analysed using various methods in order to form a comprehensive view of the data.

The first was an Analysis of Variance (ANOVA). There were many methods for completing a three factorial ANOVA, making use of R-statistics software the data can be listed in such a way that the general linear model (lm), as explained in Chapter 3, can be used to find the main effects and the various interactions quickly and easily. The distribution of the various treatments' response samples are *near*-normally distributed, thus an ANOVA can be used. In order to confirm which one of the factors was responsible for the interaction, a post-hoc analysis was needed. This was done in the form of a Tukey Honestly significant difference (HSD) test. Further an analysis was conducted that fitted a chi-squired distribution to the data; the model fit was proven to be sufficient as a result of the skewness of the distribution. This opened the door to allow non-parametric analysis methods as explained in the Chapter 3. The two non-parametric tests chosen were the Friedman -rank sum test and the Wilcoxon rank sum test, the need for the Wilcoxon test arouse from literature as a Friedman test is seldom applied to a factorial design with only two levels.

5.2.3 Usability - User Preference

System usability is a mixture between performance, perception and preference. These parameters are interdependent, for example: A user might be able to complete a task faster using input modality A, although the user perceives modality B to be easier to use. Further the same user might prefer viewing mode B and input mode A, but performs better using input mode A with viewing mode A than with B. It can be expected that users would prefer those modes or combination of modes in which they performed best, however this was not always the case for the given system.

Various system attributes were considered to be important. Among them are the three main factors namely ***input mode***, ***field of view*** and ***viewing mode***. Here the user perception and preference are used to show configurations that users preferred in order to guide future development efforts or stream line future testing decision making.

5.2.3.1 Input Mode Preferences

It has been shown that users in general perform better using haptic input devices and control. This however does not guarantee that user perception of success will be in line with actual performance. User preference is an important factor, and as the results in Figure 5.2.2 show, in only one of the two experiments haptic control was preferred. The other was found to be inconclusive as an equal number of users opted for the joystick control as well.

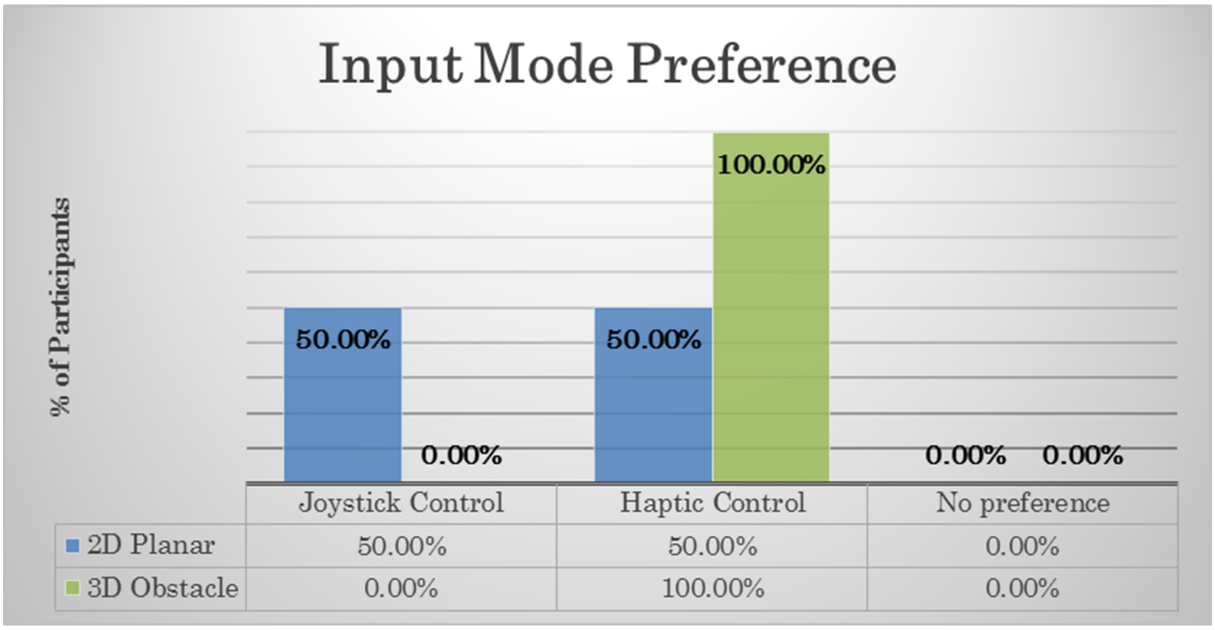


Figure 5.2.2: User Input Preference

As shown in Figure 5.2.2 there was no clear preference in the 2D planar experiment testing. This may be attributed to the nature of the experiment, where movement is restricted to a single plane, users could only perform movements in the x and y directions. This may have contributed to complacency when users must consider their preference. This may also be interpreted as being perceived equally easy or difficult. Either way, no clear deductions could be made from this alone and further investigation is required. It is important to note that all the users had a preference and none opted for the "no preference" option. This indicates that the results show their true preferences for input mode. A more comprehensive picture of general preference may be achieved if a larger user group is tested.

In the 3D obstacle avoidance experiment, however, users were required to move a ring from one position to another with a target as fast as possible with the highest accuracy they can achieve. Users showed a disproportionate affinity for the haptic input mode. This can be attributed to the control scheme, where the control mode allowed for kinaesthetic feedback. This allows users to construct a 3D model of the environment in their minds. It was also observed that nearly all the users first lightly impacted the work surface of the work volume in order to establish a cognitive baseline of the position of the work surface.

5.2.3.2 Viewing Angle Preference

The field of view is considered as a main factor in establishing situational awareness. Most systems rely on a combination of views to create an artificial sense of situational awareness. The viewing modes were explained in Section 5.2.2. For both experiments the field of view was fixed.

The field of view also plays a role in the accuracy to which users are able to complete specific tasks. This can mainly be attributed to the manipulator partially or fully obscuring the user's view. Some users reported that the positioning of the manipulator seemed to be unnatural and difficult to manoeuvre in the output environment, due to viewing angle and field-of-view.

For the 2D Planar experiments one would have expected the FF view to be preferred as the entire maze will be visible at all times. However this was not the case, as was shown in Figure 5.2.3. Most of the users expressed a preference for the point-of-view or lower field of view, reporting that it is more difficult to judge the movement of the manipulator when using the FF view when completing the maze. A further two users of the test group showed no preference to a specific viewing mode, claiming that both viewing modes were adequate to complete the task required and both were considered equally intuitive and easy.

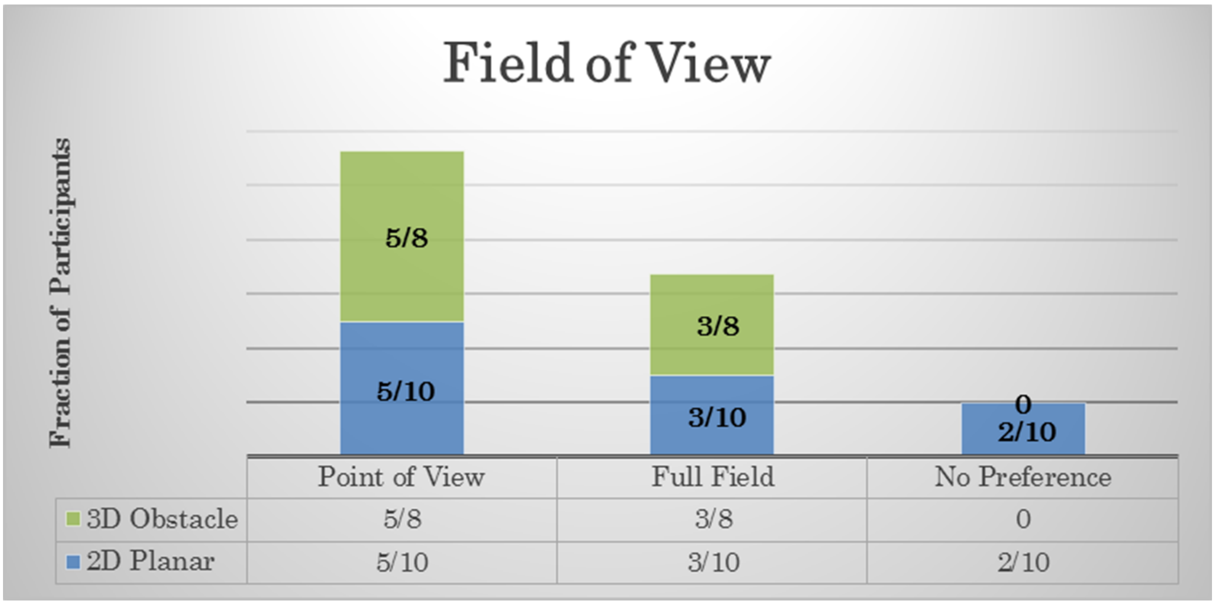


Figure 5.2.3: User Field of View Preference

The lower angle, hence field of view might be preferred as it is the more natural of the two views. This view may also have allowed for user perspective to play a role, as the full-field view is perpendicular to the work surface and it is far more difficult to judge depth, even under a 3D viewing mode condition.

5.2.3.3 Viewing Mode Preference

As was described in Section 5.2.2, the viewing mode represents the mode in which the video was presented to the user, either 3D or 2D video feeds. Intuitively one can surmise that for the 3D obstacle experiment 3D video would be advantageous in completing tasks as the depth perception gained would makes it possible to judge distance more easily. This was confirmed by the user study and as shown in Figure 5.2.4, all users opted for the 3D viewing mode.

Users reported that the 3D viewing mode required some time to adjust to. This is similar to passive 3D cinema, where some users may require time to adapt to the eye strain that may be experienced. During this time some users did experience eye fatigue, although the time required was minimal and after the adaptation time the users had no further complaints. This phenomenon is common with most available 3D viewing technologies.

On the other hand there was a wide spread of responses to the viewing mode preference for the 2D Planar experiment. Most users opted for the 2D viewing mode overall. This may be attributed to the loss of the need for depth perception. When considering the tracking data collected during the experiments the 2D viewing mode is slightly less accurate when compared with the 3D viewing mode's tracking data as the tracking paths followed are much more random and far away from the optimal path to follow.

Further the user that had "no preference" commented that both scenarios were equally difficult, due to a misconception of the directional positioning of the input work area as compared to the output.

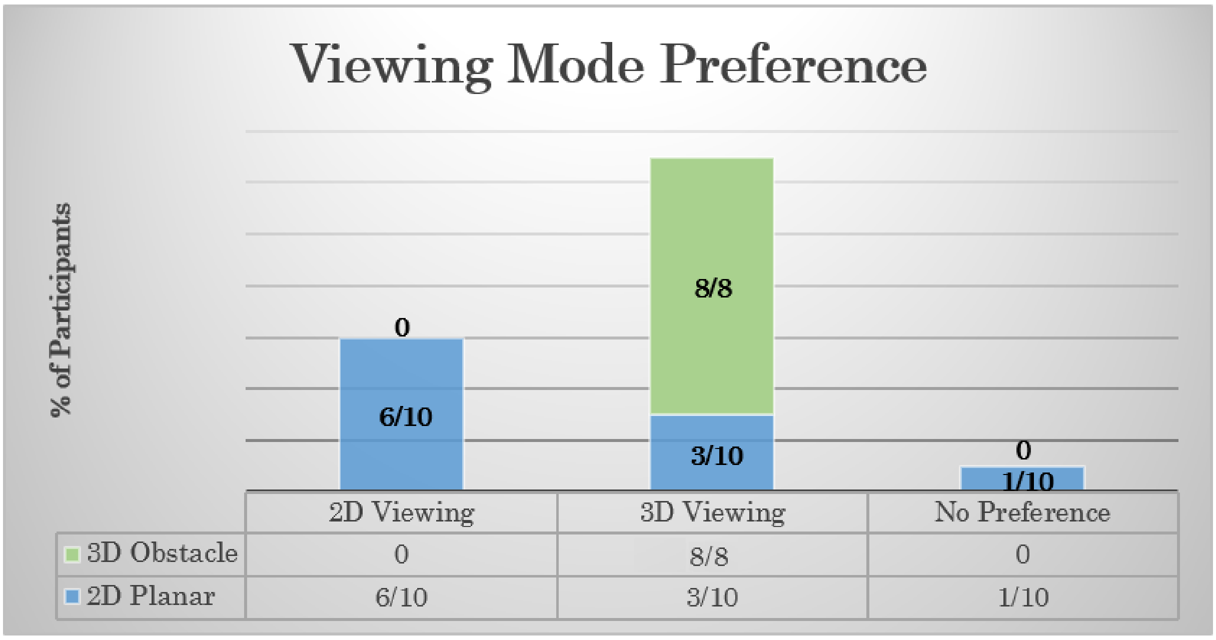


Figure 5.2.4: User Viewing Mode Preference

5.2.3.4 System Speed and Response

The system response speed was kept constant for the duration of the experiments. In general the system response characteristic dictates that if the system speed is increased the stability of the system response deteriorates. This behaviour was observed and was shown in Section 5.1.1.2.

Users were presented with four options, if they were allowed to change the system response speed for a general use case:

1. ***Increase*** - Increases the speed of the system response.
2. ***Decrease*** - Slows down system response, this in effect acts as a dampener for the system response.
3. ***Don't care*** - Indifferent.

4. ***Dynamic*** - Users control the system response speed, for example: for general large movements where system stability is of less concern, the response would be faster than when small manipulation is required, where -users would like to decrease response in order to increase accuracy and controllability.

As can be seen in Figure 5.2.5, the initial usability assertion, used to develop the user interface, which stated that users *should* be able to dynamically change system speed has been confirmed. In both cases more than 50% of the users would like to have dynamic control of the system. Further nearly 25% of the users are indifferent to the system speed. Only a small percentage of the users in the 3D obstacle experiment would like to increase the system response speed. This can be attributed to the experimental setup and system limitation imposed for safety reasons. The system speed was limited due to both manipulators moving and which poses considerable risk to the user if speed was unlimited and a system error occurs.

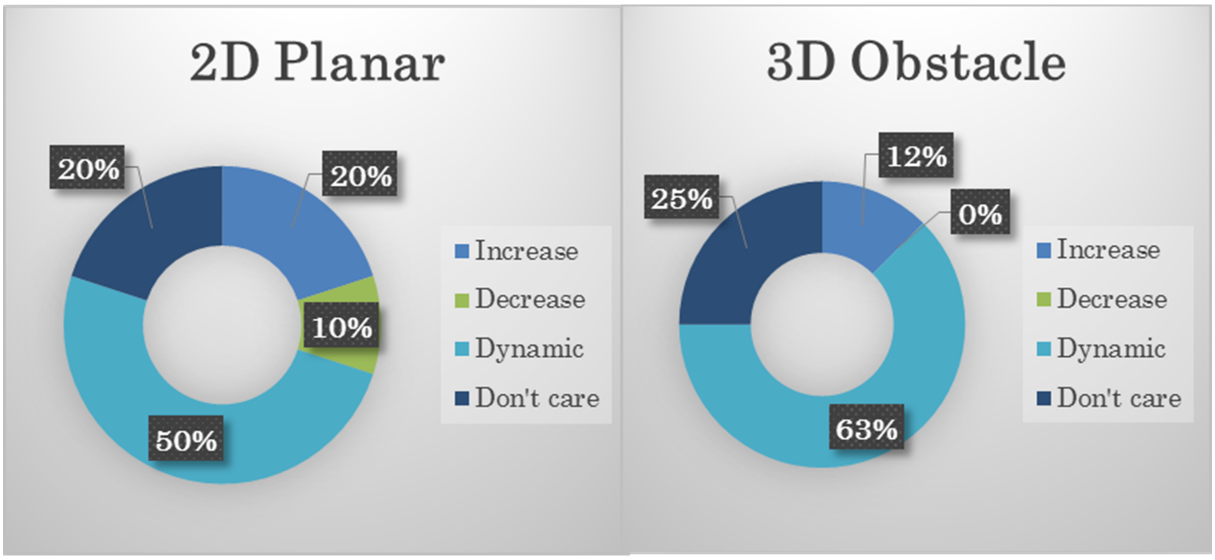


Figure 5.2.5: User Preference for System speed

5.2.4 User Preference Ranking

The combined user preference data is shown in Figure 5.2.6. This figure also shows the relative ranking by each user for a given input and experimental combination of factors. The data is shown in descending order from first choice to fourth. By implication the data on the left can be considered as the most important as the users were required to make a choice and there were no "no preference" options available.

Figure 5.2.6 and 5.2.7 shows the results for user ranking data in completing both the 2D planar experiments and the 3D obstacle experiments. The data was separated with respect to input mode and grouped according to rank of choice, for given combinations of field-of-view and vision mode. The overall user ranking preference results can be found in Table 5.2.5 and Table 5.2.6 for 2D Planar and 3D Obstacle experimentation respectively. The table can be used for decision making for developmental goals.

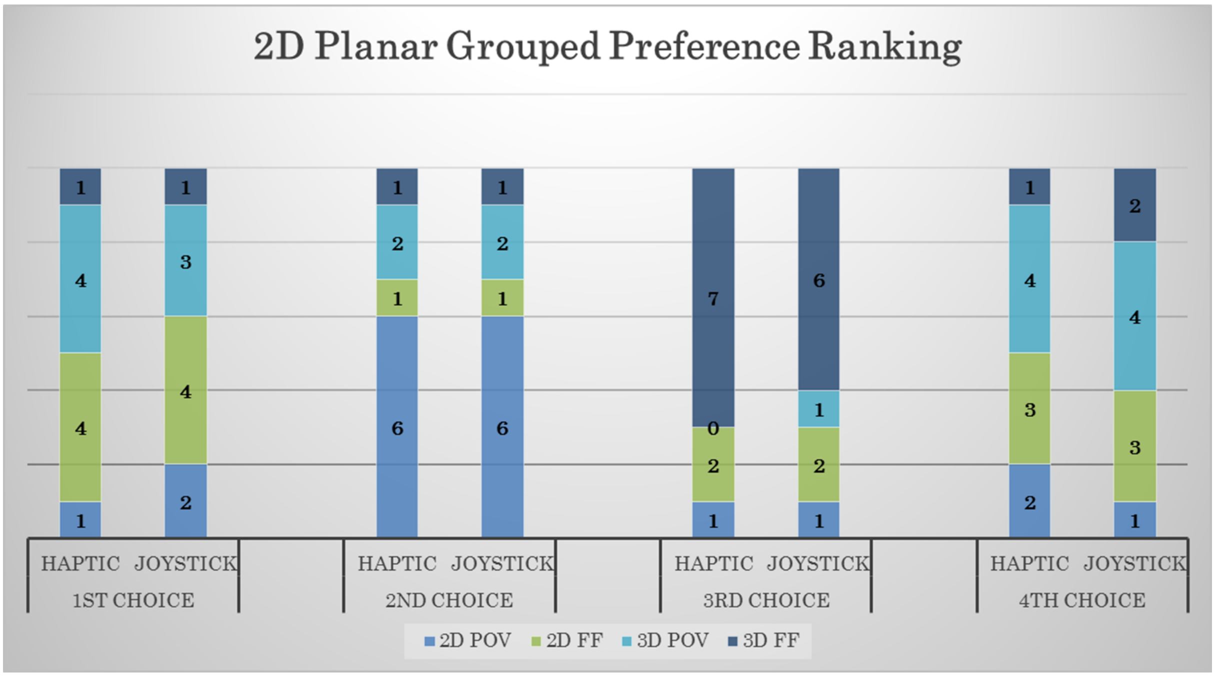


Figure 5.2.6: Grouped User Ranking for 2D Planar Experiment

Table 5.2.5: 2D Planar Preference Combination

User Choice	Video Mode	Field of View	Input Mode
1st	2D / 3D	FF / POV	Haptic
	2D / 3D	FF	Joystick
2nd	2D	POV	Haptic
	2D	POV	Joystick
3rd	3D	FF	Haptic
	3D	FF	Joystick
4th	2D	POV	Haptic
	2D	POV	Joystick

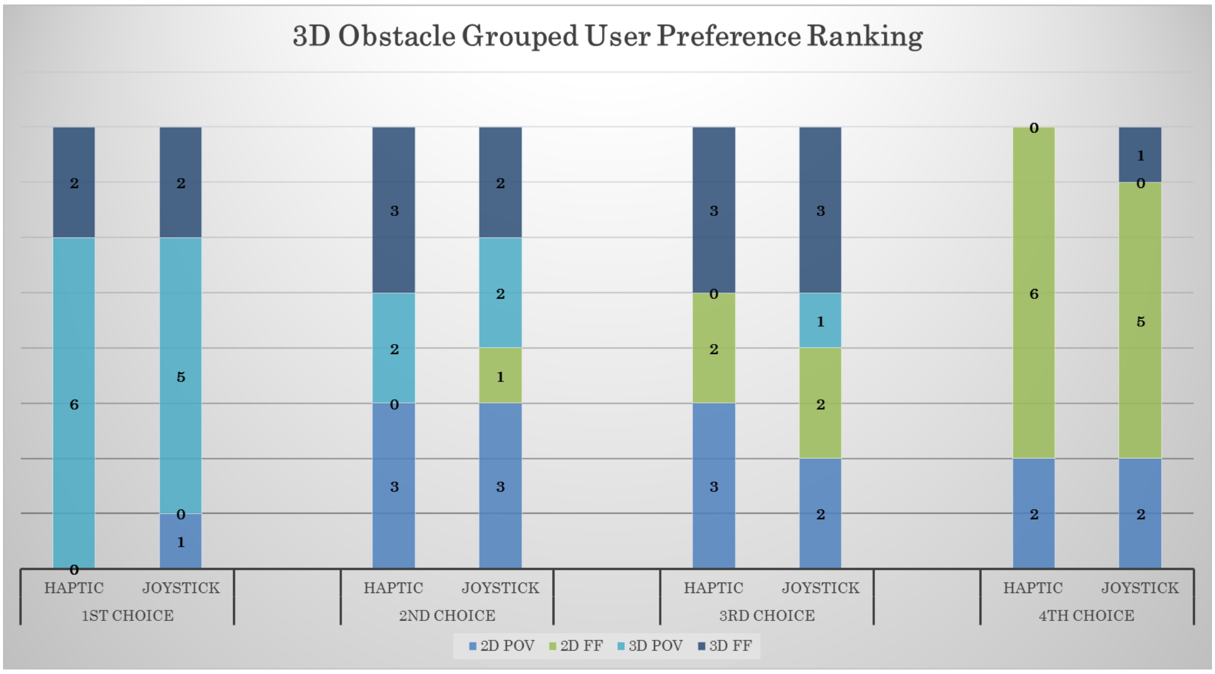


Figure 5.2.7: Grouped User Ranking for 3D Obstacle Experiment

Table 5.2.6: 3D Obstacle Preference Combination

User Choice	Video Mode	Field of View	Input Mode
1st	2D	POV	Haptic
	2D	POV	Joystick
2nd	2D / 3D	FF / POV	Haptic
	2D	POV	Joystick
3rd	2D / 3D	FF / POV	Haptic
	3D	FF / POV	Joystick
4th	2D	FF	Haptic
	2D	FF	Joystick

The preference rankings can be useful in determining the future development goals of the system. User preference data is also useful in deciding the combination of vision mode, input modality and field-of-view will work best for the current system. for development or should be considered first when designing use case guidelines.

From the user ranking data, the user preference multi-criteria decision matrix in Figure 5.2.8 was compiled. The matrix was constructed using weighted ranking data, similar to a Borda race, of which the highest weighting is four (4) and the lowest one (1). The weighted ranking data was then constructed into a matrix, which can be used for this system only. This matrix allows future developers for the current system to quickly decide the appropriate factor level for a given restricted set of fixed factors. It is also clearly shown that users prefer the Haptic, 3D view, POV field of view for 3D obstacle testing and the Joystick, 2D view, POV field of view for 2D planar testing. This should ideally be the experimenter's first choice, if possible.

In the general case, point of view was preferred over Full field as a field of view. Along with this it was also found that users prefer the 2D planar testing modality and that the Haptic input modality is preferred. Although for the 3D obstacle testing there was no clear winner in the input category.

		2D Planar Exp		3D Obstacle Exp	
		Haptic	Joystick	Haptic	Joystick
2D Viewing Mode	POV	26	29	17	19
	FF	25	24	6	9
3D Viewing Mode	POV	24	21	32	30
	FF	23	20	25	22

Figure 5.2.8: User Ranking Decision Matrix

The decision matrix shown in Figure 5.2.8 can be used in deciding which modality to complete a given test. If some of the factors are fixed due to testing constraints the matrix may be used to decide which modality of the other free factors to run the test in, based on user preference.

For example, if the experiment must be completed in a 3D environment, such as pick and place operations and the other factors are free. It is best to do the test under haptic control with a POV with a 3D video feed as it has the highest block ranking. Similarly if the experiment must be completed in a 2D planar manner and only a POV field of view is available with the other factors free. User preference shows that the best modality will be the 2D video feed with joystick input.

This is based solely on user preference. Later analyses in user performance may show some contrasting results and have been discussed in Section 5.2.7.

5.2.5 Performance Analysis

In the previous sections user preference and ranking data were analysed in order to guide future work and help clarify the usability of the system. Detailed results can be found in Appendix K.

Further it is required that the real-world performance data must be evaluated. Many performance measures were chosen in order to form a collective over view of the overall user and system performance. The system performance aspects such as system errors and recoveries are an indication of the system stability under varying user control. The other performance measures that were chosen were primarily focussed on user performance within the system under varying real-world scenarios. These included the number of collisions as well as the modality of the collision e.g. was the collision a bump or a scrape. This data was collected to highlight any inherent system flaws, such as recurring control modality specific errors. This data may have also highlighted any drawbacks related to viewing mode or field of view, which can be improved.

A main performance measure was also identified from literature namely time-to-completion (TTC). The TTC is combined measure of performance for the system as a whole. This includes the user within the system. It may be argued that the empirical study assesses user performance as a function of system performance although under the test condition an effort was made to keep all controllable variables constant. For an initial experimental run, 10 users were chosen at random from an available final year and Masters students. The performance tests were conducted as described in Chapter 3.

The statistical tests were broken up into three tests. The first was an ANOVA with replications, 10 for the 2D planar experiment and 8 for the 3d obstacle experiment, which corresponds with the number of users tested. The second test was a Friedman Rank Sum test and the final test was a Wilcoxon signed rank-sum test hypothesis test. The last two tests may be considered to be similar, although the Friedman test is rarely used for factors with only two levels.

As is the case for most hypothesis tests, a post-hoc analysis is required in order to identify the responsible treatment for the rejection of the null hypothesis. The Wilcoxon signed rank-sum test was also used to perform within-subjects design evaluations. The results from these tests helped with the evaluation by clarifying which factors and at what level contributed to the null hypothesis being rejected. This also showed the effect sizes of the different factors.

5.2.5.1 ANOVA

An ANOVA is useful in determining cause and effect, for this case the univariate case where TTC is the continuous response variable. This method of hypotheses testing may be based on many models. The model used for the purposes of this project is the linear model.

As the experimental design is a balanced 2^k, with completely randomized designed experimental runs. It is also useful in determining if there are interaction effects in a given data set, especially for a simultaneous 3 factor evaluation. Each factor had two levels as previously discussed in Section 3.7 and was tested using this hypothesis test. In the cases where the interactions are significant or where the distributions of the two sample sets differ significantly, a post-hoc analysis will have to be completed in order to specify which factor is responsible. For this there were numerous options of which the first was a paired-t test. The second was a Tukey HSD test along with non-parametric options.

Non-parametric tests can find the variation in distribution at a loss of power.
The data was analysed with a $\alpha = 0.05\%$, although due to the small sample size few significant results were found.

5.2.5.2 2D Planar Experiment ANOVA and Non-parametric Analysis

The sample user group showed interesting characteristics. Although only the Input comparison differed significantly, one may still use the data to show effect and cross effects. As shown in Table 5.2.7, there is also a less significant effect between Viewing Mode and Field of View. By removing an outlier from the data, namely user J, it is shown that this interaction becomes more significant and lies marginally outside the $\alpha = 5\%$ significance bounds.

The smaller sample size has proven to be problematic in that statistical inference can be unreliable for small sample sizes. The test was run for the 10 users and the aim of the test was to find the factors that have the most influence on user performance. The test may also have helped in finding if there were any hidden effect combination that must be considered for future work. An investigation into what each factors' influence will be relative to the others was also needed as this will help determine the parameters for a test design decision matrix.

The analysis only showed a significant result for the main effect of *input mode* below the 5% level while below the 20% level a marginally significant result for the cross effect of *Viewing Mode* and *Field of view* was found. The main effect plots and interaction plots are shown in Appendix K.

Table 5.2.7: 2D Anova with interaction and significance shown

	Df	**Sum Sq**	**Mean Sq**	**F value**	**Pr(>F)**
Input	1	392.1	392.09	4.0434	0.04809
FieldofView	1	45.7	45.65	0.4708	0.49484
ViewMode	1	7.5	7.54	0.0778	0.78113
Input:FieldofView	1	0	0	0	0.99648
Input:ViewMode	1	137.5	137.45	1.4175	0.23773
FieldofView:ViewMode	1	199.2	199.19	2.0541	0.15612
Input:FieldofView:ViewMode	1	3.5	3.51	0.0362	0.84955
Residuals	72	6981.8	96.97		

It is useful to visualise the data used for the 2D experiment geometrically as was described in Section 3.7.1. The design cube as shown in Figure 5.2.9, the data points at the positive and negative plains are shown.

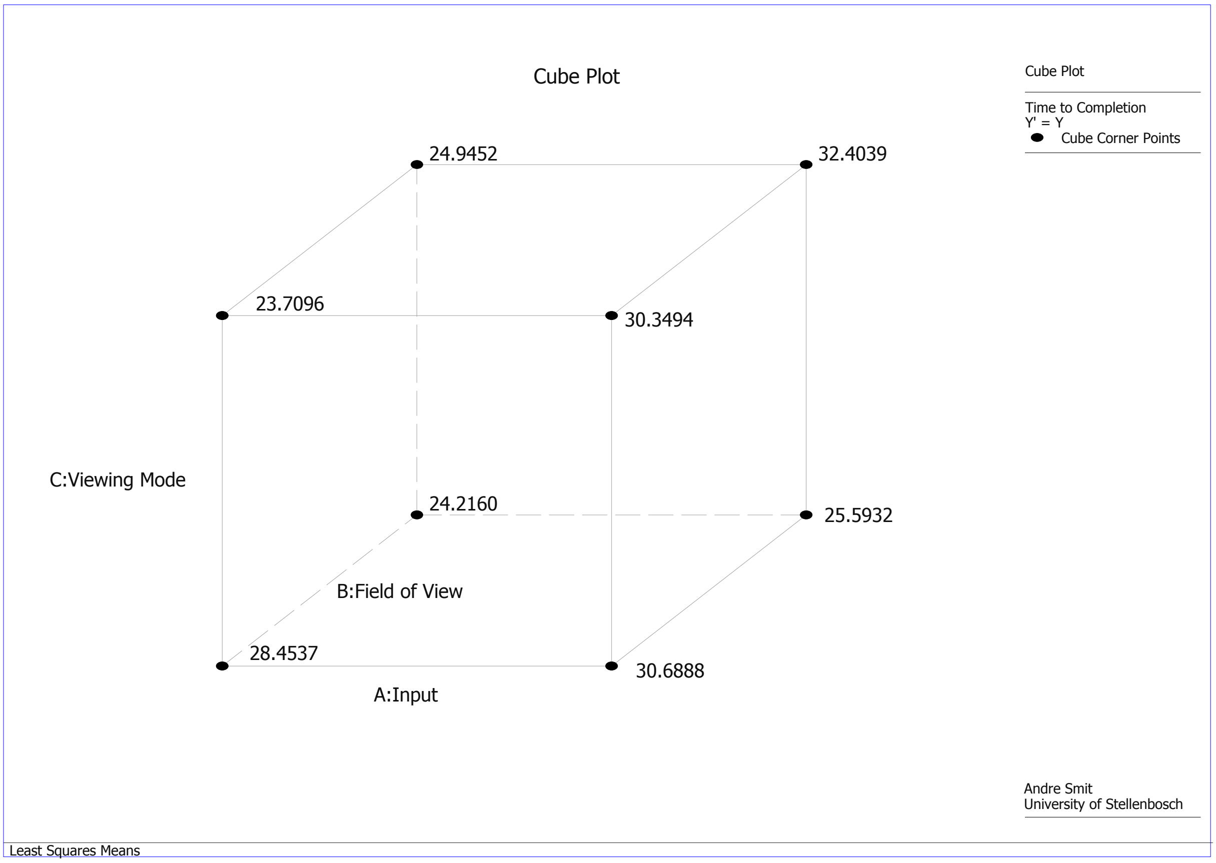


Figure 5.2.9: Geometric ANOVA Results, 2D planar experiment

When completing the Friedman hypothesis test, it was found that many smaller interactions were found to be responsible for the larger interactions. The minor interactions are shown in Table 5.2.8. As can be seen under haptic control with a full field - field of view, there was a major difference under 2D and 3D viewing modes. Further there was a less significant result for other treatment combinations. Similar results were found for the Wilcoxon Rank-sum test.

Table 5.2.8: Friedman Results for 2D Planar experiment

Significant Results	Friedman chi-squared	df =	p-value =
Haptic Full 2D vs. 3D	10	1	0.001565
Haptic 3D Full vs. POV	3.6	1	0.05778
Joystick POV 2D vs. 3D	3.6	1	0.05778
Full 2D Joystick vs. Haptic	3.6	1	0.05778
Full 3D Joystick vs. Haptic	3.6	1	0.05778

The data was also analysed using the Modde experimental design software. The residuals were analysed and showed the relative effect sizes. In order of effect from highest to lowest, the factorial effect on time to completion is, *Input Modality*, *Viewing Angle*, *Viewing Mode*.

As discussed earlier, to test which of the factors has the largest effect within the resulting ANOVA interactionsa post-hoc test will be required.

The test chosen is the Tukey and it was completed for the entire test set. The test proved useful in identifying the sets of treatments were significant in influencing the results obtained by the ANOVA. Other non-parametric tests such as the Friedman test yielded similar results. Below is a selected extract of the results that proved to be, the complete result set can be found in Appendix K.

The Tukey HSD test yielded confirmed the results from the ANOVA. Although no other significant interactions were found at the $\alpha = 0.05$ level, there is a marginally significant interaction between input mode and viewing mode at below $\alpha = 0.15$, as shown in Table 5.2.9. This implies that the even under the interaction of *viewing mode* and *field of view* the input mode still had the largest effect.

Table 5.2.9: 2D Planar - Tukey HSD test

	diff	lower	upper	p adj
Haptic-Joystick	4.428	0.03852	8.81748	0.04808
Viewing Mode & Field of View : Haptic:3D-Joystick:3D	7.0495	-1.1405	15.2395	0.11629

5.2.5.3 3D Obstacle ANOVA and Non-parametric Analysis

The ANOVA for the 3D obstacle testing did not reject the null hypothesis for the given data. The results obtained are shown in Table 5.2.10. As can be seen there is no significant effects or interactions. This may be attributed to the small sample size which would imply that there is a lack of variance. As the response variable is time to completion, which incorporates system speed, one can assume that for the given test, the system speed was the limiting factor and that which caused no significant differences, interaction plots are shown in Appendix K.

System operating limitations are an unavoidable part of any system. Here, however, it would seem that a major part of the problem can be attributed to the manner in which the system executes commands. Upon further inspection of user comment, it became apparent that a system limitation, namely “jerky motion” could be to blame as the user indicated that the jerky motion was “disconcerting”. This may be to blame, causing users to feel unsafe and thus act overly cautious, although further investigation is suggested for further study.

Table 5.2.10: 3D Obstacle ANOVA

Interaction	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Input	1	10.5	10.47	0.0743	0.7862
Field of view	1	0	0.041	0.0003	0.9865
ViewMode	1	113.3	113.348	0.8042	0.3737
Input:Field of view	1	112.6	112.593	0.7988	0.3753
Input:ViewMode	1	28.8	28.751	0.204	0.6533
Field of view:ViewMode	1	1.1	1.086	0.0077	0.9304
Input:Field of view:ViewMode	1	199.8	199.805	1.4176	0.2388
Residuals	56	7893.2	140.949		

The Friedman and Wilcoxon hypothesis test also failed to yield any significant results and as such no post-hoc analysis was deemed necessary for the 3D obstacle experiment.

It can be argued that there is no significant result for the analysis as a result of system limitations when considering time-to-completion of a given task in the 3D experiment only.

5.2.6 System Error Characteristic

System errors were a rare occurrence as most errors can be attributed to communication errors on the interpreter side. The risk of communication errors occurring increases the longer the system stays active.

A total of 19 system errors were observed which results in a total error rate of 13.2% of errors to experiments. The error rate was much higher for the 2D experiments, as shown in Figure 5.2.10. This adds merit to surmise that the errors are caused by a communication error.

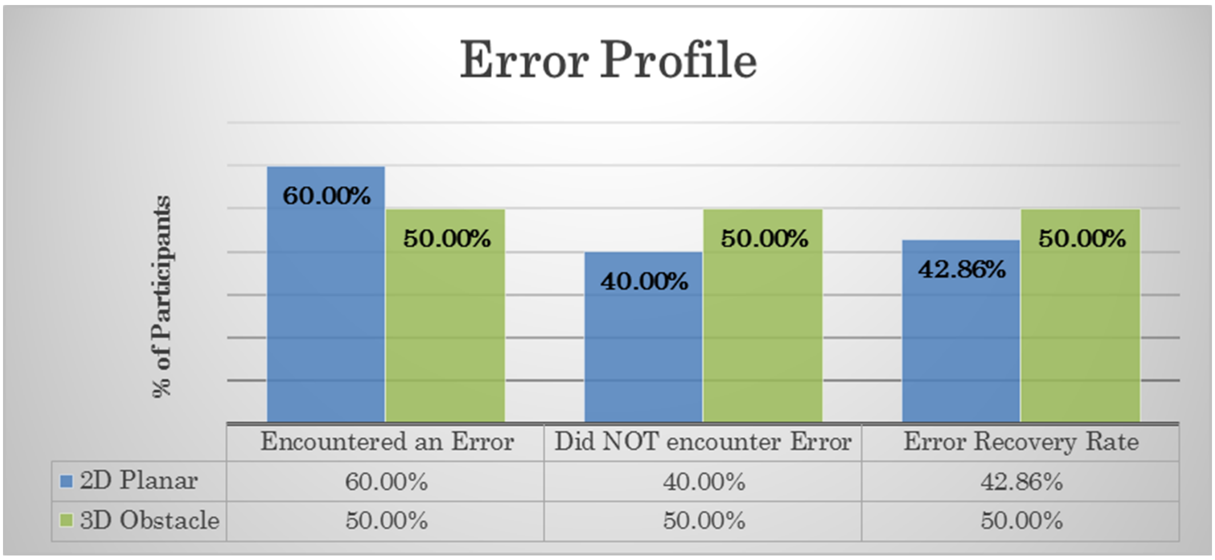


Figure 5.2.10: User Experiment Error Profile

The error recovery rate for both of the experiments is nearly 50%. System errors that could not be recovered were simple to reset. A reset is accomplished by isolating the system and forcing a home position far away from the work volume, which was made possible by the user interface. This procedure also forced a reset of the communication module. This was done by making use of the robot error reset command. This reset the robot communication module along with clearing any non-essential data. The user interface was then restarted.

Upon further analysis most of the errors were found to be caused by the communication module. Due to the protocol used for communication between the interface server and the robot controller, the server misses a crucial part of the command executed packet. This is unfortunately part of the robotic controller software that cannot be changed. A possible solution will be to insert another layer between controller and communication module, which function is to regulate communication. This may be completed by a automatic queued communication interruption functionality, which will be explored in Section 6.1.

5.2.6.1 Error Handling

As in any complex system, there will inevitably be some errors. The errors are most likely to come from four main components/aspects of the system. The first is mechanical machine failure. Fortunately mechanical failure was not encountered in the testing of the project, although this is a very real point of failure in the life-cycle of the system. The second is in signal conversion which can be found in any transducer within the system, although the component with the highest probability of failure is the F/T sensors. Due to the high speed at which the sensors operate the chances of erroneous communication or signal conversion is high, especially when the interaction between the interface (software) and hardware (sensor) is considered.

Further an operator my encounter software errors. Software errors are expected to be the most common, as it is the interface to most of the peripheral systems. Software errors can be caused by many types of malfunctions. From system logs, nearly all the system errors were caused by communication errors. As discussed in previous sections the communication protocol was to blame for most errors. In some cases it may be attributed to acknowledge packets becoming lost due to timing issues.

Electrical errors may also be encountered. This can be attributed to custom connections that get stressed as in the case of hard impacts to rigid bodies. Rigid and soft body impacts are a real risk as the user group was not trained before the commencement of the experiments.

5.2.6.2 Error Recovery

Error recovery is a key part of any system. The ease of recovery is a good measure of system design. The current system has a 50% user recovery rate. This means that the user is able to reset or overcome the error by themselves without the help of a technician, although all the errors could be overcome with help from a technician. This can be attributed to the system's inherent conservative/cautious design regime. User error did not result in a single incident of damage to person or property. The system erred on the side of caution in that an emergency stop of all system motion was initiated as soon as an error was detected. The user is then prompted to acknowledge the error and manually reset it from the user interface dashboard.

Error recovery was experienced as fairly simple, unless there was a major malfunction. This presented in a slow sideways motion of the output manipulator. This was caused by a communication loop, where the software interpolates to infinity causing a overflow in the position parameters. As a result the virtual model used inside the server for kinematic motion determination was adapted to add more filtering options in order to prevent this from happening in future.

5.2.7 Usability Results Discussion

As a measure of the overall performance one can also consider the average time to completion for each set of factors i.e. treatments. Table 5.2.11 shows the average time to completion for any given combination for the 2D planar experiments. As can be seen, the same task is performed using the joystick roughly the same for 3D viewing and the two perspectives. Furthermore there is a noticeable difference in completion time between 2D and 3D viewing for FF field of view when considering the viewing modes of nearly 7 seconds. There is also a noticeable difference of nearly 10 seconds between the haptic-FF-3D and joystick-POV-3D tests. There is, however, nearly no difference in average time to completion between the FF modality of the Joystick-2D vs. 3D tests.

Table 5.2.11: Average time to completion - 2D Planar Experiment

	Joystick		**Haptic**	
Field Of View / **Vision Mode**	**POV**	**FF**	**POV**	**FF**
2D:	28.45	24.22	30.69	25.59
3D:	23.71	24.95	30.35	32.40

From this one can deduce that Joystick-POV-3D is the highest ranked performance wise for the 2D planar test while Haptic-FF-3D is the lowest performing. When considering the results from Section 5.2.3 and 5.2.4, user preference is consistent with the timed results. Although the user preference between POV and FF are similar, POV does have a 2/5 majority vote. In contrast to the timed results, the most users preferred the 2D viewing mode to complete the experiments, getting double the votes that 3D viewing received. Furthermore when considering Section 5.2.4 specifically, the user ranking decision matrix ranks the best performing treatment as the second lowest ranked while the worst performing treatment is ranked second highest according to user preference. This shows that user preference may be contradictory to actual performance for a given measure.

One can also consider the collision data as a performance measure, as shown in Table 5.2.12. From this one may deduce that the worst performing treatment is Joystick-POV-3D with 13 collisions. This stands in contrast to results from the average time to completion. One can surmise that the average time to completion is much shorter due to the lack of accuracy with which the task is completed, although when considering the worst performing average time to completion treatment, we see that it also is a low performer with 12 collisions.

Table 5.2.12: 2D Planar experiment - Total number of collision for given treatment

	Joystick		Haptic	
Field Of View / **Vision Mode**	**POV**	**FF**	**POV**	**FF**
2D	7	6	9	8
3D	13	9	7	12

Similarly for the 3D experiment there are significant differences in the average time to complete the task, which is clearly shown in Table 5.2.13. The largest difference exists of nearly 7 seconds exists between Joystick-POV-3D and Joystick-FF-3D. Further the tasks have an average completion time between 30 and 32 seconds. This would lead one to believe that some system/user limitation was reached in the testing and requires further investigation to clarify. In terms of average time to completion the highest performing treatment is Joystick-FF-3D where as the worst performing is the Joystick-POV-3D, although there is a negligible difference between the latter and Haptic-FF-2D.

This final result comes as a surprise as user preference data would lead one to believe that the haptic input modality will be the highest performing as it is highest preferred. Furthermore the field of view that is highest preferred is POV although it does not consistently perform better in terms of average time to completion. In contrast to this, the 3D viewing mode does perform slightly better and is highest on the user preference list.

Table 5.2.13: Average time to completion - 3D Obstacle Experiment

	Joystick		Haptic	
Field Of View / **Vision Mode**	**POV**	**FF**	**POV**	**FF**
2D:	29.77	31.04	32.00	32.31
3D:	32.42	25.93	26.87	31.58

The user ranking shows that the first choice is Haptic-POV-3D which is the second best performer, while the lowest ranked is Haptic-FF-2D, which is consistent with the average time to completion data.
Furthermore, if the collision data is considered in Table 5.2.14, the user preference ranking

once again stands in contrast to collision data where Joystick-POV-3D is ranked highest while having the most collisions. On the other hand Haptic-FF-3D is performing the best with the least amount of collisions while having a mediocre ranking in terms of user preference.

Table 5.2.14: 3D Obstacle experiment - Total number of collision for given treatment

	Joystick		**Haptic**	
Field Of View / **Vision Mode**	**POV**	**FF**	**POV**	**FF**
2D	4	6	5	4
3D	8	5	2	1

From these results one can argue that time to completion is a function accuracy, task complexity and user preference as users are more likely to like some modality that they are comfortable with and as a result will be able to complete tasks faster. User performance is also a trade-off between accuracy and time. In general one can argue that there is a nominal time required to complete a given task with limitations and operating parameters. In order for a teleoperation system to be successful the system must enable the user to complete this task in the same as with working directly with the work volume, although one can make allowances for time delays in the system.

From the user responses, one can judge that the users are satisfied with the system performance and accuracy although there are some matters that need addressing. The concerns raised will be discussed in the future work section of this thesis.

Further one can consider the correlation between other performance measures. If one considers the 3D Obstacle experiment, both the user and a professional in system usage was asked to rate the accuracy with which a ring was placed. From Table 5.2.15, the logic of the table is clear, if users feel that they achieved the goal more accurately the pro rating should increase as well. Also if there are more collisions then the user should take longer to complete the experiment, although the value of the correlation lies in the increase shown in TTC the pro-rating increases. Similarly the results from the 2D Planar experiments showed that for a large increase in TTC there is a very small increase in the number of collisions.

Interestingly for a given decrease in TTC there was an increase in user rating of accuracy and a much larger decrease in the pro rating. This could imply that users also take into account the time required to achieve higher task accuracy and associates a fast time with an higher accuracy.

Table 5.2.15: Correlation for 3D obstacle experiment

	User rating	***Pro Rating***	***TTC***	***# Collisions***
User rating	1			
Pro Rating	0.708856085	1		
TTC	-0.05960382	0.125887	1	
# Collisions	-0.29923277	-0.29234	0.305921	1

When considering other performance measures gathered for the 2D experiments, the data yields the correlation results shown in Table 5.2.16. Here it is shown that for a given decrease in TTC the decrease in number of collisions will be far less than the decrease in severity of a collision.

Table 5.2.16: 2D planar experiment - Collision correlation

	TTC	*# Collisions*	*severity*
TTC	1		
# Collisions	0.01768	1	
severity	0.03992	0.79673	1

Chapter 6

Conclusions and Limitations

The system showed a significant improvement, when compared to the initial implementation, in both technical specification and usability factors. This can be attributed not only to the improvements in communication hardware and hardware speed but also to the software components developed to manage the various processes and communication. The introduction of a more secure communication protocol, namely TCP/IP, proved to have a minimal effect on communication latency while increasing the security of data when compared to the UDP protocol used in earlier implementations. The observed communication latency was negligible even when tested through a distributed local network, for the given experimental setup.

The following results were found in response to the research objective that call for a study to be conducted, to compare the initial and current system implementations to show the differences in technical performance. General system response (system rise-time) required improvement. Rise-time for system response was reduced from 6 seconds to 0.6 seconds, which is only marginally outside the performance specification of 0.5 seconds for system response rise time. Further, the system showed a general improvement in stability. The initial implementation had a severe jitter while in haptic operation. Although this phenomenon is still present in the latest implementation, it is only found when system parameters are unrestricted and sharp input is given or a continual large input is given. Further stability was improved by making use of a PI controller and filtering input and limiting output, negating the effects of integral wind-up. Using force as a control variable is not ideal, although the use of a kinematic model with variable parameters helped in mapping the response of the system.

Further the system showed improvement in force tracking, which was separated into two main components, namely force magnitude and force-angle. Force magnitude error was first considered, as the FT sensors have varying force measuring characteristics for each axis, each axis was considered independently. This data was then used to calculate the final force magnitude error at $\alpha = 0.1$ as 0.03 $\pm$0.11 Newton. This was a marginal improvement over the initial implementation's 0.123 $\pm$0.034 Newton. Further the force angle was considered, the initial implementation did not clearly define the force angle, so a pragmatic approach was adopted. The force angle is comprised of three angles. Decomposed from F_x, F_y and F_z, the angles are α, β and ϵ respectively.

The force angle as shown to be $\alpha = 1.821.78\,\text{deg}$, $\beta = 1.542.07\,\text{deg}$ and $\epsilon = 1.671.67\,\text{deg}$. No assertion could be made about the observed angular error. This was a result of nonreplicatable results from Pretorius (2012), which claimed a 0.07 deg angular error, with no explanation of its origin. This result does however show that for a worst-case scenario the error will be a maximum of 4.15 deg and a minimum of 1.23 deg. Although not of great importance for the experimentation conduced for the current case study, it will however become problematic for sensitive experiments such as micro material handling.

Although video feeds where incorporated into the system, the 3D high resolution feed was unable to be streamed successfully on the local network. The latency for the video feed exceeded the 1.5 second point and it was decided that the stream must be captured and streamed directly to the display computer. This reduced video latency to the same region as the IP camera, where it is not noticeable by any of the users tested. The feed was supplied via a direct feed to an on-board capture device located in the server. The display was then directly sent to the user interface, the video feeds were executed in a separate feed than the rest of the UI.

Further the following was found in response to the research objective to have determined the usability and user preference in order to facilitate future testing and use cases. The user-centric usability case study showed that in general users were comfortable using the system. User preference and ranking data was used to construct an experimental decision matrix. This was shown in Section 5.2.4, the value of matrix lies within the experimental framework, allowing the experimenter to make quick decisions based on user preference. Users are generally more comfortable in a specific modality, although the decision matrix allows the experimenter to determine (for the current system) which modality to choose given a set of limiting factors. This matrix was constructed for a small user group, but as was shown, Nielsen (1994*b*) only 5 users are required to find usability issues. Further usability results were discussed at length in Section 5.2.7.

It was found that users preferred haptic control joystick control and POV over full-field viewing angles, under a 3D vision condition. This also showed how user preference stands in contest with TTC data. In general considering average time to completion data users performed better for both types of experiments using the full field modality, completing tasks in less time. Further user preference data is corroborated, where users performed best under the 3D vision condition and haptic control, for input and vision condition factors respectively.

It was also found that the 3D obstacle experiment did not yield any clear interaction results due to a system limitation. This system limitation was most likely the users' perception of the jitter that made users uncomfortable while using the system, causing an overly cautious approach to the experiments. The ANOVA of the 2D experiments did yield significant differences in ***Input modality*** at $\alpha = 0.05$ and a marginally significant at $\alpha = 0.11$ interaction effect between viewing mode, input mode. The interaction effect was found for the 3D viewing condition and varying input.

It is recommended that the experiments be conducted with a larger test group. The larger test group will yield more data that can be used to find a more accurate statistical model for user performance, which can assist in inference. Three of the most common factors influencing user-centric performance of the system were identified and tested. A varying array of data was collected and helped identify possible weaknesses in the system.

The system serves as a proof of concept test bench not only for user studies, as is the case with the case-study presented, but also as a test bench for hardware and software components. Although the system was developed on a combined robotic unit, consisting of 3 robots, it can be easily expanded to use more than one robotic controller that uses a RJ45 - TCP/IP connection. This is enabled using a modular software design. Communication modules will require further robot-specific development in order to enable mixed communication between robots.

Overall the system showed improvements in most technical performance measures. Further the system was found to be easier to use and more responsive. Human-factor analysis revealed that users preferred the haptic input modality and there was no notable significant improvement for the incorporation of 3D video. When considering TTC user performed better under joystick input. Further users preferred the point of view viewing angle as it is the more natural of the two and allowed for discerning some depth from the image.
The system serves as a proof of concept that shows the improvement that was possible with minor capital injection. This proof of concept shows what is possible with off-the-shelf hardware components and software based teleoperation and is a good starting point for the project for further development.

Concluding Remarks:
It was found that through the application of the SE process and by making use of a combination of developmental process models, UCSD and XP a successful system could be implemented. The software-based architecture allowed for a modular system design that could be user focused in its approach. Further human-factor experimentation revealed that the system is usable and that the use of haptic input devices promotes ease of use and improves user performance, thus reconfirming that the use of haptic feedback should be continued in future systems. User preference was also analysed to contribute to future user-centric development. The preference data was constructed into a weighted decision matrix for use in testing scenarios.

6.1 Recommended Future Work

The system shows promise for future development. With this comes specific limitations that are currently part of the system that requires improvement. Although the software has been developed to be locally hosted on a application server, for true distributed teleoperation to be a reality it will have to be moved to an external server. This will bring its own set of challenges which the current system cannot resolve. The user interface (UI) at present integrates all the modular components of software. The UI and other modular components such as the force torque reader and interpreter and kinetic model, were purposefully developed in C-sharp utilising the .NET framework. This allows for a C-sharp server side execution and ASP.NET to be used for the user (web-based) user interface. This was done in order to facilitate a move to ASP.NET for (internet based) interfacing with the C-sharp server-side execution of the developed applications.

The move will not be seamless and requires further development of many sub systems. Especially the software based controller. It is recommended that the system should be developed to incorporate advanced controller(s), as it is far more robust to varying time delays that will be inevitable in an internet-based system.

System architecture will also have to be adapted for future applications. The need for local servers is a possible solution to try helping reduce the latency effects produced by interfacing directly with the robotic controllers. Although the applicability of the robotic controllers in future applications. The current jagged / jerky response is caused directly by the robotic controller. This was remedied by using small incremental steps in generating a displacement profile. But this is not a long term solution.

The current robotic controller architecture does not allow for input commands to interrupt the current command being executed. This is a major hurdle that must first be overcome before the migration to a wholly internet based system. The use of custom robotic controllers are recommended and has already been proven to be successful in that allows a dynamic instruction set, in essence allowing a new command to interrupt an old one. The advanced kinematics suite and controller was developed by Kinematics M and has been shown to be successful. However this raises issues in the overall safety of when using the system and in effect requires another layer of protection to be required to ensure that the interrupt command does not put any of the elements at risk.

Furthermore advanced encoding is required for the video feeds. Video is the most data intensive component of the system. Compression is crucial to the feasibility of the systems, especially where there is limited bandwidth for communication. Current encoders have already proved to be very efficient such as the H.264 codec and are widely employed for video streaming. As was the case with the IP camera used for full-field viewing, although there will be inherent time delays present and a solution must be found to reduce its effects.

The selection and testing of application dependent grippers are also recommended as is the case in Read *et al.* (2013) which proved the efficacy of the current system. The system was used in a micro milling application.

List of References

Ambrose, R., Aldridge, H., Askew, R., Burridge, R., Bluethmann, W., Diftler, M., Lovchik, C., Magruder, D. and Rehnmark, F. (2000). Robonaut: Nasa's space humanoid. *Intelligent Systems and their Applications, IEEE*, vol. 15, no. 4, pp. 57–63.

Asai, T., Kojima, J., Asakawa, K. and Iso, T. (2000). Inspection of submarine cable of over 400 km by auv. In: *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on*, pp. 133–135. IEEE.

Ballantyne, G. (2002). Robotic surgery, telerobotic surgery, telepresence, and telementoring. *Surgical Endoscopy*, vol. 16, no. 10, pp. 1389–1402.

Ballantyne, G., Moll, F. *et al.* (2003). The da vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *The Surgical clinics of North America*, vol. 83, no. 6, p. 1293.

B&B Electronics (2013). *Ethernet Cables - RJ45/Colors & Crossover* [Online]. Available at: http://www.bb-elec.com/Learning-Center/All-White-Papers/Ethernet/Ethernet-Cables-RJ45-Colors-Crossover.aspx [2013, June 20].

Blanchard, B.S. and Fabrycky, W.J. (2010). *Systems Engineering and Analysis*. Prentice Hall International Series in Industrial and Systems Engineering. Pearson. ISBN 9780137148431.

Brown, J. (1998). *Passive implementation of multibody simulations for haptic display*. Ph.D. thesis, Northwestern University.

Bucknell University (2012). *Robot Manufacturers* [Online]. Available at: http://www.eg.bucknell.edu/~robotics/manufctr.html [2012, June 24].

Casper, J. and Murphy, R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 33, no. 3, pp. 367–385.

Centers for Medicare & Medicaid Services (2005). Selecting a develpment approach. [Online] http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf [2013, August 10].

Chellali, R. (2009). Tele-operation and human robots interactions. In: *Proc. of 18th IEEE Int. Symp. on Robot & Human Interactive Communication (RO-MAN 2009), Toyama, Japan.*

Chen, J., Haas, E. and Barnes, M. (2007). Human performance issues and user interface design for teleoperated robots. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1231–1245.

Chen, J.Y. and Thropp, J.E. (2007). Review of low frame rate effects on human performance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 6, pp. 1063–1076.

CHRISTIE (2012). *Christie CAVE, Cave Automatic Virtual Environment* [Online]. Available at: http://www.christiedigital.com/en-us/3d/virtual-simulation/virtual-reality-simulator/pages/cave-automatic-virtual-environment.aspx [2012, June 16].

Committee on theoretical foundation for decision making in engineering design (2001). *Theoretical Foundations for Decision Making in Engineering Design*. The National Academies Press [Online]. http://www.nap.edu/openbook.php?record_id=10566 [2013, June 6].

Cui, J., Tosunoglu, S., Roberts, R., Moore, C. and Repperger, D. (2003). A review of teleoperation system control. In: *Proceedings of the 2006 Florida Conference Recent Advances in Robotics (FCRAR), Florida Atlantic University, FL*.

da Vinci Surgery (2012). *The da Vinci Surgical System* [Online]. Available at: http://www.davincisurgery.com/davinci-surgery/davinci-surgical-system/ [2012, October 11].

DAU (2012). *Christie Technical Performance Measurement (TPM)* [Online]. Available at: https://acc.dau.mil/CommunityBrowser.aspx?id=243206 [2012, August 5].

Duff, E., Usher, K., Taylor, K. and Caris, C. (2007). Web-based tele-robotics revisited. In: *Proceedings of Australian Conference on Robotics and Automation. Brisbane, Australia*. Citeseer.

Elvander, J. and Hawkes, G. (2012). Rovs and auvs in support of marine renewable technologies. In: *Oceans, 2012*, pp. 1–6. IEEE.

Fabrlzio, M.D., Lee, B.R., Chan, D.Y., Stoianovici, D., Jarrett, T.W., Yang, C. and Kavoussi, L.R. (2000). Effect of time delay on surgical performance during telesurgical manipulation. *Journal of endourology*, vol. 14, no. 2, pp. 133–138.

Ferretti, G., Magnani, G. and Rocco, P. (2009). Web-based industrial robot teleoperation: An application. *Web-Based Control and Robotics Education*, pp. 249–266.

Fong, T., Thorpe, C. and Baur, C. (2002). Robot as partner: Vehicle teleoperation with collaborative control. *Multi-Robot Systems: From Swarms to Intelligent Automata*, vol. 1.

Fowler, M. and Highsmith, J. (2001). The agile manifesto. *Software Development*, vol. 9, no. 8, pp. 28–35.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, vol. 32, pp. 675–701.

Gage, D. (1995). Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts. Tech. Rep., DTIC Document.

geomagic (2013). *PHANTOM DESKTOP HAPTIC DEVICE* [Online]. Available at: http://www.sensable.com/haptic-phantom-desktop.htm [2013, April 22].

Gitlow, H.S., Oppenheim, A.J., Oppenheim, R. and Levine, D.M. (2005). *Quality Management*. McGraw-Hill International Edition.

Glover, C., Russell, B., White, A., Miller, M. and Stoytchev, A. (2009). An effective and intuitive control interface for remote robot teleoperation with complete haptic feedback. In: *Proceedings of the 2009 Emerging Technologies Conference (ETC), Ames, IA, USA*.

Goertz, R.C. (1952). Fundamentals of general-purpose remote manipulators. *Nucleonics*, vol. 10, no. 11, pp. 36–42.

Halme, A., Suomela, J. and Savela, M. (1999). Applying telepresence and augmented reality to teleoperate field robots. *Robotics and autonomous systems*, vol. 26, no. 2, pp. 117–125.

Haskins, C., Forsberg, K. and Krueger, M. (2007 August). Systems engineering handbook: A guide for system life cycle processes and activities.

Haskins, C. *et al.* (2006). Systems engineering handbook. *INCOSE. Version*, vol. 3.

Hayward, V. and Astley, O.R. (2000). Performance measures for haptic interfaces. In: *Robotics Research*, pp. 195–206. Springer.

Hayward, V., Astley, O.R., Cruz-Hernandez, M., Grant, D. and Robles-De-La-Torre, G. (2004). Haptic interfaces and devices. *Sensor Review*, vol. 24, no. 1, pp. 16–29.

Hecht, D. and Reiner, M. (2009). Sensory dominance in combinations of audio, visual and haptic stimuli. *Experimental brain research*, vol. 193, no. 2, pp. 307–314.

Hosek, A. and Erin, H. (2011). Analysis of variance (anova). Course Notes. [Online]. Available at: http://ekhartman.berkeley.edu/work/ANOVA.pdf [2013, July 5].

Hurmuzlu, Y. and Nwokah, O. (2001). *The mechanical systems design handbook: modeling, measurement, and control*. CRC.

International Ergonomics Association (2013). *Definition of Ergonomics* [Online]. Available at: http://iea.cc/01_what/What%20is%20Ergonomics.html [2013, August 1].

ISO/IEC (2002). Iso/iec 15288:2002(e) - systems engineering - system life cycle processes.

Jet Propulsion Laboratory (2012). *NASA Facts - Mars Exploration Rover* [Online]. Available at: http://marsrover.nasa.gov/newsroom/factsheets/pdfs/Mars03Rover041020.pdf [2012, September 12].

Kaber, D., Riley, J., Zhou, R. and Draper, J. (2000). Effects of visual interface design, and control mode and latency on performance, telepresence and workload in a teleoperation task. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, pp. 503–506. SAGE Publications.

Kahney, L. (2001 September). *Robots Scour WTC Wreckage* [Online]. Available at: http://www.wired.com/science/discoveries/news/2001/09/46930?currentPage=all [2013, April 15].

Khan, M. and Gu, J. (2012). Web based teleoperation architecture and h. 264 video encoder. In: *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pp. 1–4. IEEE.

Kim, D., Kim, K., Kim, K. and Cha, S. (2001). Dexterous teleoperation for micro parts handling based on haptic/visual interface. In: *Micromechatronics and Human Science, 2001. MHS 2001. Proceedings of 2001 International Symposium on*, pp. 211–217. IEEE.

Kim, T. and Biocca, F. (1997). Telepresence via television: Two dimensions of telepresence may have different connections to memory and persuasion. *Journal of Computer-Mediated Communication*, vol. 3, no. 2. ISSN 1083-6101.

Kioskea (2013). *Agile Methods (RAD, XP)* [Online]. Available at: http://en.kioskea.net/contents/236-agile-methods-rad-xp [2013, November 26].

Lee, D. and Spong, M.W. (2005). Bilateral teleoperation of multiple cooperative robots over delayed communication networks: theory. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 360–365. IEEE.

lerosen4 (2012 August). *Communications in the 21st Century - Part 5: Telepresence, Telerobotics* [Online]. Available at: http://www.21stcentech.com/communications-21st-century-part-5-telepresence-telerobotics/ [2013, April 15].

Li, B., Ma, S., Liu, J., Wang, M., Liu, T. and Wang, Y. (2009). Amoeba-i: a shape-shifting modular robot for urban search and rescue. *Advanced Robotics*, vol. 23, no. 9, pp. 1057–1083.

Li, L., Wei, Q., Hou, Z. and Zhao, L. (2011). Design and realization of the experimental platform of space robot bilateral teleoperation system. In: *Control Conference (CCC), 2011 30th Chinese*, pp. 3968–3972. IEEE.

Lichiardopol, S. (2007). A survey on teleoperation. *Dept. Mech. Eng., Dynamics Control Group, Technische Universiteit Eindhoven, Eindhoven, Dept., Mech. Eng., Dyn. Control Group, The Netherlands, Tech. Rep. DCT2007*, vol. 155.

Lowry, R. (1999). *One-Way Analysis of Variance for Independent Samples* [Online]. Available at: http://vassarstats.net/textbook/ch14pt2.html [2013, September 3].

Lowry, R. (2011). *The Wilcoxon Signed-Rank test* [Online]. Available at: http://www.vassarstats.net/textbook/ch12a.html [2013, September 15].

Maguire, M. (2001). Methods to support human-centred design. *International journal of human-computer studies*, vol. 55, no. 4, pp. 587–634.

Mansfield, C.L. (2005 March). *Robonaut shows sensitive side* [Online]. Available at: http://www.nasa.gov/missions/science/robo_sensors.html [2012, December 6].

Marescaux, J., Leroy, J., Rubino, F., Smith, M., Vix, M., Simone, M. and Mutter, D. (2002). Transcontinental robot-assisted remote telesurgery: feasibility and potential applications. *Annals of surgery*, vol. 235, no. 4, p. 487.

Maurer, F. and Martel, S. (2002). Extreme programming. rapid development for web-based applications. *Internet Computing, IEEE*, vol. 6, no. 1, pp. 86–90.

Mitsuishi, M., Sugita, N., Nagao, T. and Hatamura, Y. (1996). A tele-micro machining system with operational environment transmission under a stereo-sem. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 3, pp. 2194–2201. IEEE.

Murphy, D. (1991). Advances and experience with teleoperated systems incorporating remote presence. Tech. Rep., DTIC Document.

Nawrocki, A. (2003). *Space Tele-Robotics*. Ph.D. thesis, Californio Coast University.

Nielsen, J. (1994*a*). Enhancing the explanatory power of usability heuristics. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pp. 152–158. ACM.

Nielsen, J. (1994*b*). Heuristic evaluation. *Usability inspection methods*, vol. 24, p. 413.

Nielsen, J. (2000 March). *Why You Only Need to Test with 5 Users* [Online]. Available at: http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ [2013, May 23].

Ohio University Russ College of Engineering and Technology (2012). What is engineering management? [Online]. Available at: http://engineering.online.ohio.edu/engineering-management/engineering-management-degree/what-is-em/ [2013, January 21].

Opperman, G. (2009). Micro-scale machine vision using a robot mounted camera in a google earth manner. Final year project, University of Stellenbosch.

Pallot, M., Trousse, B., Senach, B., Scapin, D. *et al.* (2010). Living lab research landscape: From user centred design and user experience towards user cocreation. In: *First European Summer School'Living Labs'*.

Passenberg, C., Peer, A. and Buss, M. (2010). A survey of environment-, operator-, and task-adapted controllers for teleoperation systems. *Mechatronics*, vol. 20, no. 7, pp. 787–801.

Paulk, M.C. (2001). Extreme programming from a cmm perspective. *Software, IEEE*, vol. 18, no. 6, pp. 19–26.

Pavani, F., Spence, C. and Driver, J. (2000). Visual capture of touch: Out-of-the-body experiences with rubber gloves. *Psychological Science*, vol. 11, no. 5, pp. 353–359.

Pearson Education (2011). *Wilcoxon Signed Ranks Test: Nonparametric Analysis for Two Related Populations*. [Online]. Available at: http://wps.prenhall.com/wps/media/objects/11886/12171343/OnlineTopics/bbs12e_onlinetopic_ch12-8.pdf [2013, October 1].

Perfetti, C. (2010 June). *Are you really prepared for your usability study? The three steps for success* [Online]. Available at: http://www.uie.com/articles/usability_testing_three_steps/ [2012, August 8].

Pretorius, J. (2012). *Development and Implementation of a Telerobotic System with Video and Haptic Feedback*. Master's thesis, University of Stellenbosch.

ProfIT Labs Ltd. (2013). *Development process*. [Online]. Available at: http://www.profit-labs.com/development-process/ [2013, August 25].

Pruim, R. (1999). ANOVA: Analysis of Variation. Class Notes (Math 243). Michgan, USA: Calvin College.

Rait, A. and Mukherjee, S. (na). Teleoperation with haptics. ME, ITT Delhi.

Ralston, J., Hainsworth, D., Reid, D., Anderson, D. and McPhee, R. (2001). Recent advances in remote coal mining machine sensing, guidance, and teleoperation. *Robotica*, vol. 19, no. 05, pp. 513–526.

Rayman, R., Croome, K., Galbraith, N., McClure, R., Morady, R., Peterson, S., Smith, S., Subotic, V., Van Wynsberghe, A. and Primak, S. (2006). Long-distance robotic telesurgery: a feasibility study for care in remote environments. *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 3, pp. 216–224.

Read, S., Van der Merwe, A.F., Dirkse Van Schalkwyk, T., Mueller, M., Smit, A. and Bezuidenhout, M. (2013 January). Micro machining using a motoman sda 10. Unpublished.

Read, S.E.A. (2012). *The Development of a Robotic Coarse-to-Fine Positioning System*. Master's thesis, University of Stellenbosch.

Resceanu, I. and niculescu, M. (2007 December). Telerobotics–control over the internet. In: *5th WSEAS Int. Conf. on ENVIRONMENT, ECOSYSTEMS and DEVELOPMENT*, pp. 341–346.

Resceanu, I. and Niculescu, M. (2008 March). Real time monitoring in web telerobotics. *WSEAS Trans. Sys. Ctrl.*, vol. 3, no. 3, pp. 184–194. ISSN 1991-8763.

Reynolds, R., Burke, W. and Advincula, A. (2005 April). *Preliminary Experience With Robot-Assisted Laparoscopic Staging of Gynecologic Malignancies.* [Online]. Available at: http://openi.nlm.nih.gov/detailedresult.php?img=3015578_jsls-9-2-149-g01&req=4 [2013, January 15].

robotmatrix.org (). *SCARA ROBOT* [Online]. Available at: http://www.robotmatrix.org/SCARARobotic.htm [2012, August 10].

robotmatrix..org (2012). *CARTESIAN or GANTRY ROBOT* [Online]. Available at: http://www.robotmatrix.org/CartesianGantryRobot.htm [2012, August 10].

RobotWorx (2013). *What are the main types of robots?* [Online]. Available at: http://www.robots.com/faq/157/what-are-the-main-types-of-robots [2012, August 2].

Rosenberg, L.B. (1993). The effect of interocular distance upon operator performance using stereoscopic displays to perform virtual depth tasks. In: *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pp. 27–32. IEEE.

Sanfeliu, A., Hagita, N. and Saffiotti, A. (2008). Network robot systems. *Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 793–797.

Sanz, P., Prats, M., Ridao, P., Ribas, D., Oliver, G. and Ortiz, A. (2010). Recent progress in the rauvi project: A reconfigurable autonomous underwater vehicle for intervention. In: *ELMAR, 2010 PROCEEDINGS*, pp. 471–474. IEEE.

Sarris, Z. and ATLAS, S. (2001). Survey of uav applications in civil markets (june 2001). In: *The 9 th IEEE Mediterranean Conference on Control and Automation (MED'01)*.

Sato, T., Ichikawa, J., Mitsuishi, M. and Hatamura, Y. (1994). A new micro-teleoperation system employing a hand-held force-feedback pencil. In: *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1728–1733. IEEE.

Schwartz, I. (2000). Primus: autonomous driving robot for military applications. In: *AeroSense 2000*, pp. 313–323. International Society for Optics and Photonics.

Science Daily (2007 June). *Rescue Robot Tests To Offer Responders High-Tech Help* [Online]. Available at: http://www.sciencedaily.com/releases/2007/06/070609112916.htm [2012, April 9].

SERENA (2007 June). *An introduction to agile software developement.* [Online]. Available at: http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf [2012, August 26].

Sheldon, M.R., Fillyaw, M.J. and Thompson, W.D. (1996). The use and interpretation of the friedman test in the analysis of ordinal-scale data in repeated measures designs. *Physiotherapy Research International*, vol. 1, no. 4, pp. 221–228.

sheng Liu, W. and Li, Y. (2012). The research for control strategies and methods of teleoperation system. In: *World Automation Congress (WAC), 2012*, pp. 1–4. IEEE.

Sheridan, T. (1986). Human supervisory control of robot systems. In: *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 808–812. IEEE.

Sheridan, T. (1989). Telerobotics. *Automatica*, vol. 25, no. 4, pp. 487–507.

Sheridan, T. (1995). Teleoperation, telerobotics and telepresence: A progress report. *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214.

Siciliano, B. and Khatib, O. (2008*a*). *Springer handbook of robotics.* Springer.

Siciliano, B. and Khatib, O. (2008*b*). *Springer handbook of robotics.* Springer, pp.751–755.

Smith, G. (2008). *The Effect of Latency on Rescue Robot Control.* Master's thesis, University of Cape Town.

Smith, P.F. (2010). *Types of Data, Descriptive Statistics, and Statistical Tests for Nominal Data.* [Online]. Available at: http://www.accp.com/docs/bookstore/biosampl.pdf [2013, August 9].

Smith-Atakan, S. (2006). *Human-Computer Interaction.* Thomson Learning.

Sonnenwald, D.H., Maglaughlin, K.L. and Whitton, M.C. (2004). Designing to support situation awareness across distances: an example from a scientific collaboratory. *Information Processing & Management*, vol. 40, no. 6, pp. 989–1011.

SRI International (2005 March). *DARPA Selects SRI International to Lead Trauma Pod Battlefield Medical Treatment System Development Program.* [Online]. Available at: http://www.sri.com/newsroom/press-releases/darpa-selects-sri-international-lead-trauma-pod-battlefield-medical-treatmen [2012, December 18].

Sridhar (2011 January). *DHCP (Dynamic Host Configuration Protocol)* [Online]. Available at: http://myunixlab.wordpress.com/2011/01/30/dhcp/ [2012, October 8].

Stark, L., Kim, W., Tendick, F., Hannaford, B., Ellis, S., Denome, M., Duffy, M., Hayes, T., Jordan, T., Lawton, M. *et al.* (1987). Telerobotics: Display, control, and communication problems. *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 67–75.

Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, vol. 42, no. 4, pp. 73–93.

Tanaka, F. and Takahashi, T. (2011). Linking children by telerobotics: Experimental field and the first target. In: *Proceedings of the 6th international conference on Human-robot interaction*, pp. 267–268. ACM.

Tanikawa, T. and Arai, T. (1999). Development of a micro-manipulation system having a two-fingered micro-hand. *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 1, pp. 152–162.

Visser, W. and J.G. (2012 May). Software engineering a brief introduction. Unpublished Class Notes (Altech 2012 SE Short Course). Stellenbosch: University of Stellenbosch.

VRLogic (2012). *6-DOF DELTA* [Online]. Available at: http://www.vrlogic.com/html/6-dof_delta_haptic_device_.html [2012, August 28].

Walker, A. (2012 May). *ISO 155288 - Systems Engineering standard* [Online]. Available at: http://theywalkedtogether.blogspot.com/2012/05/iso-15288-systems-engineering-standard.html [2013, September 2].

Wall, J., Jacques Marescaux, M. *et al.* (2013). History of telesurgery. In: *Telemicrosurgery*, pp. 15–18. Springer.

Webopedia (2012). *Decoupled* [Online]. Available at: http://www.webopedia.com/TERM/D/decoupled.html [2012, September 6].

Webster, G. (2012 January). *Durable NASA Rover Beginning Ninth Year of Mars Work* [Online]. Available at: http://marsrover.nasa.gov/newsroom/pressreleases/20120124a.html [2012, December 4].

Whitley, E. and Ball, J. (2002). Review statistics review 6: Nonparametric methods. *Critical Care*, vol. 6, pp. 509–513. [Online]. http://ccforum.com/content/6/6/509 [2013, September 6].

Williams II, R. (1998). Cable-suspended haptic interface. *International Journal of Virtual Reality*, vol. 3, no. 3, pp. 13–21.

Wynekoop, J. and Russo, N. (1997). Studying system development methodologies: an examination of research methods. *Information Systems Journal*, vol. 7, no. 1, pp. 47–65.

YASAKAWA Motoman (2011). *MotoCom SDK - Communication Software Development Kit* [Online]. Available at: http://www.motoman.com/datasheets/MotoCom%20SDK.pdf [2012, January 12].

YASAKAWA Motoman (2012). *SDA10D* [Online]. Available at: http://www.motoman.de/de/produkte/roboter-nx-xrc-und-dx/product-view/?tx_catalogrobot_pi1%5Buid%5D=590&cHash=3d109e303e3f6ddc1630b2d9d4609be3 [2012, September 26].

Zhang, B., Wang, J. and Fuhlbrigge, T. (2010). A review of the commercial brain-computer interface technology from perspective of industrial robotics. In: *Automation and Logistics (ICAL), 2010 IEEE International Conference on*, pp. 379–384. IEEE.

Zhao, J.L. (2012). Mis541a - application development approaches. [Online]. http://mis541.wikidot.com/system-development-approaches [2013, August 1].

Appendix A

Post - Experimentation Surveys

A.1 2D Experiment - Post Completion Survey

2D Planar Survey

Personal Information

1. Please enter your personal information:

Name: [ ]

2. Please supply the following information:

Email Address: [ ]

Age [ ]

***3. Please select your gender:**

- ○ Female
- ○ Male

Overall Impressions

When considering the experiments you just completed (2D fixed planar), please answer the following questions.

***4. Which of the Input Modes do you prefer?**

- ○ No preference
- ○ Joystick Control
- ○ Haptic Control

***5. Which of the Viewing Angles do you prefer?**

- ○ Full Field
- ○ No preference
- ○ Point of View

***6. Which of the Viewing Modes do you prefer?**

- ○ 3D
- ○ 2D
- ○ No preference

***7. What would you change about the speed of the system (response)?**

- ○ Increase
- ○ Decrease
- ○ Dynamic
- ○ Don't care

2D Planar Survey

***8. Did you experience any system errors?**

- Yes
- No

Error Reporting

This page focuses on the effects of errors while using the SenCon System.

***9. How many system errors did you experience today? (if unsure please ask the system rep)**

No. of errors:

***10. Could you recover all the errors on your own?**

- Yes
- No

11. How many restarts did you have today?

No. of Restarts

12. In general how easy was the errors to overcome?

	Very Easy	Easy	Moderately Easy	Difficult	Very Difficult	N/A
Ease of overcoming an error:						

Overall Impression Ranking

Relative user preference will be investigated.

***13. Please rank the following in order of preference**

	1st Choice	2nd Choice	3rd Choice	4th Choice
Haptic				
Joystick				

Additional Comments

***14. Any additional comments will be appreciated**

A.2 3D Experiment - Post Completion Survey

3D Box Survey

Personal Information

1. Please enter your personal information:

Name: ______

2. Please supply the following information:

Email Address: ______

Age ______

***3. Please select your gender:**

- ○ Male
- ○ Female

Overall Impressions

When considering the experiments you just completed (2D fixed planar), please answer the following questions.

***4. Which of the Input Modes do you prefer?**

- ○ Haptic Control
- ○ Joystick Control
- ○ No preference

***5. Which of the Viewing Angles do you prefer?**

- ○ Full Field
- ○ No preference
- ○ Point of View

***6. Which of the Viewing Modes do you prefer?**

- ○ No preference
- ○ 3D
- ○ 2D

***7. What would you change about the speed of the system (response)?**

- ○ Increase
- ○ Decrease
- ○ Dynamic
- ○ Don't care

Page 1

3D Box Survey

***8. Did you experience any system errors?**

- Yes
- No

Error Reporting

This page focuses on the effects of errors while using the SenCon System.

***9. How many system errors did you experience today? (if unsure please ask the system rep)**

No. of errors:

***10. Could you recover all the errors on your own?**

- Yes
- No

11. How many restarts did you have today?

No. of Restarts

12. In general how easy was the errors to overcome?

	Very Easy	Easy	Moderately Easy	Difficult	Very Difficult	N/A
Ease of overcoming an error:						

Overall Impression Ranking

Relative user preference will be investigated.

***13. Please rank the following in order of preference**

	1st Choice	2nd Choice	3rd Choice	4th Choice
Haptic				
Joystick				

Additional Comments

***14. Any additional comments will be appreciated**

Appendix B

Sample R-code

B.1 ANOVA

For the 2D-Planar User performance analysis the following structure was used to import and analyse the data:

INPUT:

Table B.1.1: Input Data table for R 2D-Planar analysis

	Joystick		**Haptic**	
ViewAngle	**POV**	**FF**	**POV**	**FF**
2D:	31.20	19.15	30.15	17.44
	18.75	9.79	30.71	25.57
	18.54	14.53	30.25	15.97
	21.93	23.65	46.32	34.31
	39.76	22.95	43.86	20.55
	21.88	22.69	26.49	26.15
	27.54	19.68	21.31	25.35
	30.11	61.94	22.84	24.87
	20.91	17.76	35.73	21.83
	53.92	30.01	19.22	43.90
3D:	14.67	28.19	20.36	21.72
	18.12	15.96	30.60	27.79
	21.16	16.07	18.92	20.73
	19.05	17.90	36.12	40.10
	26.88	32.97	28.88	45.60
	19.24	32.40	26.78	27.27
	22.18	19.31	23.33	28.50
	41.35	26.50	38.73	27.83
	20.08	19.88	36.61	37.44
	34.37	40.27	43.17	47.06

R-Code:

```
>> Y<-scan()
>> Input<-gl(2,2,2*2*2*10,labels=c("Joystick","Haptic"))
>> ViewAngle<-gl(2,1,2*2*2*10,labels=c("POV","FF"))
>> ViewMode<-gl(2,40,2*2*2*10,labels=c("2D","3D"))
>> anova(lm(Y~Input*ViewAngle*ViewMode))
```

The R-code utilises the ***scan*** command this command awaits the data, the data was copied from MS Excel to with out labels. The structure of the *factors* is then assigned using the gl command. **NOTE**: here *ViewAngle* is used instead of *field of view*.

OUTPUT:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Input	1	392.1	392.09	4.0434	0.04809	*
ViewAngle	1	45.7	45.65	0.4708	0.49484	
ViewMode	1	7.5	7.54	0.0778	0.78113	
Input:ViewAngle	1	0	0	0	0.99648	
Input:ViewMode	1	137.5	137.45	1.4175	0.23773	
ViewAngle:ViewMode	1	199.2	199.19	2.0541	0.15612	
Input:ViewAngle:ViewMode	1	3.5	3.51	0.0362	0.84955	
Residuals	72	6981.8	96.97			
—						
Signif. codes:	0	'***' 0.001	'**' 0.01	'*' 0.05	'.' 0.1	' ' 1

- As can be seen here, there is only one significant interaction at the 95% confidence interval.

B.2 Friedman Test with Post-hoc, R-statistics Code

```
friedman.test.with.post.hoc <- function(formu, data, to.print.friedman = T, to.post.hoc.if.signif = T,  to.plot.parallel = T, to.plot.boxplot = T, signif.P = .05, color.blocks.in.cor.plot = T, jitter.Y.in.cor.plot =F)
{
        # formu is a formula of the shape:      Y ~ X | block
        # data is a long data.frame with three columns:    [[ Y (numeric), X (factor), block (factor) ]]

        # Note: This function doesn't handle NA's! In case of NA in Y in one of the blocks, then that entire block should be removed.


        # Loading needed packages
        if(!require(coin))
        {
                print("You are missing the package 'coin', we will now try to install it...")
                install.packages("coin")
                library(coin)
        }

        if(!require(multcomp))
        {
                print("You are missing the package 'multcomp', we will now try to install it...")
```

```
			install.packages("multcomp")
			library(multcomp)
		}

		if(!require(colorspace))
		{
			print("You are missing the package 'colorspace', we will now try to install it...")
			install.packages("colorspace")
			library(colorspace)
		}


		# get the names out of the formula
		formu.names <- all.vars(formu)
		Y.name <- formu.names[1]
		X.name <- formu.names[2]
		block.name <- formu.names[3]

		if(dim(data)[2] >3) data <- data[,c(Y.name,X.name,block.name)]  # In case we have a "data" data frame with more then the three columns we need. This code will clean it from them...

		# Note: the function doesn't handle NA's. In case of NA in one of the block T outcomes, that entire block should be removed.

		# stopping in case there is NA in the Y vector
		if(sum(is.na(data[,Y.name])) > 0) stop("Function stopped: This function doesn't handle NA's. In case of NA in Y in one of the blocks, then that entire block should be removed.")

		# make sure that the number of factors goes with the actual values present in the data:
		data[,X.name ] <- factor(data[,X.name ])
		data[,block.name ] <- factor(data[,block.name ])
		number.of.X.levels <- length(levels(data[,X.name ]))
		if(number.of.X.levels == 2) { warning(paste("'",X.name,"'", "has only two levels. Consider using paired wilcox.test instead of friedman test"))}

		# making the object that will hold the friedman test and the other.
		the.sym.test <- symmetry_test(formu, data = data,	### all pairwise comparisons
								teststat = "max",
```

```
                                        xtrafo = function(Y.data) { trafo( Y.data, factor_trafo = function(x) { model.matrix(~ x - 1) %*% t(contrMat(table(x), "Tukey")) } ) },
                                        ytrafo = function(Y.data){ trafo(Y.data, numeric_trafo = rank, block = data[, block.name] ) }
                                      )
    # if(to.print.friedman) { print(the.sym.test) }


    if(to.post.hoc.if.signif)
    {
        if(pvalue(the.sym.test) < signif.P)
        {
        # the post hoc test
        The.post.hoc.P.values <- pvalue(the.sym.test, method = "single-step")  # this is the post hoc of the friedman test

        # plotting
        if(to.plot.parallel & to.plot.boxplot)  par(mfrow = c(1,2)) # if we are plotting two plots, let's make sure we'll be able to see both

        if(to.plot.parallel)
        {
        X.names <- levels(data[, X.name])
        X.for.plot <- seq_along(X.names)
```

```
plot.xlim <- c(.7 , length(X.for.plot)+.3) # adding some spacing from both sides of the plot

if(color.blocks.in.cor.plot)
{
blocks.col <- rainbow_hcl(length(levels(data[,block.name])))
} else {
blocks.col <- 1 # black
}

data2 <- data
if(jitter.Y.in.cor.plot) {
data2[,Y.name] <- jitter(data2[,Y.name])
par.cor.plot.text <- "Parallel coordinates plot (with Jitter)"
} else {
par.cor.plot.text <- "Parallel coordinates plot"
}

# adding a Parallel coordinates plot
matplot(as.matrix(reshape(data2,  idvar=X.name, timevar=block.name,
direction="wide")[,-1])   ,
type = "l",  lty = 1, axes = FALSE, ylab = Y.name,
xlim = plot.xlim,
col = blocks.col,
main = par.cor.plot.text)
axis(1, at = X.for.plot , labels = X.names) # plot X axis
axis(2) # plot Y axis
points(tapply(data[,Y.name], data[,X.name], median) ~ X.for.plot, col = "red",pch = 4, cex = 2, lwd = 5)
}

if(to.plot.boxplot)
{
# first we create a function to create a new Y, by substracting different combinations of X levels from each other.
subtract.a.from.b <- function(a.b , the.data)
{
        the.data[,a.b[2]] - the.data[,a.b[1]]
}
temp.wide <- reshape(data,  idvar=X.name, timevar=block.name, direction="wide")
```

```
                        #[,-1]
                wide.data <- as.matrix(t(temp.wide[,-1]))
                colnames(wide.data) <- temp.wide[,1]
                Y.b.minus.a.combos <- apply(with(data,combn(levels(data[,X.name]), 2)), 2, subtract.a.from.b, the.data =wide.data)
                names.b.minus.a.combos <- apply(with(data,combn(levels(data[,X.name]), 2)), 2, function(a.b) {paste(a.b[2],a.b[1],sep=" - ")})
                the.ylim <- range(Y.b.minus.a.combos)
                the.ylim[2] <- the.ylim[2] + max(sd(Y.b.minus.a.combos))         # adding some space for the labels
                is.signif.color <- ifelse(The.post.hoc.P.values < .05 , "green", "grey")
                boxplot(Y.b.minus.a.combos,
                names = names.b.minus.a.combos ,
                col = is.signif.color,
                main = "Boxplots (of the differences)",
                ylim = the.ylim
                )
                legend("topright", legend = paste(names.b.minus.a.combos, rep(" ; PostHoc P.value:", number.of.X.levels),round(The.post.hoc.P.values,5)) , fill =  is.signif.color )
                abline(h = 0, col = "blue")
                }
                list.to.return <- list(Friedman.Test = the.sym.test, PostHoc.Test = The.post.hoc.P.values)
                if(to.print.friedman) {print(list.to.return)}
                return(list.to.return)
        }       else {
                print("The results where not significant, There is no need for a post hoc test")
                return(the.sym.test)
                }
        }
# Original credit (for linking online, to the package that performs the post hoc test) goes to "David Winsemius", see:
# http://tolstoy.newcastle.edu.au/R/e8/help/09/10/1416.html
}
```

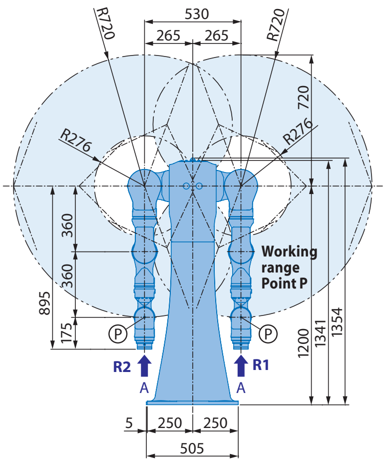
Appendix C

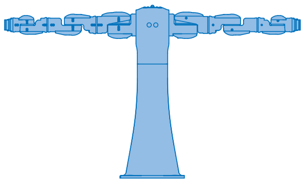
Robot, controller specification

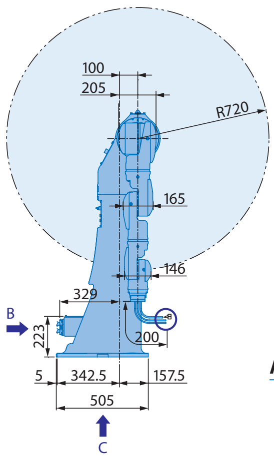
C.1 Robot

Flexible Applications with the SDA-series
The SDA-series are slim an agile 15-axis dual-arm robots providing „human-like" flexibility of movement and fast acceleration. They distinguish themselves through slim and lightweight design. Superior dexterity and best-in-class wrist characteristics make slim, dual-arm robots ideally suited for assembly, part transfer, machine tending, packaging and other handling tasks, that formerly could only be done by people.
These all-round robots can be driven by the high performance DX100 controller as well as the FS100 compact controller.
YASKAWA
• 15 axes
• Flexible applications
• Compact design allows maximum performance
MOTOMAN SDA-series
SDA5F
SDA10D
SDA10F
SDA20D
SDA20F
YASKAWA

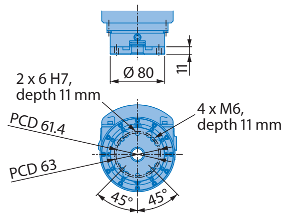
MOTOMAN SDA10D, SDA10F Technical Data



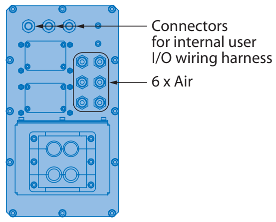




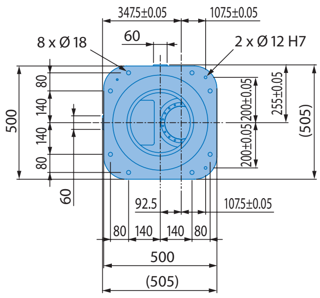
View A



View B

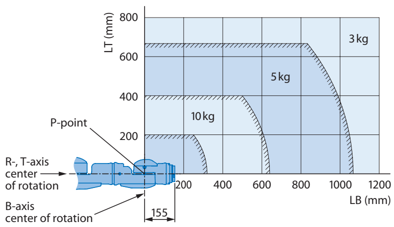


View C



Mounting options: Floor

Allowable wrist load



EN ISO 9001
Zertifikat Nr. 20 100 9864
www.tuv.at

All dimensions in mm | Technical data may be subject to change without previous notice | Please request detailed drawings at robotics@yaskawa.eu.com –YR-SDA010D-A11 IP54/65/67, YR-SDA010F-A00, YR-SDA010F-A11 IP54/65/67, B-06-2013, A-Nr. 157200

Specifications SDA10D, SDA10F

Axes	Maximum motion range [°]	Maximum speed [°/sec.]	Allowable moment [Nm]	Allowable moment of inertia [kg · m²]
Rotation	±170	130	–	–
S	±180	170	–	–
L	±110	170	–	–
θ	±170	170	–	–
U	±135	170	–	–
R	±180	200	31.4	1.0
B	±110	200	31.4	1.0
T	±180	400	19.6	0.4

Controlled axes	15
Max. payload [kg/arm]	10
Repeat. pos. accuracy [mm]	±0.1
Max. reach [mm]	R = 720/720
Temperature [°C]	0 to +40
Humidity [%]	20 – 80
Weight [kg]	220
Power supply, average [KVA]	2.5 SDA10D 1.5 SDA10F

C.2 XRC Controller

Technical data

Controller

Configuration	Free-standing, Enclosed type
Dimensions	800(W)x900(H)x650(D) mm
Weight	Approx. 170 kg
Cooling system	Indirect cooling
Ambient temperature	During operation 0° to +45° During transport -10° to +60°
Relative humidity	Max. 90% (non-condensing)
Power supply	3 x 400/415/440V AC, 50/60 Hz
Grounding	Less than 100 ohm
Digital I/O	Specialised signal (hardware) 12 inputs and 3 outputs General signals (standard) 40 inputs and 40 outputs 4 direct inputs
Positioning system	Absolute encoder / Serial interface
Drive units	Servopacks for AC servomotor
Accel / Decel	Software servo control
Programming capacity (standard)	5.000 steps and 3.000 instructions 1.500 ladder steps

Safety features

Personal safety	3-position "dead-man´s handle" Low speed in teaching mode
Teach Lock Mode	Prohibits operation from operator´s panel
Collision proof frames	Doughnut-sector frame, cubic frame
Machine lock	Test-run peripherial device without robot motion
Self-diagnosis	Alarm and error messages displayed together with explanation
User Alarm display	Possible to display alarm messages for peripheral devices

Operator´s panel

Buttons provided	Mode, Start, Hold, Emergency stop Servo power ON

Programming pendant

Material	Reinforced thermoplastic enclosure
Dimensions	211(W)x382(H)x71(D) mm
Weight	1.2 kg
Display	5.7 inch, 40 characters x 12 lines
Safety feature	3-position "dead-man´s handle"
Interface	RS-232C

Programming functions

Coordinate system	Joint, rectangular/cylindrical, tool, user coordinates
Robot Motion Control	Joint coordinates, linear/circular, interpolation, tool coordinates
Speed setting	Percentage for joint coordinates, 0.1 mm/s units for interpolations, angular velocity for T.C.P. fixed motion
Program Control Instructions	Jump, call, timer, robot stop, execution of some instructions during robot motion
Modification of teaching point	Adding, deleting, correcting (robot axes and/or external axes)
Position control	Manually forwards and backwards in the job (even circular)
Speed adjustment	Fine adjustment possible
I/O-function	Discrete I/O control, pattern I/O processing
Programming	Interactive programming
Language	Robot language: INFORM II
Display text	English, Swedish, French, Spanish, Italian, Finnish, German
Tool Centre Point	Max. 24 and up to 24 external TCP´s
TCP-calibration	Automatically calibrates parameters for end effectors using master jig

Maintenance functions

Software time usage meters	Control power-ON time, servo power-ON time, playback time, work time and operation time displayed
Alarm display	Alarm messages and previous alarm records
I/O-diagnosis	Simulated enable/disabled output possible

Options

Digital I/O	I/O-boards, total max. 256/256 MIO02: 32 inputs and outputs MIO03: 16 inputs and outputs
Analog output	12 channel (MEW/XEW-board)
Memory expansion (up to...)	Max 60.000 points for 6 axes and 20.000 instructions 3.000 ladder steps
External axis	Total max. 27 axes
Enclosure classification	IP54, by add on kit

MOTOMAN®

XRC Robot Controller

Software functions

Example of general functions

- ARM control
- Vibration control
- Station coordinated motion control
- P-start function
- Coordinated motion 2 manipulators
- Coordinated motion 3 manipulators
- Coordinated motion 3 manipulators and ex-axis
- Twin drive function
- Welding condition slope up/down
- Interrupt job function
- Search function
- Servo float function
- Linear servo float function
- T-axis endless rotation
- External axis endless rotation
- External reference point control function
- PMT function
- Start point search function
- High speed start point search function
- General purpose sensor function
- Data tansmission function
- Relative job function
- Parallel shift function
- PAM function
- TCP function
- Weaving function
- Pause weaving function
- Online tool modification function
- Independant control function (6 tasks)
- External storage function PC-Card
- Analogue output function
- Analogue output function related to speed

Examples of special functions

- COMARC arc welding control
- Multi-layer welding function
- Conveyor synchronous function
- Press synchronous function
- Laser sensor function
- Pitch control for spot welding
- Ethernet communication
- Fieldbus communication

Available optional boards

- 16/16 IO-board
- 32/32 IO-board
- MEW-Welding interface board
- XEW-Welding interface board
- Ethernet communication board
- Fieldbus communication board
- Sensor board

Software applications

- Arc welding application
- Handling application
- Spot welding application
- General purpose application

MOTOMAN ROBOTICS EUROPE AB
Box 504 • SE-385 25 Torsås • Sweden
Tel: +46-486-48800 • Fax: +46-486-41410
www.motoman.se

Reg 7P-01-04-99GB

Group Companies: **France:** MOTOMAN Robotics, Nantes +33-2-40131919, **Germany:** MOTOMAN Robotec, Allershausen +49-8166-90-0 • MOTOMAN Robotec, Frankfurt +49-6173-6077-30, **Great Britain:** MOTOMAN Robotics, Banbury +44-1295-272755, **Italy:** MOTOMAN Robotics, Modena +39-59280496, **Netherlands:** MOTOMAN Benelux, Breda +31-76-5424278, **Slovenia:** RISTRO, Ribnica +386-61-861113, **Spain:** MOTOMAN Robotics, Barcelona +34-3-6303478, **Sweden:** MOTOMAN Robotics Europe, Torsås +46-486-48800 • MOTOMAN Mecatron Robotic Systems, Kalmar +46-480-444600

Distributors: **Czech Republic:** MGM Spol, Tabor +420-361-254571, **Denmark:** HN Automatic, Vejle +45-79428000, **Finland:** MOTOMAN-Robotit, Turku +358-22145600, **Greece:** Kouvalias Industrial Robots, Kallithea +30-1-95892436, **Norway:** Robot-Teknikk +47-32217830, **Portugal:** Electro-Arco, Amadora +351-14968160, **Switzerland:** Geiger Handling, Schwarzenburg +41-31-7343111 • Messer SAG, Dällikon +41-18471717

MOTOMAN XRC Robot Controller

MOTOMAN XRC will be the premier robot controller.
It is physically minimised but optimised for functions, simple operations and expandability.

Outlines of the XRC

Built-in transformer
Simple installation and setup by plug-connections.
Used for robot in UP-series and SK-X, SP-X and SV-X-series.
INFORM programming language
Expandable slots for optional boards
-Sensor function by MSL-board
-Welding interface MEW and XEW-board
-I/O-interface MIO-board
-Ethernet and Fieldbus board
XRC cabinets may be put on top of each other or docked side by side
Fulfils the EC-directive incl. EMC and LVD
Made in Sweden

Advanced control system

Powerful 32-bit micro processor for rapid data processing.
PLC-programs for different applications, for example: arc welding and handling.
Advanced PLC-unit in the control system with expanded program capacity and new logical functions e.g. for controlling pneumatic functions in fixtures.
Multi tasking makes it possible to run several jobs simultaneously.
Digital servopacks with software controlled speed and position feedback.

Shock detection function

This function stops the robot when it comes into contact with an object. Robot or peripherical devices can be protected in case of accidental collision.

Maintenance functions

Remote maintenance function monitors remotely robot status through ethernet network. Production processing monitoring and quality control can be managed from a central control room.
Maintains history of when and who changed robot job instructions.

Programming pendant

All programming functions are in one unit with clear and large 12-line screen. Optimal programming time is achieved while all functions are available during programming. The menu text can be switched between several different European languages by a simple key operation. The programming pendant is equipped with a 3-position "dead-man´s handle".

Communication

XRC is like the previous control system for MOTOMAN robots a data technical solution that is PC-compatible. Therefore common computer tools may be used for creating, reading and editing jobs in a PC.
The flexibility of the control system makes it possible to communicate with other systems such as PLC´s, host computers, vision system, etc. through serial links, Ethernet, digital I/O, fieldbus. Controller memory is also accessable through PC-card.

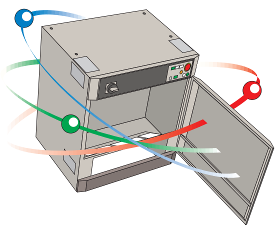
Path control

Robot working path is controlled to perform consistently precision and quality. Even at high speed, the robot will not vary from the target execution line.
XRC improves high speed welding, cutting and sealing more than ever.

ARM control

Advanced Robot Motion Control improves quality and productivity by improved servo response for the motion control. Optimised acceleration/deceleration and vibration control obtains correct path and cycle time. Speed is automatically optimised for circles and corners. Collision between robot, robot tool and other devices is immediately detected to stop robot and protect from damages.

Controller features



LCD-display
Large LCD-display with back-lighting. High contrast for day-light operation.

Cursor operation
Cross shape cursor button for instruction basic operations. Operates by icons and pull-down menues.

Lightweight
Lightweight thermoplastic p-pendant reduces fatigue.

Key operation
Key allocation is based on human engineering research. The number of keys are minimised for required functions only.

3 Robots
CPU can control up to 3 robots simultaneously.

27 Axes
CPU can control up to 27 axes simultaneously.

6 Tasks
Multi-task CPU can handle up to 6 tasks simultaneously.

Easy placing
Cables and air vent only at back. Optional cables, IO's, etc. are accessed behinde the front door.

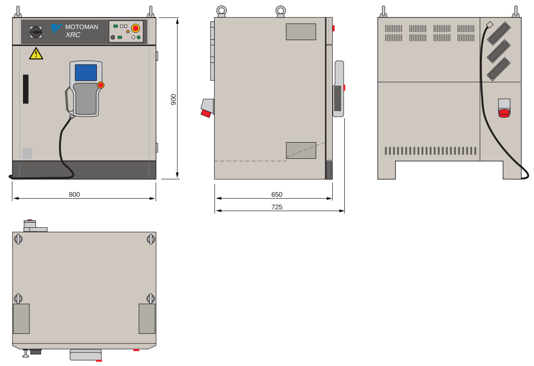
Easy installation
Cables for power, robots and IO´s are connected by plugs.

Expansion units
Optional units such as external axis servopacks may be placed in optional cabinets. These cabinets are docked to the controller cabinet by means of ports on top and on the sides. Which means tidy cable installation.

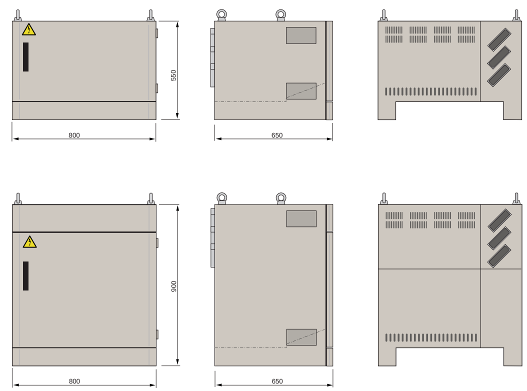
PC-Card
PC-Card (PCMCIA) is used for expansion interface. It simplifies expanding functions and speeds up backing up data.

Network
Standard field networks are applicable: Ethernet, Profibus, Interbus-S, Device-NET, etc. Your manufacturing line can be integrally controlled through your existing network.

Controller layout



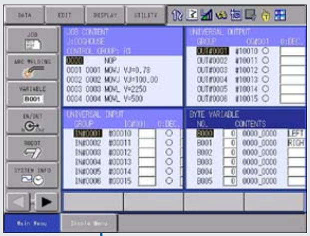
External axis cabinet layout



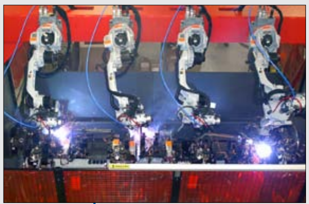
C.3 DX 100 Controller

SOLUTIONS IN MOTION®

YASKAWA
MOTOMAN ROBOTICS



MULTIPLE WINDOW DISPLAY



MULTIPLE ROBOT CONTROL

DXM100 CONTROLLER

DX100

ROBOT/SYSTEM CONTROLLER

Option:
DXM100 Controller (smaller cabinet)

KEY FEATURES

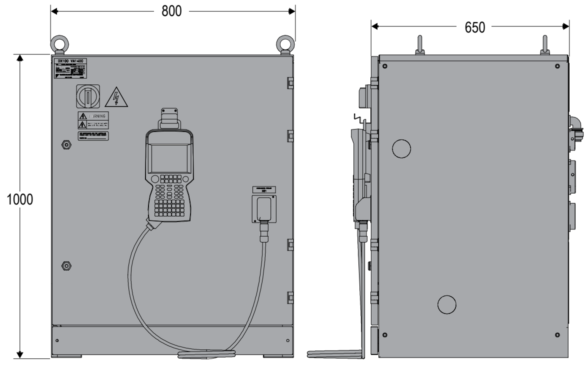
- Patented multiple robot control (up to 8 robots/72 axes)
- Faster processing, high performance
- Integrated cell (system-level) control capabilities
- Open communication
- Energy savings
- Compliant to safety standards
- Controller connections through back of unit optimize floorspace

Dynamic Next-Generation Controller

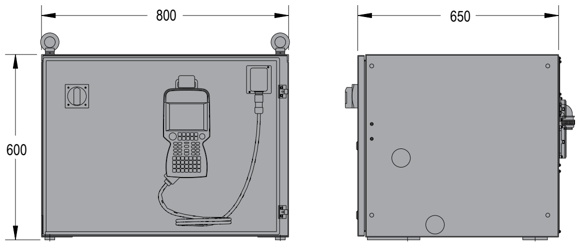
- Features robust PC architecture. Provides system-level control for robotic workcells.
- Patented multiple robot control (up to 8 robots/72 axes), as well as I/O devices and communication protocols. Dynamic interference zones protect robot arm and provide advanced collision avoidance.
- Fast processing speed provides smooth interpolation.
- Advanced Robot Motion (ARM) control provides high performance. Best-in-class path planning dramatically reduces teaching time.
- Small, lightweight Windows® CE programming pendant features color touch screen with multiple window display capability. Unique cross-shaped navigation cursor reduces teaching time. All operator controls are located on pendant. Program file names can be up to 32 characters long.
- Convenient compact flash slot and USB port facilitate memory backups.
- Conserves power during robot idle time, providing up to 25% energy savings.
- Highly flexible fieldbus support. Easy connection to information infrastructure through standard network options.
- Compliant to ANSI/RIA R15.06-1999 and other relevant ISO and CSA safety standards. Includes dual-channel E-Stop functionality, integrated speed monitoring and manual brake release for robot. Optional Category 3 functional safety unit.
- Often eliminates need for separate PLC and human machine interface (HMI). Delivers significant cost savings at system level, while decreasing workcell complexity and improving overall reliability.
- Connections to controller cabinet are made through the back of the unit, optimizing floorspace.
- DX100 control cabinet allows for up to three external axes and can be remote-mounted. DXM100 supports up to two external axes. Top- or side-mount expansion options available for DX100 controller only.
- Easy maintenance with reducer status check function, enhanced troubleshooting and alarm recovery, and 20% improvement in MTTR.

DX100 ROBOT CONTROLLER

DX100 Controller



DXM100 Controller



All dimensions are metric (mm) and for reference only.
Please request detail drawings for all design/engineering requirements.

Standard I/O - NPN

Forty optically isolated inputs, 32 transistor outputs, 8 relay contact outputs (configured to optimize each application), and four break-out cards are provided as standard. For arc welding applications, one YEW01 welder interface board is installed in the DX100 cabinet as standard (not available in DXM100).

I/O Expansion - DX100

The DX100 supports I/O expansion via:

- EtherNet/IP
- DeviceNet
- Profibus-DP
- Mechatrolink II
- CC-Link
- Remote I/O
- Discrete I/O, NPN or PNP
- Analog I/O
- Other networks available

I/O Expansion - DXM100

The DXM100 supports I/O expansion via:

- EtherNet/IP
- DeviceNet
- Remote I/O
- CC-Link

DX100 ROBOT CONTROLLER SPECIFICATIONS

CONTROLLER	**Dimensions**	DX100: 800 (w) x 1000 (h) x 650 (d) (31.5" x 39.4" x 25.6") DXM100: 800 (w) x 600 (h) x 650 (d) (31.5" x 23.6" x 25.6")
	Approximate Mass	150-250 kg (330.8-551.3 lbs.)
	Cooling System	Indirect cooling
	Ambient Temperature	During operation: 0° to 45° C (32° to 113° F) During transport and storage: -10° to 60° C (14° to 140° F)
	Relative Humidity	90% max. non-condensing
	Primary Power Requirements	3-phase, 240/480/575 VAC at 50/60 Hz
	Digital I/O NPN - Standard PNP - Optional	Standard I/O: 40 inputs/40 outputs consisting of 16 system inputs/16 system outputs, 24 user inputs/24 user outputs 32 Transistor Outputs; 8 Relay Outputs Max. I/O (optional): 2,048 inputs and 2,048 outputs
	Position Feedback	Absolute encoder
	Program Memory	JOB: 200,000 steps, 10,000 instructions CIO Ladder Standard: 15,000 steps Expanded: 20,000 steps
	Interface	Ethernet, RS-232C
	Multiple Robot Control	Ability to control up to 8 robots/72 axes
SAFETY FEATURES	**Safety Specs**	Controller Dual-Channel Emergency Stop & Safety Gate user interface. Programming Pendant includes: Dual-channel Emergency Stop Pushbutton, 3-Position Enable Switch with key-lock and Manual Brake Release built into programming pendant. Meets ANSI/RIA R15.06-1999, ANSI/RIA/ISO 10218-1-2007 and CSA Z434-03
	Collision Avoidance	Collision avoidance zones and radial interference zones
	Collision Detection	Protects robot by monitoring torque levels on manipulator
	Machine Lock	Permits testing of peripheral devices without robot operation
	Safety Interlock	Prevents robot operation while safety circuit is open
PENDANT	**Pendant Dimensions**	169 (w) x 314.5 (h) x 50 (d) (6.6" x 12.4" x 2")
	Pendant Display	5.7-inch full-color touch screen, 640 x 480 (VGA)
	Pendant Languages	English, German, Japanese, Spanish, Chinese
	Pendant Weight	.998 kg (2.2 lbs)
	Coordinate System	Joint, rectangular, cylindrical, tool, 24 user-coordinate frames
	Windows® Menu-Driven	User-selectable touch-screen menu, Multiple windows supported
	Interface	One Compact Flash slot; One USB port (1.1)
	Pendant O/S	Windows® CE
	Protection Rating	IP65
PROGRAMMING	**Programming Language**	INFORM III, menu-driven programming
	Robot Motion Control	Joint motion, linear, circular, spline interpolation
	Speed Adjustment	Percentage of maximum for joint motion; mm/sec, cm/min, in/min for displacement; °/sec for orientation
	Device Instructions	Application-specific (ARCON, ARCOFF, LASERON, LASEROFF, HANDON, HANDOFF)
	I/O Instructions	Discrete I/O, 4-bit and 8-bit manipulation, analog output, analog input, analog scaling, sloping
	Operation	Up to 5 levels of undo/redo
MAINTENANCE	**Maintenance Functions**	System monitor, internal maintenance clocks
	Self-Diagnostics	Classifies errors and major/minor alarms and displays data
	User Alarm Display	Displays alarm messages for peripheral devices
	Alarm Display	Alarm messages and alarm history
	I/O Diagnosis	Permits simulated enabled/disabled input/output
	TCP Calibration	Automatically calibrates parameters for end-effectors, optional TCP recovery function
	Tool Weight Calibration	Automatically calibrates total weight of tool, center of gravity and inertia for peak performance

www.motoman.com

YASKAWA
MOTOMAN ROBOTICS

MOTOMAN ROBOTICS
100 AUTOMATION WAY, MIAMISBURG, OHIO 45342
TEL: 937.847.6200 ■ FAX: 937.847.6277

Appendix D

DX 100 Communication Instruction set

YASKAWA

DX100 OPTIONS
INSTRUCTIONS

FOR DATA TRANSMISSION FUNCTION

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

]MOTOMAN INSTRUCTIONS

MOTOMAN--□□□ INSTRUCTIONS
DX100 INSTRUCTIONS
DX100 OPERATOR'S MANUAL
DX100 MAINTENANCE MANUAL

The DX100 Operator's Manual above corresponds to specific usage.
Be sure to use the appropriate manual.

YASKAWA

MANUAL NO. RE-CKI-A456

D.1 Data Transmission

1 Outline

The data transmission function is for communication with a host computer such as a personal computer in BSC complying protocol.

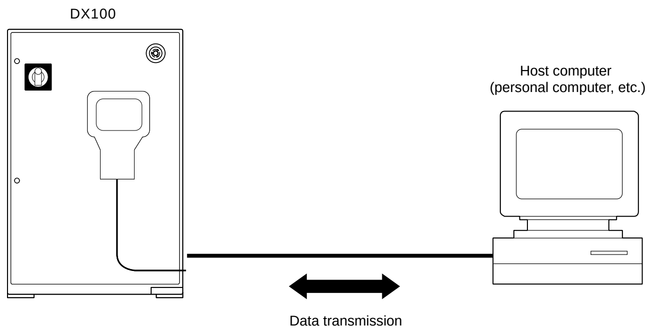
The data transmission function adopts a serial transmission line and standard protocol, making easy connection to a host computer.

The data transmission function is not only for transmission of job but also for controlling robot system by a host computer using a set of commands.

The robot commands in the ASCII code command format are easy to use and helpful for a quick development of necessary software to be run on the host computer.

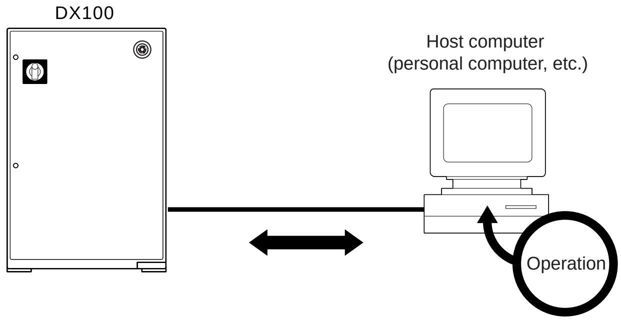
The data transmission function is divided into the following three functions.

- DCI (Data Communication by Instruction)
- Stand-alone function
- Host control function



1.3 Host Control Function

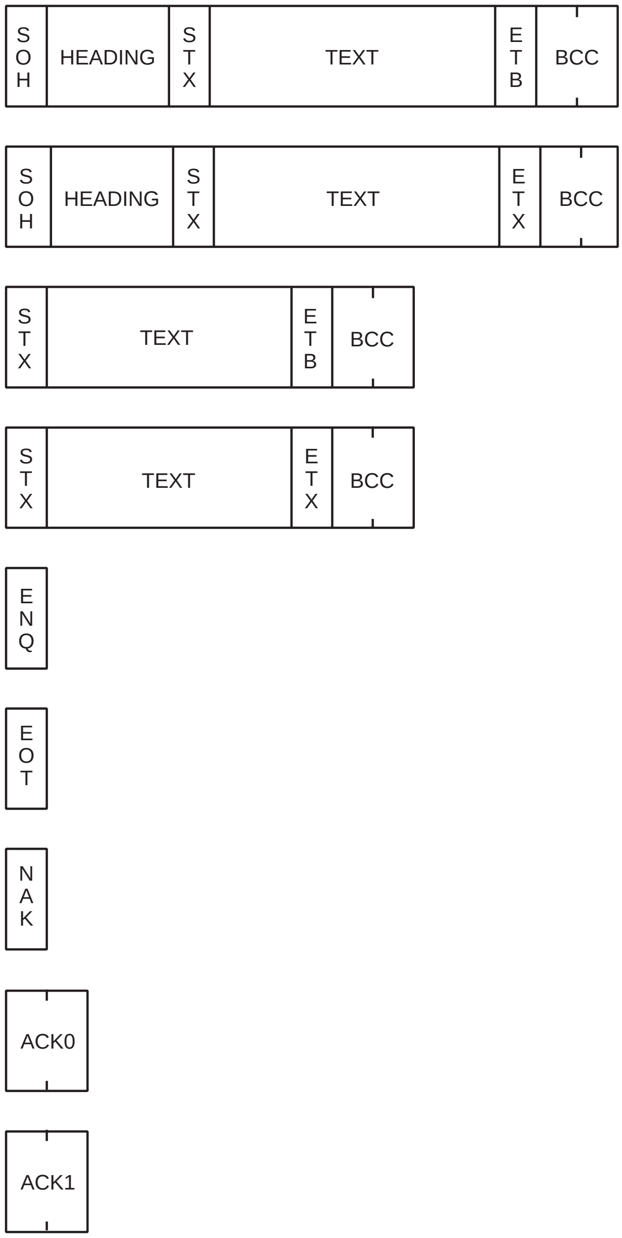
The host control function is for loading and saving jobs, reading robot status, and controlling the system by sending a command from a host computer.



D.2 Data Transmission Format

2.4.3 Transmission Format

The transmission format is as follows.



D.3 Data Transmission Procedure

5.1.1 Transmission Procedure

5.1.1.1 Load

The transmission from a host computer to the DX100 proceeds as follows.

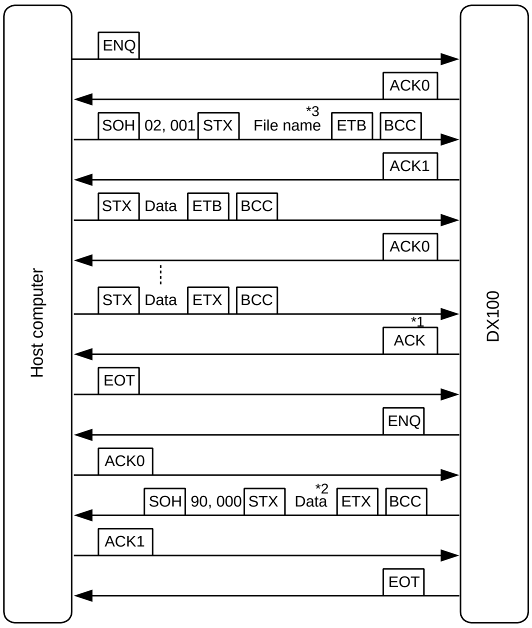
Host computer → DX100

1. The ENQ code is sent from the host computer to establish a data link.
2. After the data link is established, the data is sent from the host computer.
3. After the transmission is completed, the host computer should get ready to receive.
4. After the data link is established, a response to the data sent from the host computer is returned from the DX100 to terminate the transmission.

The data type is distinguished by the header number and the subcode number.

Refer to the header number list.

Fig. 5-1: Loading File Data (Host Control Function)



***1** ACK0 or ACK1
***2** Normal completion : 0000CR (ASCII code)
Abnormal completion : “Integer except 0000”CR (ASCII code)
***3** File name : CR (File name does not include extension)

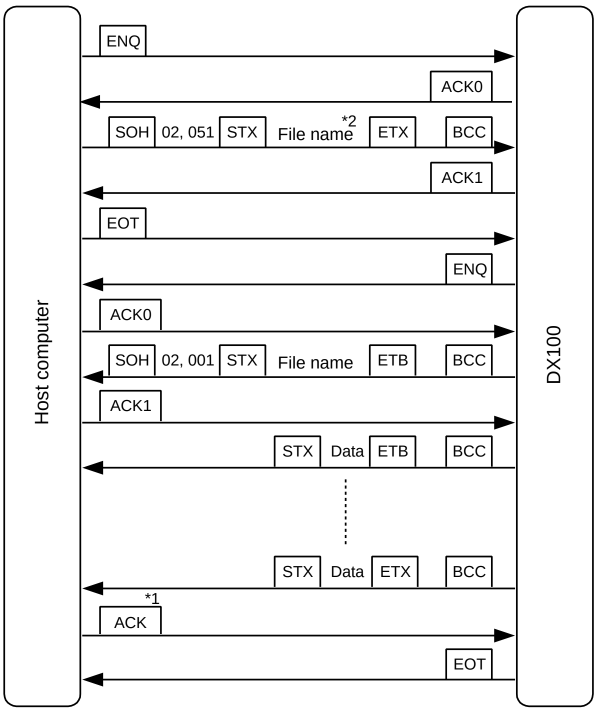
5.1.1.2 Save

The transmission from the DX100 to a host computer proceeds as follows.

DX100 → Host computer

1. The ENQ is sent from the host computer to establish a data link.
2. After the data link is established, a request to send is sent from the host computer.
3. The request to send consists of a header number and a subcode number. Refer to the header number list.
4. After the request to send is accepted, the host computer should get ready to receive data.
   The DX100 sends the ENQ code to establish a data link.
5. After the data link is established, receive the data sent from the DX100. The transmission terminates at completion of reception.
   If the data requested to send are not found, or the header of the request to send has an error, the DX100 sends the following response message instead of data.
   Check the header and take an appropriate action.
   [SOH] 90,000 [STX] DATA [CR][ETX][BCC]

Fig. 5-2: Saving File Data (Host Control Function)



***1** ACK0 or ACK1
***2** File name : CR (File name does not include extension)

5.1.2 Data Management

The jobs for the DX100 may refer to another job or condition data according to instructions. When saving a single job or condition data to the host computer, the correspondence between job and files should be controlled.

To reduce this labor, the related jobs and condition data can be transmitted in a batch as the related job data.

When specification of “related job data” is made, the master job, the related job, and the related condition data are transmitted sequentially.

The header number and the subcode number indicate that the related job data are added.

Refer to the header number list.

Appendix E

INFORM programming commands - excerpt

YASKAWA

NX100
INFORM MANUAL

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

MOTOMAN INSTRUCTIONS

MOTOMAN-☐☐☐ INSTRUCTIONS
NX100 INSTRUCTIONS
NX100 OPERATOR'S MANUAL
NX100 MAINTENANCE MANUAL

The NX100 operator's manuals above correspond to specific usage.
Be sure to use the appropriate manual.

YASKAWA

MANUAL NO. RE-CKI-A444

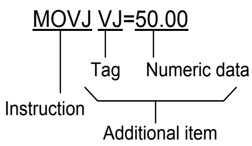
E.1 INFORM Basics

1 INFORM Manual Outline

1.1 About INFORM

1.1.1 With INFORM II

The robot programming language used with NX100 is called INFORM II. INFORM II is composed of the instruction and the additional item (tag and numeric data).



- Instruction : It is used to execute the operation and processing. In the case of a move instruction, when a position is taught, the move instruction is automatically displayed according to the interpolation method.
- Additional item : The speed, time, etc. are set according to the type of instruction. Numeric data and character data are added to the tag that specifies the condition as necessary.

1.1.2 Type of Instruction

The instruction is divided into several types in terms of each process and operation.

Type	Content	Instruction Example
I/O Instruction	It is the instruction used to control the I/O.	DOUT, WAIT
Control Instruction	It is the instruction used to control the processing and operation.	JUMP, TIMER
Operating Instruction	It is the instruction by which the variables, etc. are used and operated.	ADD, SET
Move Instruction	It is an instruction concerning the movement and the speed.	MOVJ, REFP
Shift Instruction	It is an instruction used when a present teaching position is shifted.	SFTON, SFTOF
Instruction which adheres to instruction	It is an instruction which adheres to the instruction.	IF, UNTIL
Work Instruction	It is an instruction concerning work, such as arc welding and handling.	ARCON, WVON
Optional Instruction	It is an instruction concerning optional functions. It can only be used when the function is available.	-

E.2 Move Commands

2.4 Move Instruction

MOVJ

Instruction set:

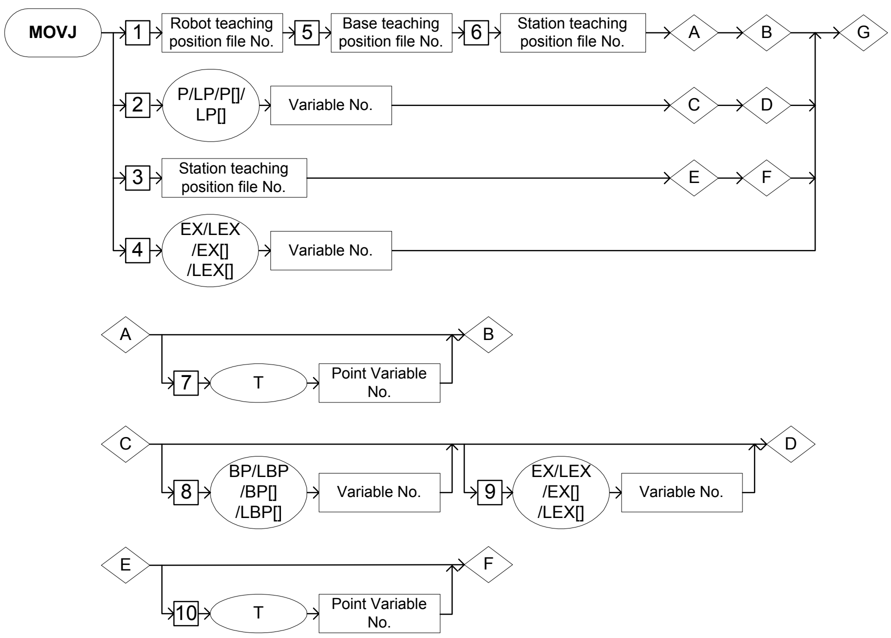
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Moves to the teaching position by joint interpolation.

Construction

The tag which can be used is limited by the type of the job.



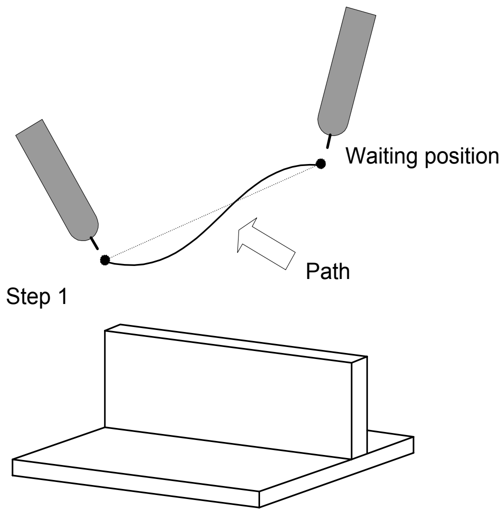
Example

MOVJ P000 VJ=50.00

Move from the manipulator's waiting position to step 1. Move by joint interpolation at a speed of 50%.

The position in Step 1 is registered to the P variable no. 0.

The path during movement is not specified. Be careful of interference.



MOVL

Instruction set:

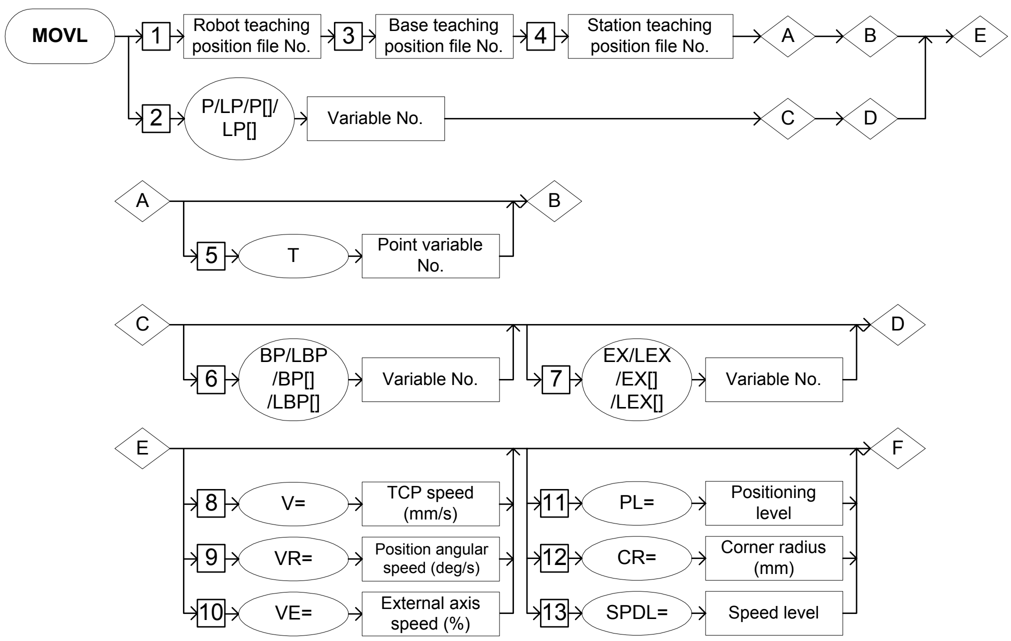
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Moves to the teaching position by linear interpolation.

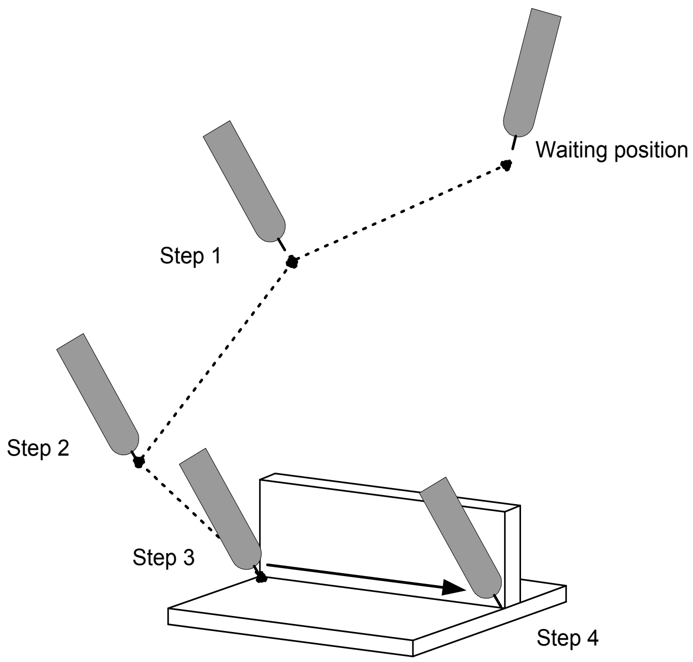
Construction

The tag which can be used is limited by the type of the job.



Example

NOP
MOVJ VJ=50.00
MOVJ VJ=25.00
MOVJ VJ=12.50 • • • Step 3
MOVL V=138 • • • Step 4
Moves from Step 3 to Step 4 by the linear interpolation at a rate of 138cm/min.



MOVC

Instruction set:

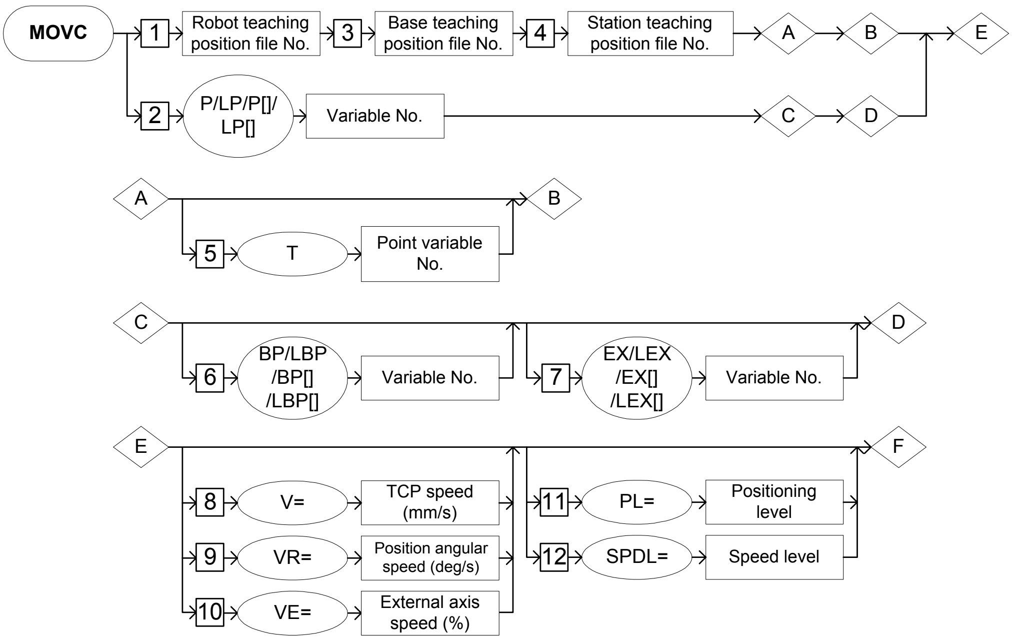
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Moves to the teaching position by circular interpolation.

Construction

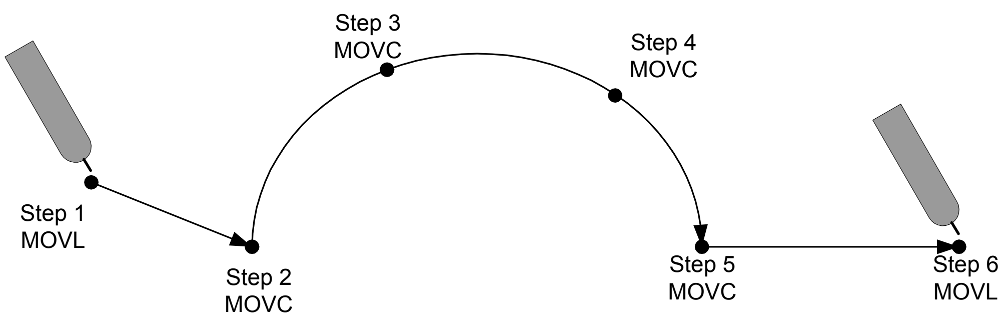
The tag which can be used is limited by the type of the job.



Example

```
NOP
MOVL V=138
MOVC V=138   •  •  • Step 2
MOVC V=138   •  •  • Step 3
MOVC V=138   •  •  • Step 4
MOVC V=138   •  •  • Step 5
MOVL V=138
END
```

Moves from Step 2 to Step 5 by circular interpolation at a rate of 138 cm/min.
Moves to Step 3 in a circular arc formed with the teaching points in Steps 2, 3, and 4.
Moves to Step 4 in a circular arc formed with the teaching points in Steps 3, 4, and 5.
Moves to Step 5 in a circular arc formed with the teaching points in Steps 3, 4, and 5.



MOVS

Instruction set:

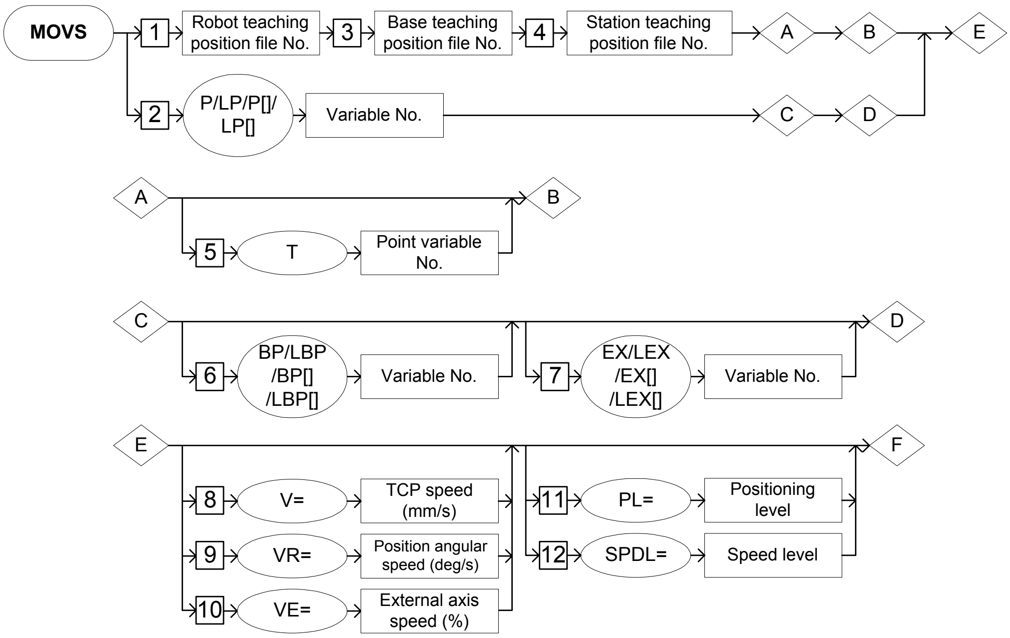
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Moves to the teaching position by spline interpolation.

Construction

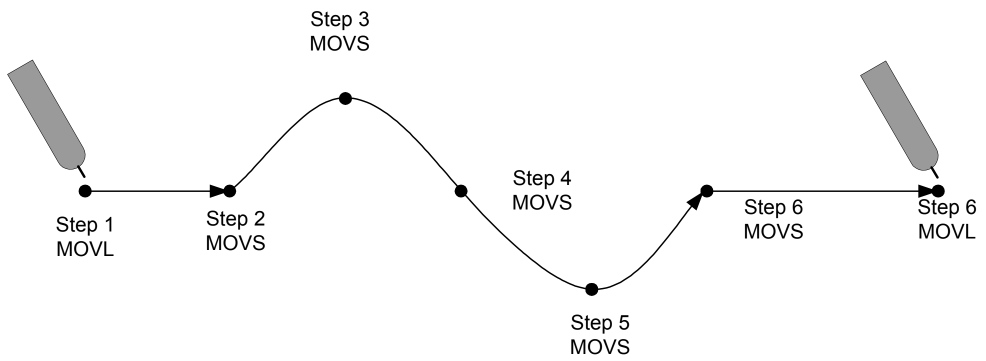
The tag which can be used is limited by the type of the job.



Example

```
NOP
    MOVL V=138
    MOVS V=138  •  •  • Step 2
    MOVS V=138  •  •  • Step 3
    MOVS V=138  •  •  • Step 4
    MOVS V=138  •  •  • Step 5
    MOVS V=138  •  •  • Step 6
    MOVL V=138
    END
```

Moves from Step 2 to Step 6 by spline interpolation at a rate of 138cm/min.
Moves to Step 3 by spline interpolation defined by the teaching points in Steps 2, 3, and 4.
Moves to Step 4 by synchronized spline interpolation defined by the teaching points in Steps 2, 3, 4 and by the synchronized spline interpolation defined by the teaching points in Steps 3, 4, and 5.
Moves to Step 5 by synchronized spline interpolation defined by the teaching points in Steps 3, 4,5 and by synchronized spline interpolation defined by the teaching points in Steps 4, 5, and 6.
Moves to Step 6 by spline interpolation defined by the teaching points in Steps 4, 5, and 6.



IMOV

Instruction set:

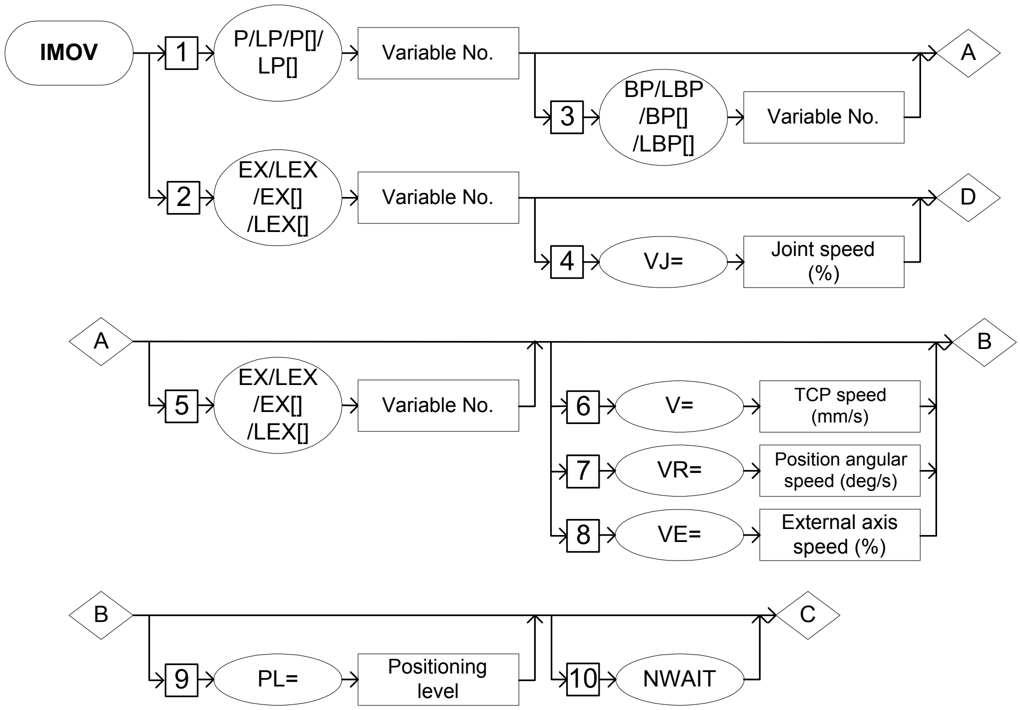
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Moves by linear interpolation from the current position for the specified incremental value.

Construction

The tag which can be used is limited by the type of the job.



10. ACC=Acceleration adjustment ratio

The following tag can be added or omitted.

No	Tag	Explanation	Note
19	ACC=Acceleration adjustment ratio	Specifies the acceleration adjustment ratio. The ACC instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used.

11. DEC=Deceleration adjustment ratio

The following tag can be added or omitted.

No	Tag	Explanation	Note
20	DEC=Deceleration adjustment ratio	Specifies the deceleration adjustment ratio. The DEC instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used.

12. +IMOV

Add the following tag.

No	Tag	Explanation	Note
22	+IMOV	Specifies the move instruction for an incremental value of the master manipulator.	Available only with the optional coordinate function. Refer to the independent/coordinated function for details.

Example

IMOV P000 V=138 RF
Moves from the current position at a rate of 138cm/min for the incremental value specified in P000 in the robot coordinate system.

SPEED

Instruction set:

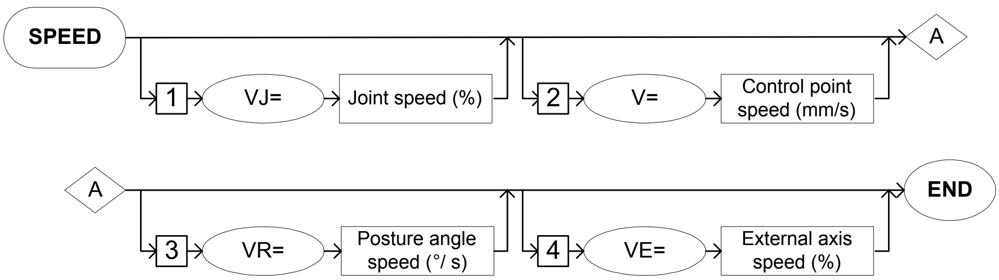
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Sets the playback speed. The manipulator operates at the speed specified in the SPEED instruction when the speed is not specified in the move instruction.

Construction

The tag which can be used is limited by the type of the job.



Job Type and Control Group

No.	Job Type	Control group	Remarks
1	-	One manipulator (standard)	
2	-	One manipulator with station axis	
3	-	Station axis only	

Availability of Each Tag

No	Tag	Control Group			Note
		1	2	3	
1	VJ=	●	●	●	
2	V=	●	●	×	
3	VR=	●	●	×	
4	VE=	×	●	×	

●: Available
×: Not available

Explanation

1. VJ=Joint speed

The following tag can be added or omitted.

No	Tag	Explanation	Note
1	VJ=Joint speed	Specifies the joint speed. The joint speed is shown in the ratio to the highest speed. Operates at the speed decided beforehand when the joint speed is omitted.	Speed: 0.01% to 100.00% Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.01 %)

2. V=Tool center point speed

The following tag can be added or omitted.

No	Tag	Explanation	Note
2	V=Tool center point speed	Specifies the tool center point speed.	Speed: 0.1 mm to 1500.0 mm/s The units can be changed by setting the parameter S2C173. Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.1 mm/s)

3. VR=Position angular speed

The following tag can be added or omitted.

No	Tag	Explanation	Note
3	VR=Position angular speed	Specifies the position angular speed.	Speed: 0.1 degrees to 180.0 degrees/s Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.1 deg/s)

4. VE=External axis speed

The following tag can be added or omitted.

No	Tag	Explanation	Note
4	VE=External axis speed	Specifies the external axis speed.	Speed: 0.01% to 100.00% Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used.(Units: 0.01 %)

Example

```
NOP
MOVJ VJ=100.00                   • • • Moves at the joint speed 100.00%.
MOVL V=138                       • • • Moves at the control point speed 138cm/min.

SPEED VJ=50.00 V=276 VR=30.0     • • • Moves at the joint speed 50.00%.
MOVJ                             • • • Moves at the control point speed 276 cm/min.
MOVL                             • • • Moves at the position angular speed 60.0
                                       degree/s.

 MOVL VR=60.0

END
```

REFP

Instruction set:

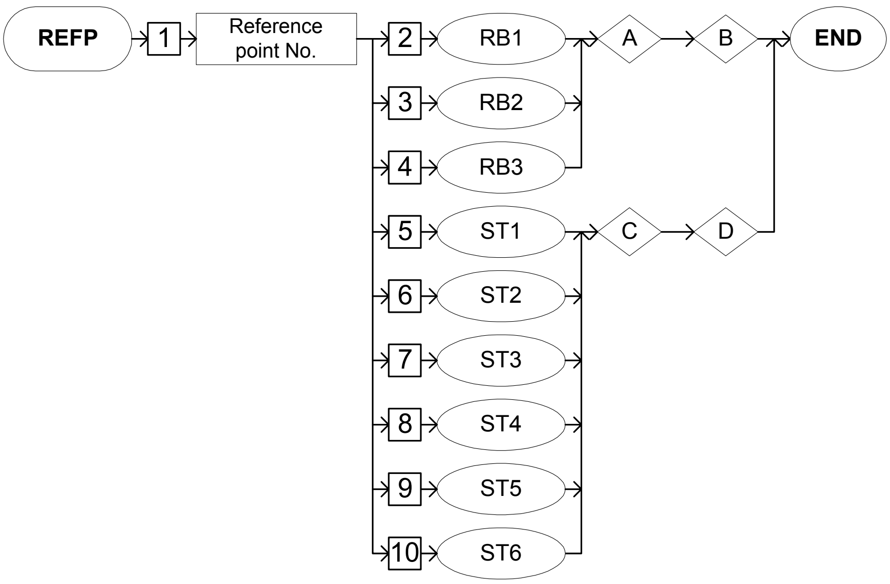
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

It is an instruction which has the position data by which a supplementary point of the wall point, etc. for weaving is set.

Construction

The tag which can be used is limited by the type of the job.



Appendix F

Systems Engineering Development Models

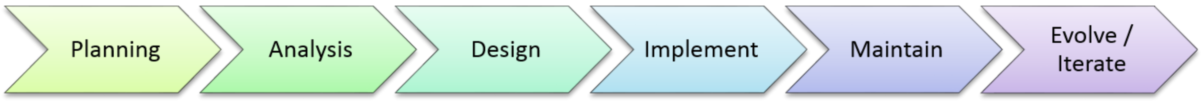


Figure F.0.1: Linear Model

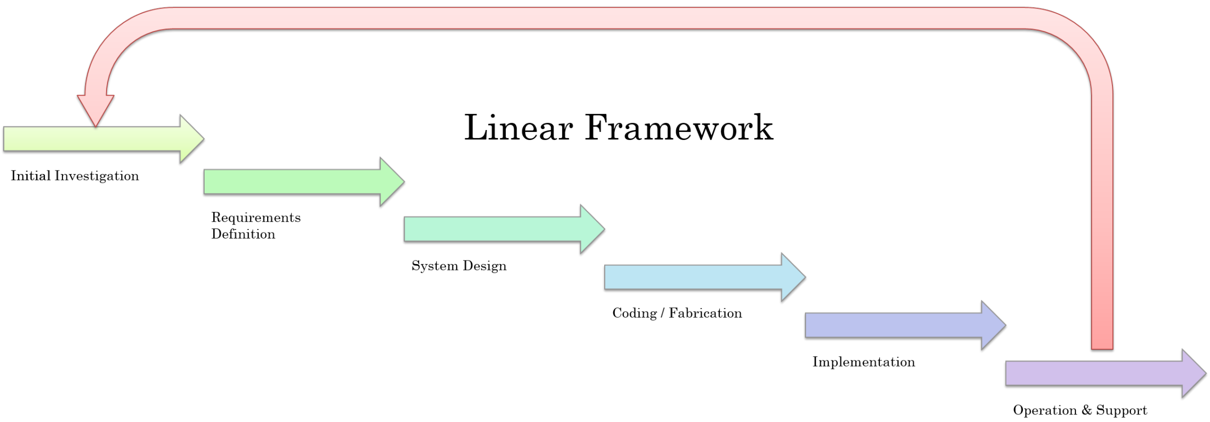


Figure F.0.2: Linear Model - Waterfall Model

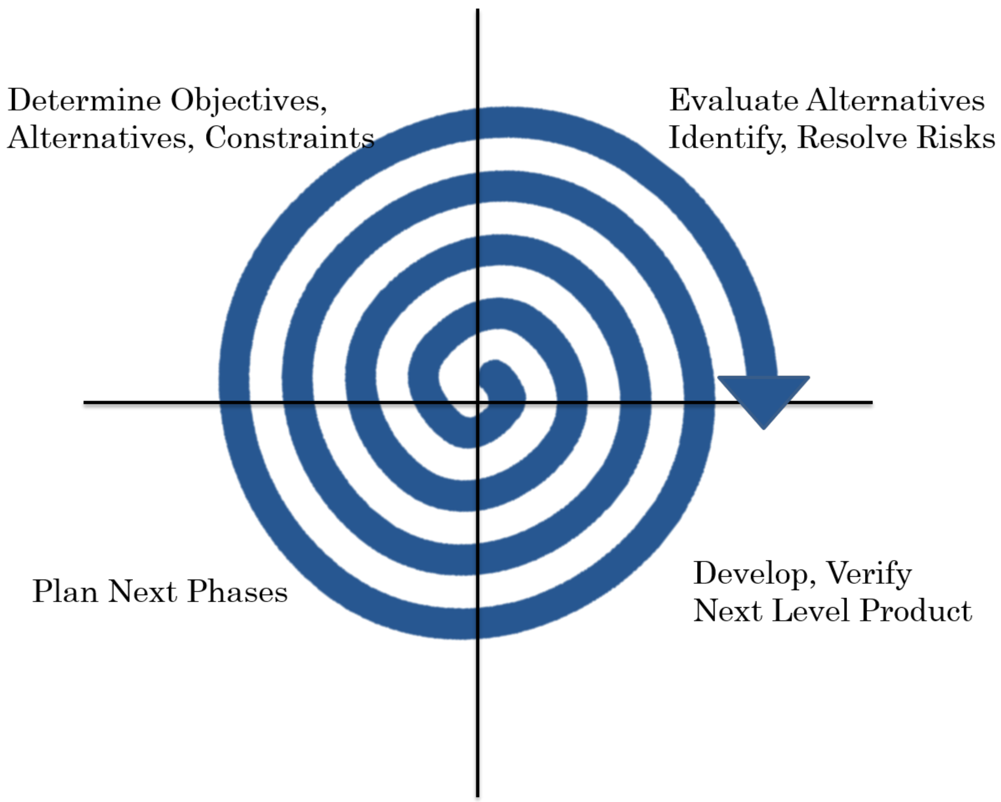


Figure F.0.3: Spiral Model

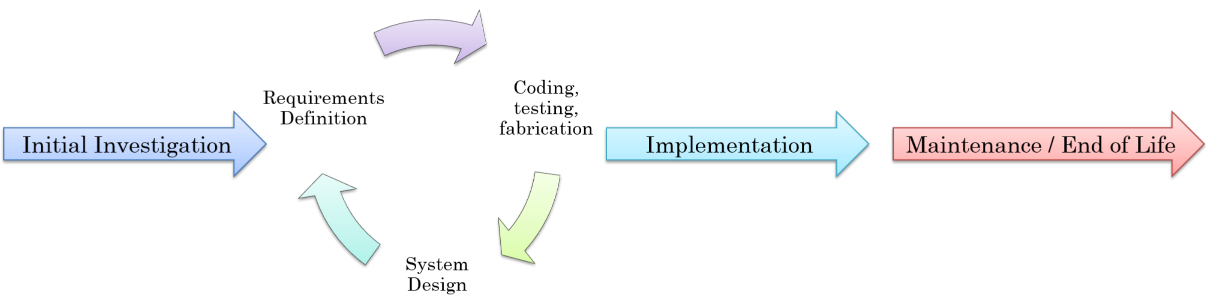


Figure F.0.4: Prototyping, Iterative Model

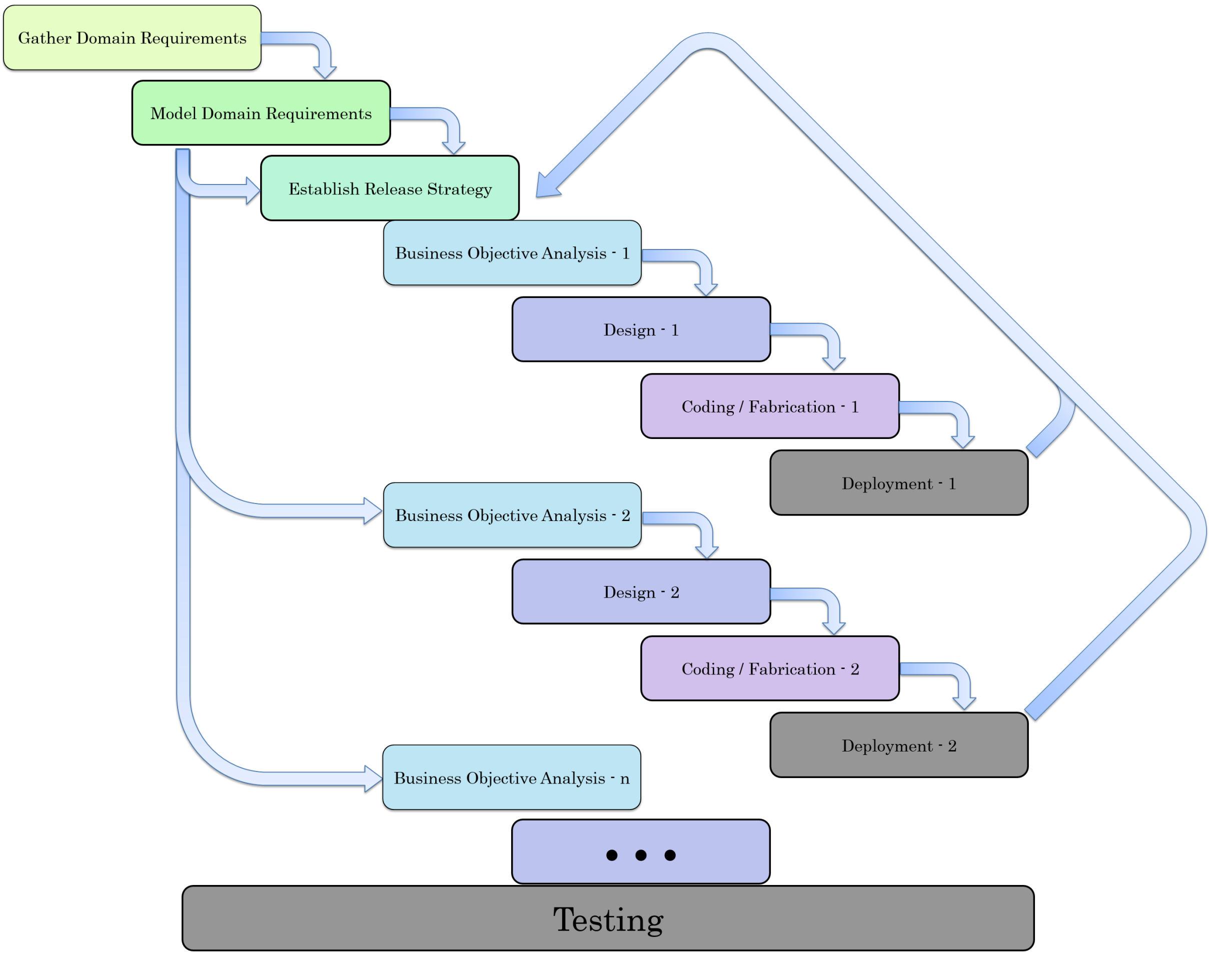


Figure F.0.5: Concurrent Model

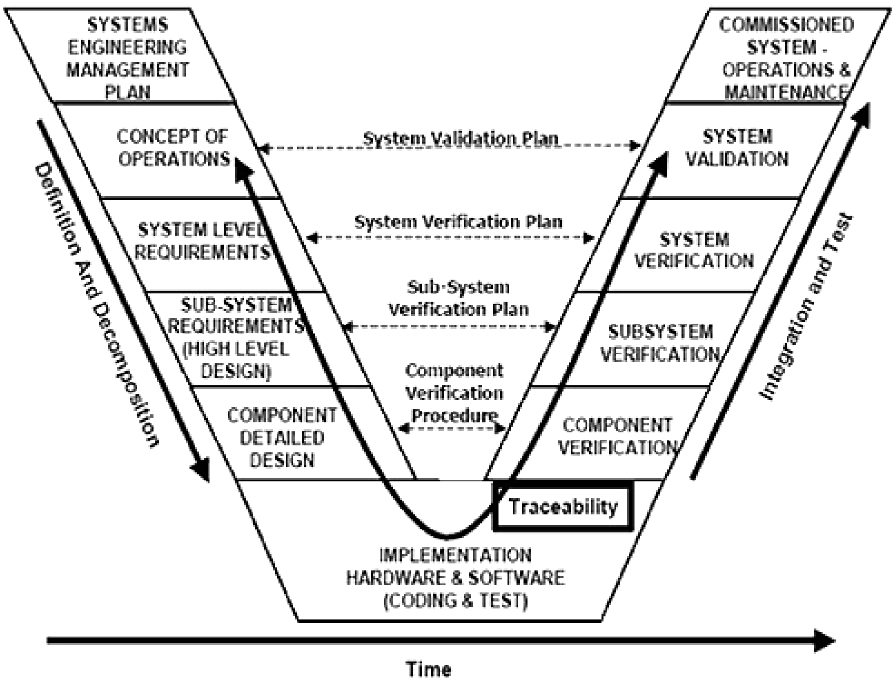


Figure F.0.6: V - Model, from Walker (2012)

Appendix G

Hardware Technical Specification

G.1 RJ45 Communication Standard

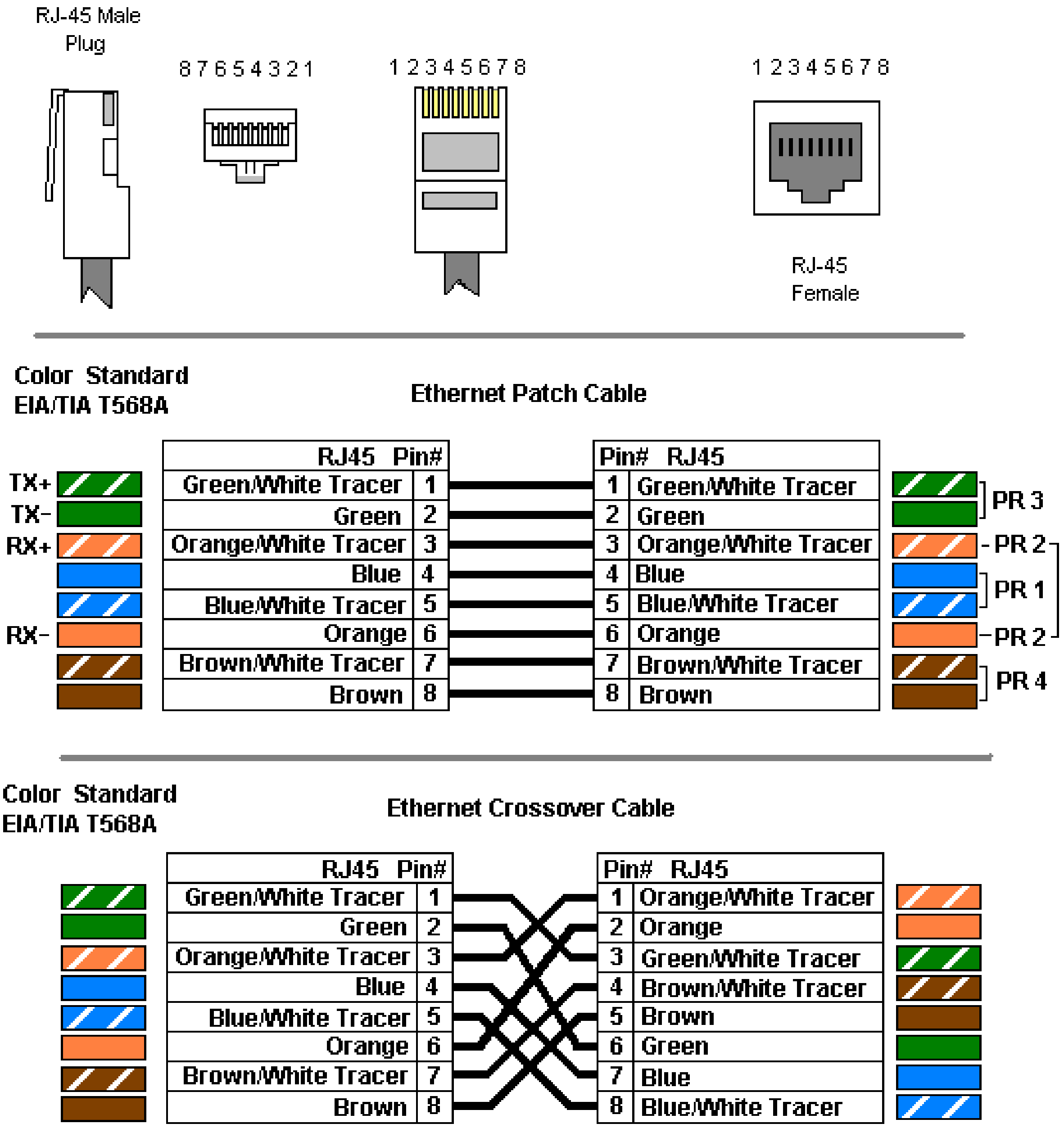


Figure G.1.1: RJ45 Standard with crossover, from B&B Electronics (2013)

G.2 Vision System Display Specification

47" FULL HD CINEMA 3D AND SMART TV WITH MAGIC MOTION REMOTE CONTROL

47LW6510

COMMON SPEC

Display Type	3D TV
Screen Size (Inch)	47

PICTURE

Resolution	1920 x 1080
Dynamic Contrast Ratio	9,000,000:1
ResponseTime(MPRT)	2ms
Full HD	Yes

VIDEO

AspectRatioCorrection	7 modes (16:9, Just scan, Original, Full Wide(DTV only), 4:3, 14:9, Cinema Zoom)
ColorTemperatureControl	Yes
Just Scan (0% overscan)	Yes
24p Real Cinema(24p 5:5/2:2 Pull dow n Mode)	Yes
XD Engine	Yes

SOUND

Audio Output	10W+10W
Speaker System	1 Way 2 Speakers
Dolby Digital Decoder	Yes
Surround System	Infinite 3D Surround
SoundMode	5 Modes (Standard/Music/Cinema/Sport/Game)
Clear Voice II	Yes

FEATURE

Invisible Speaker	Yes
Smart Energy Saving Plus	Yes
TruMotion	Yes
1080P Source Input	HDMI 60p/50p/30p/24p Component 60p/50p
SIMPLINK (HDMI CEC)	Yes
USB 2.0	MP3, JPEG, DivX
Intelligent Sensor	Yes
Picture Wizard	Yes

INTERFACE

Headphone Out	Yes
AV In	Yes
RGB In (Dsub 15pin)	Yes
PC Audio Input	Yes
HDMI/HDCP Input	HDMI In (4)
USB 2.0	Yes
RF In	Yes
Component in (Y,Pb,Pr) + Audio	Yes
LAN	Yes
Digital Audio Out	Yes
RS232C (Control / SVC)	Yes

POWER

Voltage, Hz	100 ~ 240V 50/60Hz
Consumption	150W
Standby (Off mode)	0.1W↓

WEIGHT (KG)

Set (w/o stand)	18.2
Included stand	21

DIMENSION (WXHXD)

Set (w/o stand)	1119 x 684 x 29.9 (mm)
Included Stand	1119 x 749 x 255 (mm)

Appendix H

Software development

H.1 User Stories

User stories play a crucial role in extreme programming planning and execution. The user stories can also be used as a metaphor, which gives the developer a long term coherent goal to work towards. Developers can use the user as development guides to add features and specify functional requirements.

User stories are concise and is loaded with developmental information, although user participation is still key to delivering functional software for acceptance by the user. Some users are unable to clarify the need for some functionality or has a preconceived notion of what the function is, which may be in conflict with the reality of the function.

Some of the user stories used in the development of the telerobotic system is shown in table H.1.1.

Table H.1.1: Descriptive user stories used for software and GUI development

User story no.	Detail
1	I would like to select any input arm, depending if I am left or right handed. Further I would like to move the robot and save a point in space for my experiments. I would like to be able to load the previously saved point and have the robot move there by itself so that I can rest the experimental setup for repeated testing.
2	I would like to be able to turn rotation functionality off if I want to keep the pose of the robot constant when moving an object. I would like to be able to see all the video feeds on demand in real time. Force and movement scaling would be advantageous when working in confined or sensitive spaces.
3	I would like to be able to query and record robot telemetry on demand. Along with this data I would also like to be able to set all the parameters of the robot remotely, be it position or angular values.

4	I would like to be able to load and execute a robot job file or robot command file remotely. Also, I would like to give the robot singular commands that I can enter manually, in case I need to execute movement without being present at either the input or output side robots.
5	I would like to be able to lock the robot movement in a fixed plane of my choosing and be able to move the output robot with the use of my computer mouse.
6	I would like to see what the force torque sensors are doing, what is being read by the sensors and what commands are being generated from them must be visible, for either understanding behaviour of the robot or troubleshooting if behaviour is inconsistent.
7	I want to be able to fill the video on a secondary screen so that it does not interfere with SenCon control. This will be necessary as to not impede control functionality and also allow me to see the workspace better.

Technical Client Stories	
no.	Detail
1	Must be able to execute and generate any of the robots, functional commands specified in the Yasakawa Motoman INFORM language.
2	Must have control over PI controller values for experimentation and be able to set baseline speed reference.
3	Must be able to reset robot position at any time and must always be able to activate and reset crucial functions such as Alarms and Emergency stop.
4	Must be modular design, Controller, display, interface, status query, command execution etc. must all be separate in software. This will allow changes to controllers, video hardware and feedback, audio feedback and FT sensor hardware to be upgraded or changed without effecting the rest of the software.

H.2 Functional Requirements Breakdown

For the extreme programming scenario presented, user stories and internal customer interviews were used in order to draft functional requirements. These were generated in response to two developmental questions relating to Developmental requirements and GUI design principles set out in section 3.3.3.

From a UCSD perspective the user roles need to be identified and defined, the user breakdown and roles are shown in table H.2.1.

Table H.2.1: User class functional breakdown

User Type	Description
Administrator/ Developer	Access higher functionality, direct control over all aspects of the system including the functional components and settings available to end-users. In essence an developer user must be able to access the end-user profile with the addition of hardware and software control parameters.
End User	Functional components and settings only

Further it is necessary to define the specific functions that a user requires for the software and more specifically the interface. The initial Functional requirements breakdown were generated in response to the following two questions:

- In general what must the GUI enable you to do, for you as user?
- What information would you like to see and at what stage of the interaction, if applicable?

The combined response to question one is shown in table H.2.2 and the combined information display requirement is shown in table H.2.3. These tables clearly show the user and developer specific functional and information display requirements.

Table H.2.2: User generated functional requirement breakdown for telerobotic system and GUI, response to question 1

Use scenario	Description	Functional Requirement
When ever in use		
	Must be able to control the speed of the robot. (Dev.)	Max speed controllable
	Must be able to select what modality the input is given (Haptic/Joystick).	Select Haptic/Joystick mode
	Must be able to scale the output of the system.	Force or control command scaling
	Must be able to pre-set/train a bias force/Torque setting for both input and output sensors.	Force Bias set
	Must be able to switch the robots used on and off remotely.	Control Power status
	Must be able to save and load positional data and make the robot move to the predefined or taught position. (Dev.)	Load / Save pre-set or current position
	Must be able to switch between the viewing angles.	Viewing feed select
	Select which robot arm is master and which is slave.	Input robot select
	Select if wrist rotation is active or not.	Set rotation status
	Activate software emergency stop (E-stop) interlock.	Set E-stop
	Show and hide video feeds.	Video feed pane status set
	Video Feed selection.	Select video feed
	Set video feed to full screen on secondary screen.	Video full-screen mode status set
	Error reset functionality for non-severe errors.	Error reset
When under command scenario		
	Enables direct access to core functionality only	
	Load pre-programmed job file from windows.	Load and execute ".CSV" files
	Report movement commands at time of execution.	Command log streaming

Manual command scenario (Dev.)

	Allow manual software control of both master and slave robots.	
	Able to select Axis that must be controlled, X, Y and Z.	Axis select
	Directional control, both positive and negative displacement.	Direction select
	Hold position command.	Hold enable
	Select robot that must execute current command.	Robot select
	Edit and control positional parameters of a given robot, on a joint and angle level.	Positional parameter set
	Show positional data of a given robot.	Query robot position
	Must be able to enter full command, with error reporting, from GUI.	Text command execution
	Easily set camera arm, if present, to selected viewing angle.	Easy select POV, Home

Planar use scenario

	When a plane is selected and there is no Z-displacement, i.e. no change in height.	
	Set plane height and enable planar mode.	Teach plane height
	Speed selection.	Set speed
	Force or command scaling.	Scale speed
	Experiment selection for record keeping purposes.	Experiment selection
	Choose between Joystick or haptic input for the experiment.	Input mode selection
	Timer start and stop functionality for experiments.	Experiment timer control
	Make command mouse pointer tracking window full screen.	Full screen mouse tracking

	Velocity scaling option, the faster you move the mouse, the faster the robot moves.	Mouse velocity scaling
Developer specific scenario		
	Administrator functionality that will not be available to end-users	
	Set the controller K_i , K_p , K_d values if present.	Control parameter value set

Table H.2.3: User generated information display requirement, response to question 2

Use scenario	Description	Functional Requirement
When ever in use		
	System status	Current system status
	Are there any errors	Error status
	What is the error	Error description
	Can the error be reset	Error reset status
	Is the power ON or OFF?	Servo power status
	Emergency stop status	E-Stop status
	Video feed preview pane	Video feed preview
	Rotation enable status	Rotation indicator
Manual command scenario (Dev.)		
	What command is currently being executed?	Command stream display box
	History of commands that have been executed.	Command log display box
Planar use scenario		
	Is the planar control active?	Colour change indicator (green = on / red = off)
	Which Experiment is currently active?	Experiemnt name indicator

H.3 Software Class Diagram

The class diagram was produced by reverse engineering (refactoring) the final implemented software.

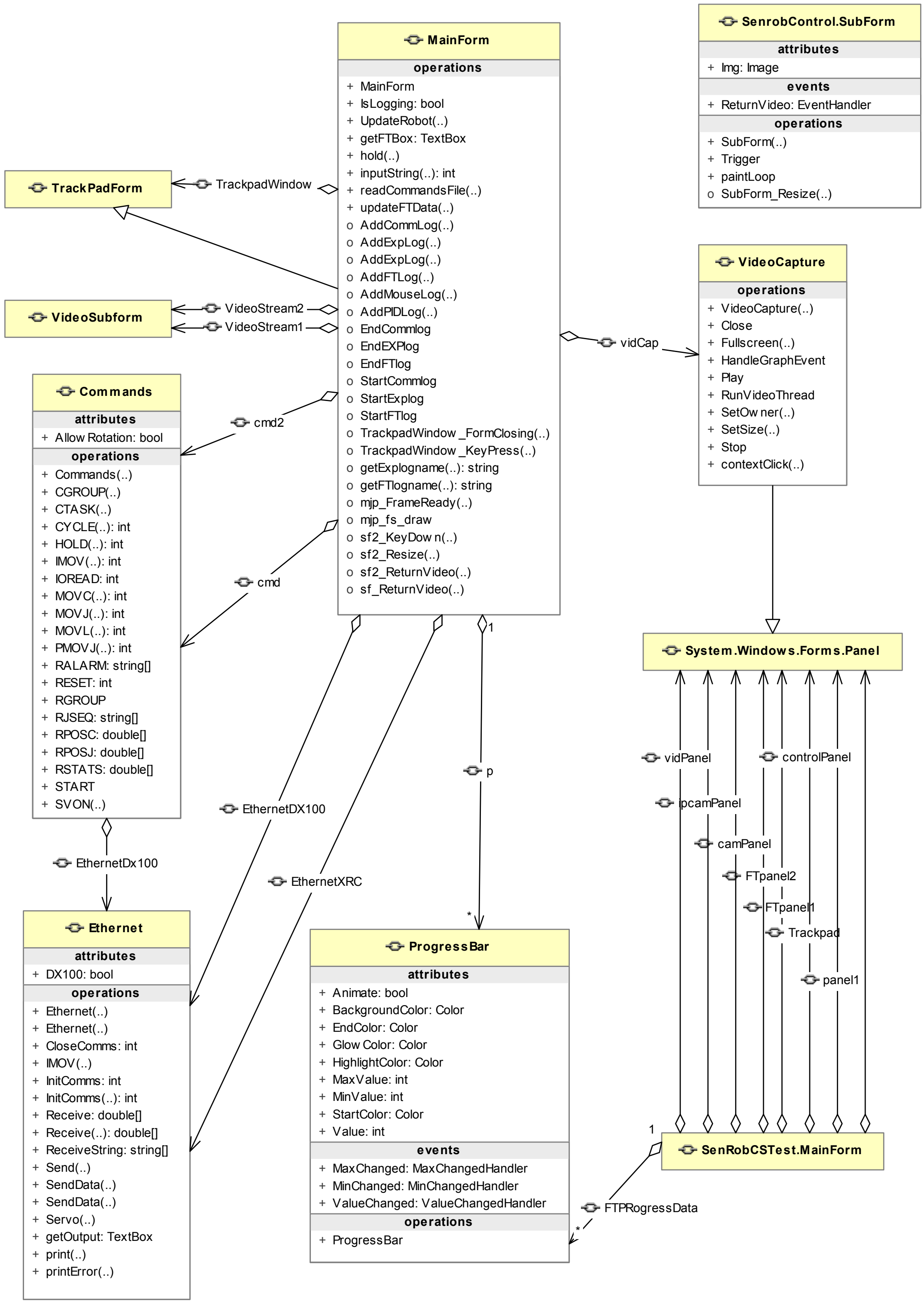


Figure H.3.1: SenCon Class Diagram

Appendix I

Development Console


Ethernet Test
Control Panel
Clear Alarms
Servo ON
Servo OFF
Show Cam Panel
Factory Reset
Rotation
mm/s °/s
50
SDA 10 memory
Left
Right
Save
Save
Load
Load
Bias FT sensors
FT activity
P value (max 1)
20
I value (max 0.5)
900
UP6
Status
RDTSeq2
FTSeq2
Fx2
Fy2
Fz2
Tx2
Ty2
Tz2
SDA 10
Status
RDTSeq1
FTSeq1
Fx1
Fy1
Fz1
Tx1
Ty1
Tz1
Haptic scale
100 %
Haptic
STOP
Joystick
SDA 10
LEFT
RIGHT
Command history
IP Camera
Fullscreen
HDMI Camera
Fullscreen
Alarm status
1
2
3
4
5
6
7
8
9
10

I.1 Development Console Definitions

Table I.1.1: Development Console Functional Definitions

Legend number	Functional Components	Purpose	Functional Description
1	Servo Control Area		
	Clear Alarms	Interaction / Status	When button is pressed all current system errors or alarms indicated in 10, is cleared and the system may be restarted.
	Servo On	Interaction	Turns robot servo power on.
	Servo Off	Interaction	Turns robot servo power off.
2	Functional Settings Area		
	Show Cam Panel - radio button	Interaction / Status	When radio button is clicked, enable / disable camera video feed panel as shown by 8.
	Factory Reset	Control action	Resets the robot control variables and position to robot native home position.
	Rotation - with button status indicator	Control action	Enable (Green) / Disable (Red) rotation control of the robot manipulator.
3	Speed and Repeated Position Area		
	mm/s - rotation and linear speed magnitude	Control variable input	Allows for control speed of both linear and rotation to be input via a text box.
	Save (left / right)	Control action	Saves the current joint position of the selected robot arm to memory.
	Load (left / right)	Control action	Loads and executes movement of the selected robot arm from memory.
4	PI Controller Settings Area		
	P Value - textbox	Control variable input	Enables developers to tune K_p value of the PI controller
	I value - textbox	Control variable input	Enables developers to tune K_i value of the PI controller

Table continues on next page...

5	Master Robot Select area		
	Left	Control action	Selects left robot arm as master input.
	Right	Control action	Selects right robot arm as master input.
6	Input Modality and Sensitivity Select Area		
	Haptic scale - slider	Control variable input	Determines the sensitivity of the control input.
	Haptic	Control action	When selected enables the haptic, kinaesthetic and force feedback, control modality
	Stop	Control action	Software enabled EMERGENCY STOP.
	Joystick	Control action	When Selected enables the joystick control modality.
7	Force Information and Control Area		
	Robot Readout bars	Status	Detailed force and torque value indication.
	FT activity - radial indicator	Status	Flashes when communication is sent or received.
	Bias FT sensors	Interaction	Biases the force / torque sensor values to zero.
8	Vision System feedback Area		
	IP Camera	Status / feedback	Shows the live PTZ, IP Camera feed.
	HDMI Camera	Status / feedback	Shows the live HDMI (3D) camera feed.
	Fullscreen (IP / HDMI)	Interaction	Enables a fullscreen view (on a secondary screen) of the chosen camera feed.
9	Command History Area		
	Command History Feed	Status	Shows a live, scrollable, complete command history feed.
10	Alarm Status Area		
	Alarm Status - status bar	Status	Shows detailed information on the current, prioritised alarms active on the system.

Appendix J

Mechanical Component Designs

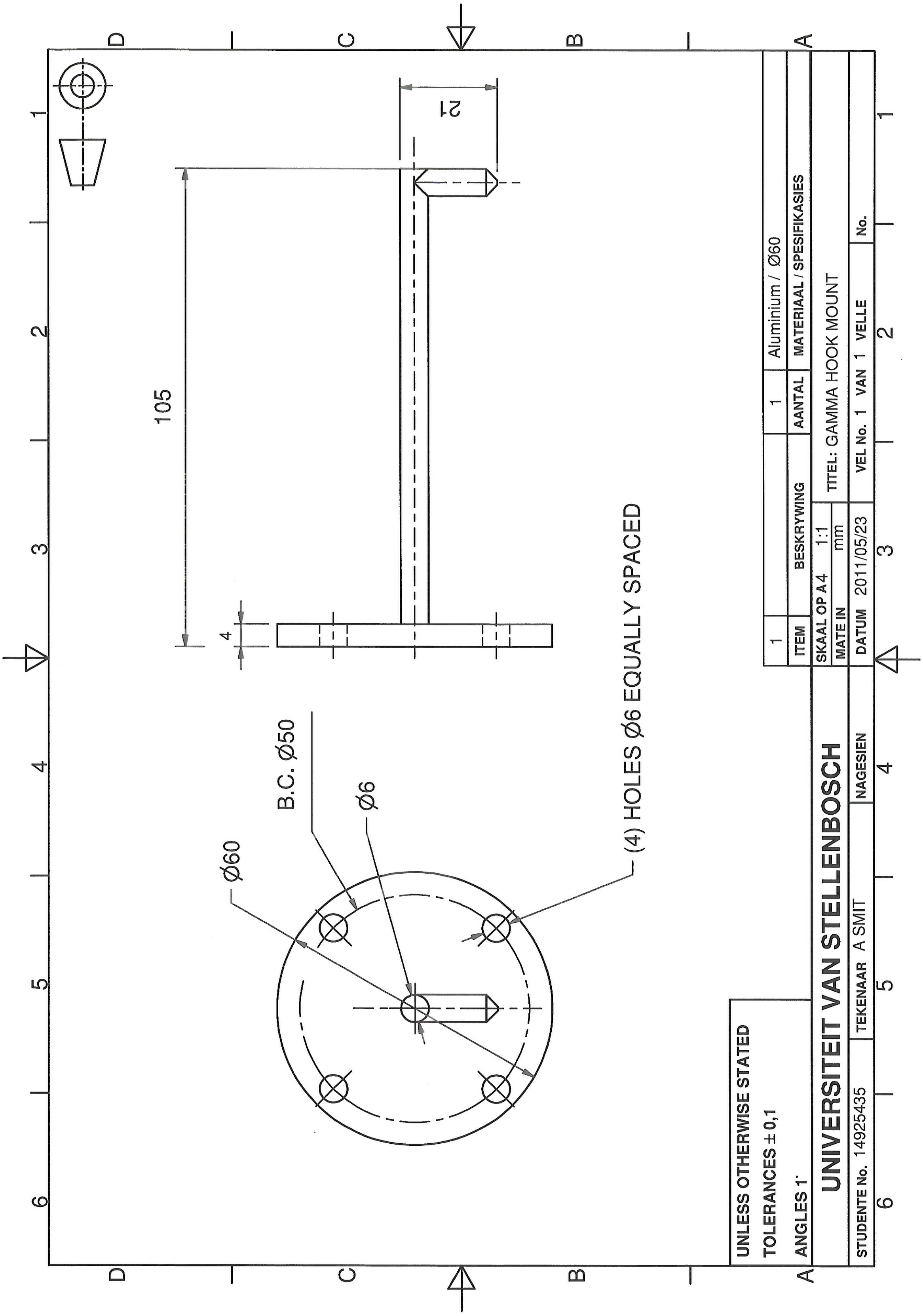


Figure J.0.1: Mechanical Hook End-effector

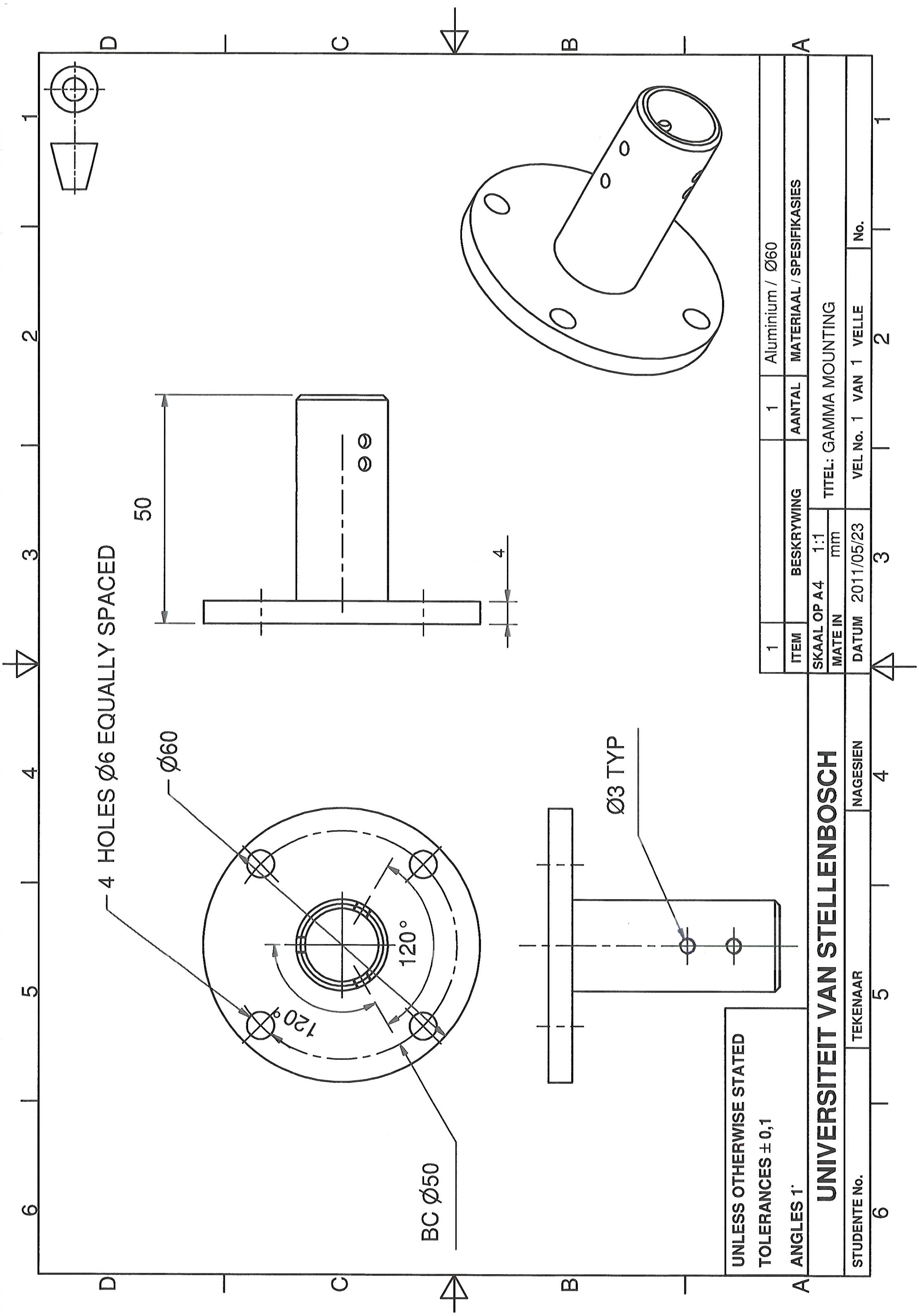


Figure J.0.2: Mechanical Hook End-effector

Appendix K

Result Set

For any further information please see attached CD for excel files. ANOVA, Wilcoxon and Tukey HSD used R-statistics' built in functions to analyse, with the exception of the Friedman test which is described in appendix B.

K.1 ANOVA - Experimental Results

Experimental Results

2D Planar Exp - Anova Results

Original User Group: 10 users — Response Variable : Time to completion

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Input	1	392,1	392,09	4,0434	0,04809	*
ViewAngl	1	45,7	45,65	0,4708	0,49484	
ViewMode	1	7,5	7,54	0,0778	0,78113	
Input:Vi	1	0	0	0	0,99648	
Input:Vi	1	137,5	137,45	1,4175	0,23773	
ViewAngl	1	199,2	199,19	2,0541	0,15612	
Input:Vi	1	3,5	3,51	0,0362	0,84955	
Residual	72	6981,8	96,97			

Signif. 0 01 '**' 0.01 '* ' 0.05 ' .' 0.1 ' ' 1

A.Smit - Thesis Case Study - User performance under varying vision modes

Experimental Results

3D Obstacle Exp - Anova Results

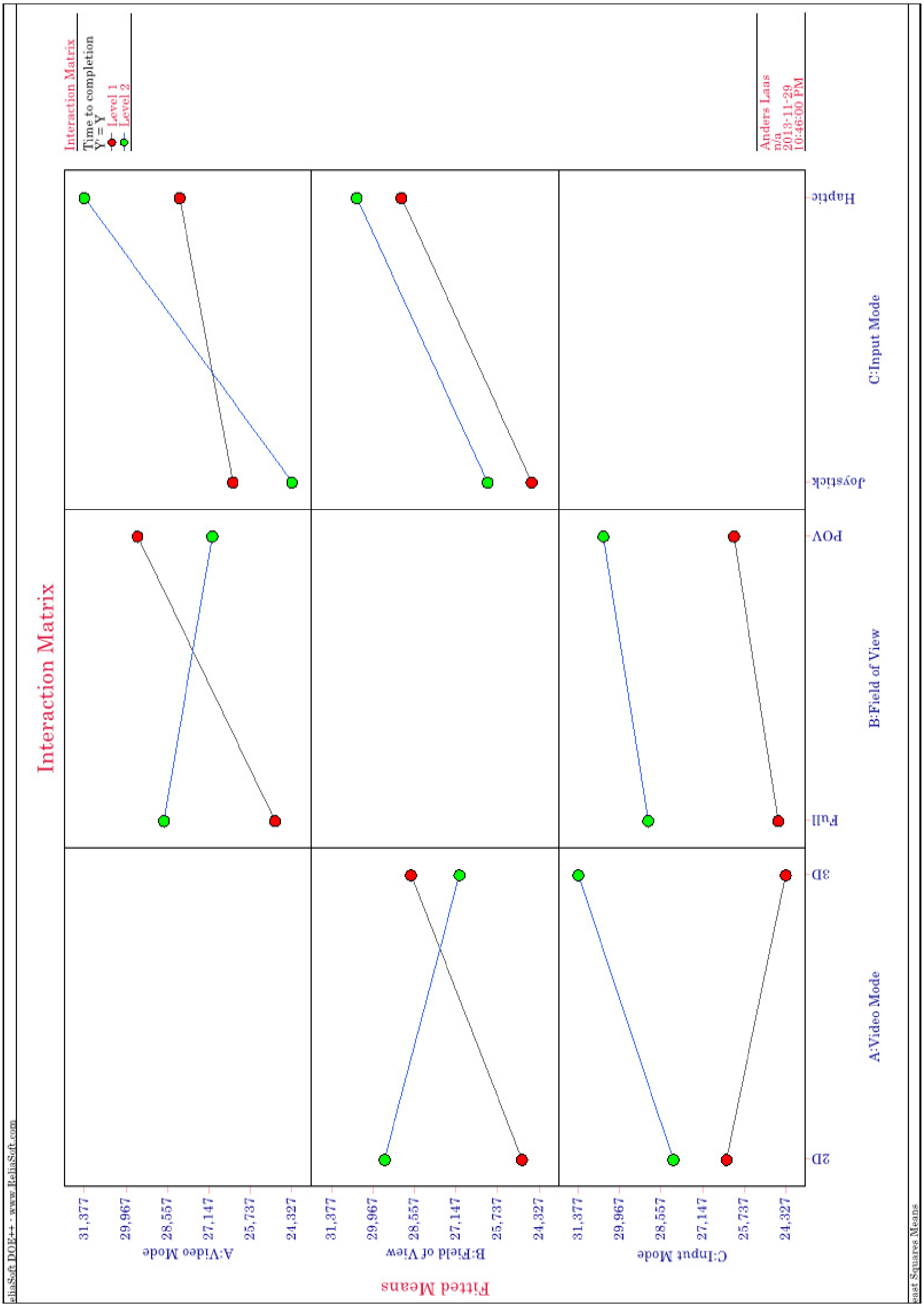
Original User Group: 8 users — Response Variable : Time to completion

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Input	1	10,5	10,47	0,0743	0,7862
ViewAngle	1	0	0,041	0,0003	0,9865
ViewMode	1	113,3	113,348	0,8042	0,3737
Input:ViewAngle	1	112,6	112,593	0,7988	0,3753
Input:ViewMode	1	28,8	28,751	0,204	0,6533
ViewAngle:ViewMode	1	1,1	1,086	0,0077	0,9304
Input:ViewAngle:ViewMode	1	199,8	199,805	1,4176	0,2388
Residuals	56	7893,2	140,949		

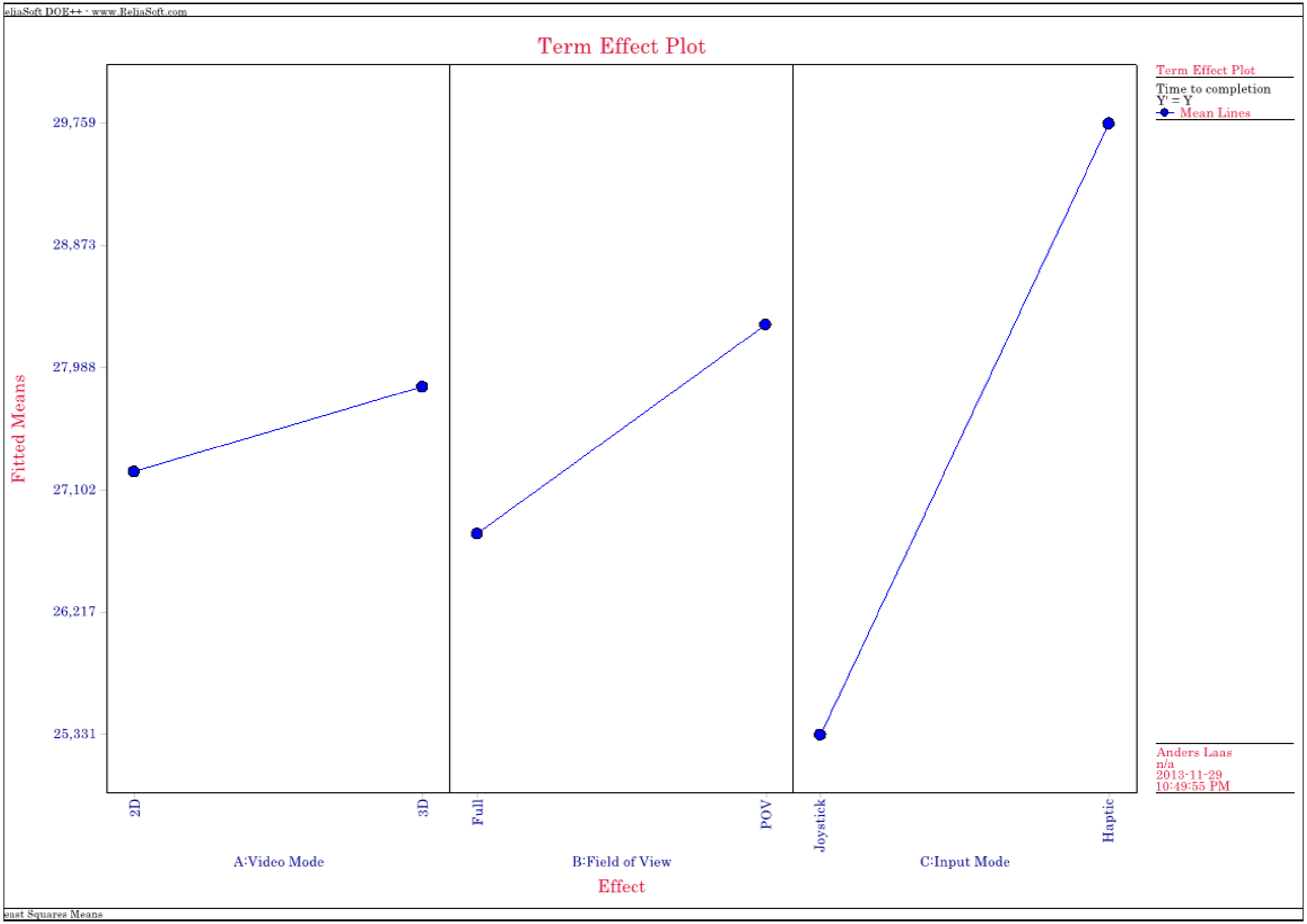
Signif. codes: 0 '***' 0 01 '**' 0.01 '* ' 0.05 ' .' 0.1 ' ' 1

A.Smit - Thesis Case Study - User performance under varying vision modes

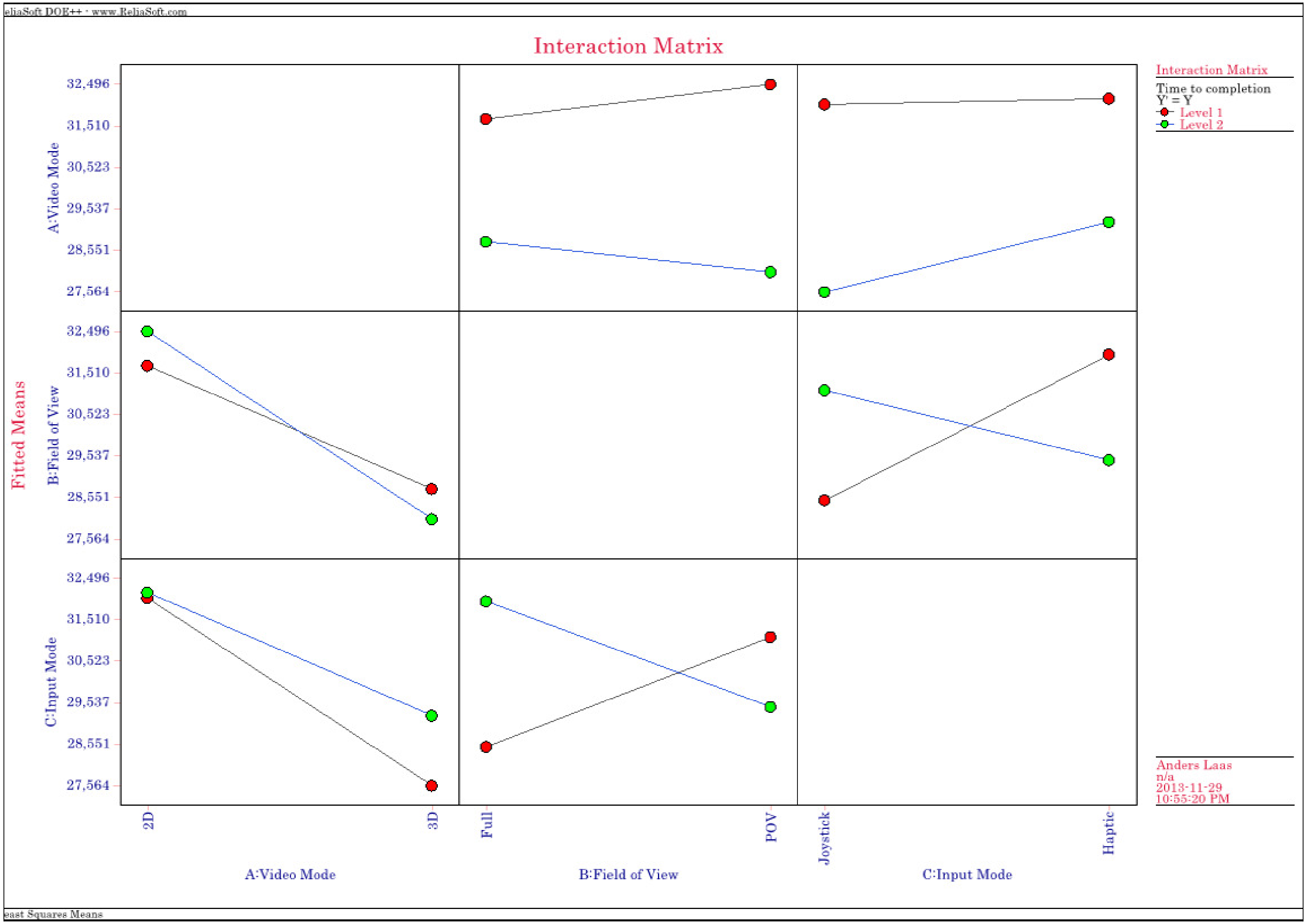
K.1.1 2D Planar Experiment Interaction Plot



K.1.2 2D Experiment Main Effect Plot



K.1.3 3D Experiment Interaction Plot



K.2 2D Experiment - Tukey HSD

Full 2D Planar Results

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Y ~ Input * ViewAngle * ViewMode)
```

$Input

	diff	lwr	upr	p adj
Haptic-Joystick	4,428	0,03851833	8,817482	0,0480751

$ViewAngle

	diff	lwr	upr	p adj
FF-POV	-1,511	-5,900482	2,878482	0,4947822

$ViewMode

	diff	lwr	upr	p adj
3D-2D	0,6145	-3,774982	5,003982	0,7809893

$`Input:ViewAngle`

	diff	lwr	upr	p adj
Haptic:POV-Joystick:POV	4,437	-3,753036	12,627036	0,488203
Joystick:FF-Joystick:POV	-1,502	-9,692036	6,688036	0,962777
Haptic:FF-Joystick:POV	2,917	-5,273036	11,107036	0,7852269
Joystick:FF-Haptic:POV	-5,939	-14,129036	2,251036	0,2341756
Haptic:FF-Haptic:POV	-1,52	-9,710036	6,670036	0,9615052
Haptic:FF-Joystick:FF	4,419	-3,771036	12,609036	0,4917483

$`Input:ViewMode`

	diff	lwr	upr	p adj
Haptic:2D-Joystick:2D	1,8065	-6,383536	9,996536	0,9377455
Joystick:3D-Joystick:2D	-2,007	-10,197036	6,183036	0,9171289
Haptic:3D-Joystick:2D	5,0425	-3,147536	13,232536	0,3743995
Joystick:3D-Haptic:2D	-3,8135	-12,003536	4,376536	0,6133929
Haptic:3D-Haptic:2D	3,236	-4,954036	11,426036	0,727095
Haptic:3D-Joystick:3D	7,0495	-1,140536	15,239536	0,1162877

$`ViewAngle:ViewMode`

	diff	lwr	upr	p adj
FF:2D-POV:2D	-4,6665	-12,856536	3,523536	0,4436986
POV:3D-POV:2D	-2,541	-10,731036	5,649036	0,8467251
FF:3D-POV:2D	-0,8965	-9,086536	7,293536	0,9916156
POV:3D-FF:2D	2,1255	-6,064536	10,315536	0,9034088
FF:3D-FF:2D	3,77	-4,420036	11,960036	0,6221639
FF:3D-POV:3D	1,6445	-6,545536	9,834536	0,9519989

$`Input:ViewAngle:ViewMode`

	diff	lwr	upr	p adj
Haptic:POV:2D-Joystick:POV:2D	2,234	-11,514054	15,982054	0,9995914
Joystick:FF:2D-Joystick:POV:2D	-4,239	-17,987054	9,509054	0,9781959
Haptic:FF:2D-Joystick:POV:2D	-2,86	-16,608054	10,888054	0,9979726
Joystick:POV:3D-Joystick:POV:2D	-4,744	-18,492054	9,004054	0,9595447
Haptic:POV:3D-Joystick:POV:2D	1,896	-11,852054	15,644054	0,9998629
Joystick:FF:3D-Joystick:POV:2D	-3,509	-17,257054	10,239054	0,992786
Haptic:FF:3D-Joystick:POV:2D	3,95	-9,798054	17,698054	0,9854461
Joystick:FF:2D-Haptic:POV:2D	-6,473	-20,221054	7,275054	0,8206119
Haptic:FF:2D-Haptic:POV:2D	-5,094	-18,842054	8,654054	0,9413235
Joystick:POV:3D-Haptic:POV:2D	-6,978	-20,726054	6,770054	0,7578314
Haptic:POV:3D-Haptic:POV:2D	-0,338	-14,086054	13,410054	1
Joystick:FF:3D-Haptic:POV:2D	-5,743	-19,491054	8,005054	0,894412
Haptic:FF:3D-Haptic:POV:2D	1,716	-12,032054	15,464054	0,99993
Haptic:FF:2D-Joystick:FF:2D	1,379	-12,369054	15,127054	0,9999842
Joystick:POV:3D-Joystick:FF:2D	-0,505	-14,253054	13,243054	1
Haptic:POV:3D-Joystick:FF:2D	6,135	-7,613054	19,883054	0,8574744
Joystick:FF:3D-Joystick:FF:2D	0,73	-13,018054	14,478054	0,9999998
Haptic:FF:3D-Joystick:FF:2D	8,189	-5,559054	21,937054	0,5825256
Joystick:POV:3D-Haptic:FF:2D	-1,884	-15,632054	11,864054	0,9998686
Haptic:POV:3D-Haptic:FF:2D	4,756	-8,992054	18,504054	0,9589963
Joystick:FF:3D-Haptic:FF:2D	-0,649	-14,397054	13,099054	0,9999999
Haptic:FF:3D-Haptic:FF:2D	6,81	-6,938054	20,558054	0,7796484
Haptic:POV:3D-Joystick:POV:3D	6,64	-7,108054	20,388054	0,8008057
Joystick:FF:3D-Joystick:POV:3D	1,235	-12,513054	14,983054	0,9999926
Haptic:FF:3D-Joystick:POV:3D	8,694	-5,054054	22,442054	0,5061295
Joystick:FF:3D-Haptic:POV:3D	-5,405	-19,153054	8,343054	0,921046
Haptic:FF:3D-Haptic:POV:3D	2,054	-11,694054	15,802054	0,9997659
Haptic:FF:3D-Joystick:FF:3D	7,459	-6,289054	21,207054	0,6911757

K.3 3D Experiment - Tukey HSD

Full 3D Tukey

```
Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Y ~ Input * ViewAngle * ViewMode)

$Input
                     diff       lwr      upr     p adj
Haptic-Joystick 0.900625 -5.085675 6.886925 0.7642388

$ViewAngle
        diff     lwr    upr   p adj
FF-POV -0.05 -6.0363 5.9363 0.98671

$ViewMode
           diff       lwr      upr     p adj
3D-2D -2.078125 -8.064425 3.908175 0.4896688

$`Input:ViewAngle`
                               diff        lwr       upr     p adj
Haptic:POV-Joystick:POV  -1.659375 -12.849625  9.530875 0.9792596
Joystick:FF-Joystick:POV -2.610000 -13.800250  8.580250 0.9260464
Haptic:FF-Joystick:POV    0.850625 -10.339625 12.040875 0.9970791
Joystick:FF-Haptic:POV   -0.950625 -12.140875 10.239625 0.9959393
Haptic:FF-Haptic:POV      2.510000  -8.680250 13.700250 0.9335037
Haptic:FF-Joystick:FF     3.460625  -7.729625 14.650875 0.8453036

$`Input:ViewMode`
                             diff       lwr      upr     p adj
Haptic:2D-Joystick:2D    1.75000  -9.44025 12.94025 0.9758355
Joystick:3D-Joystick:2D -1.22875 -12.41900  9.96150 0.9913467
Haptic:3D-Joystick:2D   -1.17750 -12.36775 10.01275 0.9923643
Joystick:3D-Haptic:2D   -2.97875 -14.16900  8.21150 0.8947538
Haptic:3D-Haptic:2D     -2.92750 -14.11775  8.26275 0.8994546
Haptic:3D-Joystick:3D    0.05125 -11.13900 11.24150 0.9999994

$`ViewAngle:ViewMode`
                   diff       lwr       upr     p adj
FF:2D-POV:2D   0.792500 -10.39775 11.982750 0.9976330
POV:3D-POV:2D -1.235625 -12.42588  9.954625 0.9912039
FF:3D-POV:2D  -2.128125 -13.31838  9.062125 0.9579014
POV:3D-FF:2D  -2.028125 -13.21838  9.162125 0.9632331
FF:3D-FF:2D   -2.920625 -14.11088  8.269625 0.9000767
FF:3D-POV:3D  -0.892500 -12.08275 10.297750 0.9966316
```

```
$`Input:ViewAngle:ViewMode`
                                   diff       lwr      upr     p adj
Haptic:POV:2D-Joystick:POV:2D    2.23125 -16.58482 21.04732 0.9999460
Joystick:FF:2D-Joystick:POV:2D   1.27375 -17.54232 20.08982 0.9999988
Haptic:FF:2D-Joystick:POV:2D     2.54250 -16.27357 21.35857 0.9998697
Joystick:POV:3D-Joystick:POV:2D  2.65500 -16.16107 21.47107 0.9998259
Haptic:POV:3D-Joystick:POV:2D   -2.89500 -21.71107 15.92107 0.9996904
Joystick:FF:3D-Joystick:POV:2D  -3.83875 -22.65482 14.97732 0.9980623
Haptic:FF:3D-Joystick:POV:2D     1.81375 -17.00232 20.62982 0.9999868
Joystick:FF:2D-Haptic:POV:2D    -0.95750 -19.77357 17.85857 0.9999998
Haptic:FF:2D-Haptic:POV:2D       0.31125 -18.50482 19.12732 1.0000000
Joystick:POV:3D-Haptic:POV:2D    0.42375 -18.39232 19.23982 1.0000000
Haptic:POV:3D-Haptic:POV:2D     -5.12625 -23.94232 13.68982 0.9885728
Joystick:FF:3D-Haptic:POV:2D    -6.07000 -24.88607 12.74607 0.9701287
Haptic:FF:3D-Haptic:POV:2D      -0.41750 -19.23357 18.39857 1.0000000
Haptic:FF:2D-Joystick:FF:2D      1.26875 -17.54732 20.08482 0.9999989
Joystick:POV:3D-Joystick:FF:2D   1.38125 -17.43482 20.19732 0.9999980
Haptic:POV:3D-Joystick:FF:2D    -4.16875 -22.98482 14.64732 0.9967429
Joystick:FF:3D-Joystick:FF:2D   -5.11250 -23.92857 13.70357 0.9887519
Haptic:FF:3D-Joystick:FF:2D      0.54000 -18.27607 19.35607 1.0000000
Joystick:POV:3D-Haptic:FF:2D     0.11250 -18.70357 18.92857 1.0000000
Haptic:POV:3D-Haptic:FF:2D      -5.43750 -24.25357 13.37857 0.9838997
Joystick:FF:3D-Haptic:FF:2D     -6.38125 -25.19732 12.43482 0.9608604
Haptic:FF:3D-Haptic:FF:2D       -0.72875 -19.54482 18.08732 1.0000000
Haptic:POV:3D-Joystick:POV:3D   -5.55000 -24.36607 13.26607 0.9818962
Joystick:FF:3D-Joystick:POV:3D  -6.49375 -25.30982 12.32232 0.9570576
Haptic:FF:3D-Joystick:POV:3D    -0.84125 -19.65732 17.97482 0.9999999
Joystick:FF:3D-Haptic:POV:3D    -0.94375 -19.75982 17.87232 0.9999999
Haptic:FF:3D-Haptic:POV:3D       4.70875 -14.10732 23.52482 0.9931147
Haptic:FF:3D-Joystick:FF:3D      5.65250 -13.16357 24.46857 0.9799128
```

K.4 2D Experiment - Friedman

Experimental Results

Friedman Test: 2D-Planar Page: 1

Test 1

Haptic	
Full	
2D	3D
17,435	21,723
25,566	27,788
15,972	20,734
34,312	40,1
20,547	45,601
26,149	27,274
25,346	28,499
24,87	27,827
21,832	37,436
43,903	47,057

Test2

Haptic	
POV	
2D	3D
30,152	20,362
30,709	30,604
30,249	18,915
46,323	36,117
43,862	28,878
26,49	26,777
21,311	23,326
22,839	38,728
35,733	36,614
19,22	43,173

Test3

Haptic	
2D	
Full	POV
17,435	30,152
25,566	30,709
15,972	30,249
34,312	46,323
20,547	43,862
26,149	26,49
25,346	21,311
24,87	22,839
21,832	35,733
43,903	19,22

Test4

Haptic	
3D	
Full	POV
21,723	20,362
27,788	30,604
20,734	18,915
40,1	36,117
45,601	28,878
27,274	26,777
28,499	23,326
27,827	38,728
37,436	36,614
47,057	43,173

Test5

Joystick	
Full	
2D	3D
19,151	28,191
12,790	15,96
14,531	16,073
23,651	17,903
22,95	32,971
22,692	32,4
19,681	19,31
61,943	26,495
17,764	19,884
30,007	40,265

Test6

Joystick	
POV	
2D	3D
31,204	14,665
18,752	18,12
18,536	21,162
21,931	19,047
39,759	26,884
21,879	19,244
27,539	22,182
30,11	41,347
20,912	20,076
53,915	34,369

Test7

Joystick	
2D	
Full	POV
19,151	31,204
12,790	18,752
14,531	18,536
23,651	21,931
22,95	39,759
22,692	21,879
19,681	27,539
61,943	30,11
17,764	20,912
30,007	53,915

Test8

Joystick	
3D	
Full	POV
28,191	14,665
15,96	18,12
16,073	21,162
17,903	19,047
32,971	26,884
32,4	19,244
19,31	22,182
26,495	41,347
19,884	20,076
40,265	34,369

A.Smit - Thesis Case Study - User performance under varying vision modes

Experimental Results

Friedman Test: 2D-Planar Page: 2

Test9 POV 2D Joystick	Test9 POV 2D Haptic	Test10 FF 2D Joystick	Test10 FF 2D Haptic	Test11 POV 3D Joystick	Test11 POV 3D Haptic	Test12 FF 3D Joystick	Test12 FF 3D Haptic
19,151	17,435	28,191	21,723	31,204	30,152	14,665	20,362
12,790	25,566	15,96	27,788	18,752	30,709	18,12	30,604
14,531	15,972	16,073	20,734	18,536	30,249	21,162	18,915
23,651	34,312	17,903	40,1	21,931	46,323	19,047	36,117
22,95	20,547	32,971	45,601	39,759	43,862	26,884	28,878
22,692	26,149	32,4	27,274	21,879	26,49	19,244	26,777
19,681	25,346	19,31	28,499	27,539	21,311	22,182	23,326
61,943	24,87	26,495	27,827	30,11	22,839	41,347	38,728
17,764	21,832	19,884	37,436	20,912	35,733	20,076	36,614
30,007	43,903	40,265	47,057	53,915	19,22	34,369	43,173

Significant Results

```
data:  test1
```
Friedman chi-squared = 10, df = 1, p-value = 0.001565
```
data:  test4
```
Friedman chi-squared = 3.6, df = 1, p-value = 0.05778
```
data:  test6
```
Friedman chi-squared = 3.6, df = 1, p-value = 0.05778
```
data:  test10
```
Friedman chi-squared = 3.6, df = 1, p-value = 0.05778
```
data:  test12
```
Friedman chi-squared = 3.6, df = 1, p-value = 0.05778

A.Smit - Thesis Case Study - User performance under varying vision modes

K.5 3D Experiment - Friedman

Experimental Results

Friedman Test: 3D-Obstacle Page: 1

Test1

Haptic	
Full	
2D	3D
27,312	31,369
28,817	25,634
29,11	36,036
30,964	32,01
19,747	18,761
35,668	25,636
52,669	45,951
43,203	37,262

Test2

Haptic	
POV	
2D	3D
18,772	26,198
29,595	28,058
38,223	25,335
25,765	14,628
15,047	16,957
37,542	32,28
51,065	44,878
39,979	26,641

Test3

Haptic	
2D	
Full	POV
27,312	18,772
28,817	29,595
29,11	38,223
30,964	25,765
19,747	15,047
35,668	37,542
52,669	51,065
43,203	39,979

Test4

Haptic	
3D	
Full	POV
31,369	26,198
25,634	28,058
36,036	25,335
32,01	14,628
18,761	16,957
25,636	32,28
45,951	44,878
37,262	26,641

Test5

Jostick	
Full	
2D	3D
23,04	21,891
18,563	30,325
25,462	24,346
52,911	36,261
20,032	10,145
44,681	20,205
22,947	26,951
40,707	37,295

Test6

Joystick	
POV	
2D	3D
21,552	30,578
18,822	21,554
37,614	65,268
22,3	24,413
54,252	19,222
28,063	28,394
38,238	36,857
43,113	17,32

Test7

Joystick	
2D	
Full	POV
23,04	21,552
18,563	18,822
25,462	37,614
52,911	22,3
20,032	54,252
44,681	28,063
22,947	38,238
40,707	43,113

Test8

Joystick	
3D	
Full	POV
21,891	30,578
30,325	21,554
24,346	65,268
36,261	24,413
10,145	19,222
20,205	28,394
26,951	36,857
37,295	17,32

A.Smit - Thesis Case Study - User performance under varying vision modes

Friedman Test: 3D-Obstacle Page: 2

Test9	2D		Test10	3D		Test11	2D		Test12	3D	
	Full			Full			POV			POV	
	Haptic	Joystick		Haptic	Joystick		Haptic	Joystick		Haptic	Joystick
	27,312	23,04		31,369	21,891		18,772	21,552		26,198	30,578
	28,817	18,563		25,634	30,325		29,595	18,822		28,058	21,554
	29,11	25,462		36,036	24,346		38,223	37,614		25,335	65,268
	30,964	52,911		32,01	36,261		25,765	22,3		14,628	24,413
	19,747	20,032		18,761	10,145		15,047	54,252		16,957	19,222
	35,668	44,681		25,636	20,205		37,542	28,063		32,28	28,394
	52,669	22,947		45,951	26,951		51,065	38,238		44,878	36,857
	43,203	40,707		37,262	37,295		39,979	43,113		26,641	17,32

Near Significant results:

```
data:  test2
Friedman chi-squared = 2, df = 1, p-value = 0.1573
data:  test4
Friedman chi-squared = 2, df = 1, p-value = 0.1573
```

K.6 3D Experiment - Wilcoxon Rank Sum Test

Test 1

Haptic	
Full	
2D	3D
17,435	21,723
25,566	27,788
15,972	20,734
34,312	40,1
20,547	45,601
26,149	27,274
25,346	28,499
24,87	27,827
21,832	37,436
43,903	47,057

Test2

Haptic	
POV	
2D	3D
30,152	20,362
30,709	30,604
30,249	18,915
46,323	36,117
43,862	28,878
26,49	26,777
21,311	23,326
22,839	38,728
35,733	36,614
19,22	43,173

Test3

Haptic	
2D	
Full	POV
17,435	30,152
25,566	30,709
15,972	30,249
34,312	46,323
20,547	43,862
26,149	26,49
25,346	21,311
24,87	22,839
21,832	35,733
43,903	19,22

Test4

Haptic	
3D	
Full	POV
21,723	20,362
27,788	30,604
20,734	18,915
40,1	36,117
45,601	28,878
27,274	26,777
28,499	23,326
27,827	38,728
37,436	36,614
47,057	43,173

Test5

Joystick	
Full	
2D	3D
19,151	28,191
12,790	15,96
14,531	16,073
23,651	17,903
22,95	32,971
22,692	32,4
19,681	19,31
61,943	26,495
17,764	19,884
30,007	40,265

Test6

Joystick	
POV	
2D	3D
31,204	14,665
18,752	18,12
18,536	21,162
21,931	19,047
39,759	26,884
21,879	19,244
27,539	22,182
30,11	41,347
20,912	20,076
53,915	34,369

Test7

Joystick	
2D	
Full	POV
19,151	31,204
12,790	18,752
14,531	18,536
23,651	21,931
22,95	39,759
22,692	21,879
19,681	27,539
61,943	30,11
17,764	20,912
30,007	53,915

Test8

Joystick	
3D	
Full	POV
28,191	14,665
15,96	18,12
16,073	21,162
17,903	19,047
32,971	26,884
32,4	19,244
19,31	22,182
26,495	41,347
19,884	20,076
40,265	34,369

Test9

POV	
2D	
Joystick	Haptic
31,204	30,152
18,752	30,709
18,536	30,249
21,931	46,323
39,759	43,862
21,879	26,49
27,539	21,311
30,11	22,839
20,912	35,733
53,915	19,22

Test10

FF	
2D	
Joystick	Haptic
19,151	30,152
12,790	30,709
14,531	30,249
23,651	46,323
22,95	43,862
22,692	26,49
19,681	21,311
61,943	22,839
17,764	35,733
30,007	19,22

Test11

POV	
3D	
Joystick	Haptic
14,665	20,362
18,12	30,604
21,162	18,915
19,047	36,117
26,884	28,878
19,244	26,777
22,182	23,326
41,347	38,728
20,076	36,614
34,369	43,173

Test12

FF	
3D	
Joystick2	Haptic2
28,191	21,723
15,96	27,788
16,073	20,734
17,903	40,1
32,971	45,601
32,4	27,274
19,31	28,499
26,495	27,827
19,884	37,436
40,265	47,057

t-Test: Two-Sample Assuming Une t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	*Variable 2*		*Variable 1*	*Variable 2*
Mean	23,7096	30,3494	Mean	24,9452	32,4039
Variance	67,94825	66,88553	Variance	70,44827	89,72396
Observatio	10	10	Observatio	10	10
Hypothesiz	0		Hypothesiz	0	
df	18		df	18	
t Stat	-1,80824		t Stat	-1,86367	
P(T<=t) one	0,043654		P(T<=t) one	0,039382	
t Critical or	1,734064		t Critical or	1,734064	
P(T<=t) twc	0,087308		P(T<=t) twc	0,078765	
t Critical tw	2,100922		t Critical tw	2,100922	

Wilcox Rank sum test:

Test 1:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and haptic.full.3D
V = 0, p-value = 0.001953
alternative hypothesis: true location shift is not equal to 0

        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -2.8031, p-value = 0.001953
alternative hypothesis: true mu is not equal to 0
```

r = -0,62679

Test 2:

```
        Wilcoxon signed rank test

data:  haptic.pov.2D and haptic.pov.3D
V = 27, p-value = 1
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.051, p-value = 1
alternative hypothesis: true mu is not equal to 0
```

r = -0,0114

Test 3:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and haptic.pov.2D
V = 15, p-value = 0.2324
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -1.2741, p-value = 0.2324
alternative hypothesis: true mu is not equal to 0
```

r = -0,2849

Test 4:

```
        Wilcoxon signed rank test

data:  haptic.full.3D and haptic.pov.3D
V = 41, p-value = 0.1934
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 1.376, p-value = 0.1934
alternative hypothesis: true mu is not equal to 0
```

r = 0,307683

Test 5:

```
        Wilcoxon signed rank test

data:  joystick.full.2D and joystick.full.3D
V = 16, p-value = 0.2754
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -1.1722, p-value = 0.2754
alternative hypothesis: true mu is not equal to 0
```

r = -0,26211

Test 6:

```
        Wilcoxon signed rank test

data:  joystick.pov.2D and joystick.pov.3D
V = 45, p-value = 0.08398
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 1.7838, p-value = 0.08398
alternative hypothesis: true mu is not equal to 0
```

r = 0,39887

Haptic	**Full**	**2D**	17,435	25,566	15,972	34,312	20,547	26,149	25,346	24,87	21,832	43,903	
		3D	21,723	27,788	20,734	40,1	45,601	27,274	28,499	27,827	37,436	47,057	
		Sign	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	max
		As(Diff)	4,288	2,222	4,762	5,788	25,054	1,125	3,153	2,957	15,604	3,154	25,054
		Rank	0,17115	0,088688	0,190069	0,231021	1	0,044903	0,125848	0,118025	0,622815	0,125888	
		Signed Rank	-0,17115	-0,08869	-0,19007	-0,23102	-1	-0,0449	-0,12585	-0,11803	-0,62281	-0,12589	

W+ 0

W- 2,718408238

r= $Z/\sqrt{N}$

Test 7:

```
        Wilcoxon signed rank test

data:  joystick.full.2D and joystick.pov.2D
V = 13, p-value = 0.1602
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -1.478, p-value = 0.1602
alternative hypothesis: true mu is not equal to 0
```

r = -0,33049

Test 8:

```
        Wilcoxon signed rank test

data:  joystick.full.3D and joystick.pov.3D
V = 30, p-value = 0.8457
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.2548, p-value = 0.8457
```

r = -0,05698

Test 9:

```
        Wilcoxon signed rank test

data:  joystick.pov.2D and haptic.pov.2D
V = 20, p-value = 0.4922
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.7645, p-value = 0.4922
alternative hypothesis: true mu is not equal to 0
```

r = -0,17095

Test 10:

```
        Wilcoxon signed rank test

data:  joystick.full.2D and haptic.full.2D
V = 15, p-value = 0.2324
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -1.2741, p-value = 0.2324
alternative hypothesis: true mu is not equal to 0
```

r = -0,2849

Test 11:

```
        Wilcoxon signed rank test

data:  joystick.pov.3D and haptic.pov.3D
V = 7, p-value = 0.03711
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -2.0896, p-value = 0.03711
alternative hypothesis: true mu is not equal to 0
```

r = -0,46725

Test 12:

```
        Wilcoxon signed rank test

data:  joystick.full.3D and haptic.full.3D
V = 7, p-value = 0.03711
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -2.0896, p-value = 0.03711
alternative hypothesis: true mu is not equal to 0
```

r = -0,46725

K.7 3D Experiment - Wilcoxon Rank Sum Test

Test1

Haptic	
Full	
2D	3D
27,312	31,369
28,817	25,634
29,11	36,036
30,964	32,01
19,747	18,761
35,668	25,636
52,669	45,951
43,203	37,262

Test2

Haptic	
POV	
2D	3D
18,772	26,198
29,595	28,058
38,223	25,335
25,765	14,628
15,047	16,957
37,542	32,28
51,065	44,878
39,979	26,641

Test3

Haptic	
2D	
Full	POV
27,312	18,772
28,817	29,595
29,11	38,223
30,964	25,765
19,747	15,047
35,668	37,542
52,669	51,065
43,203	39,979

Test4

Haptic	
3D	
Full	POV
31,369	26,198
25,634	28,058
36,036	25,335
32,01	14,628
18,761	16,957
25,636	32,28
45,951	44,878
37,262	26,641

Test5

Jostick	
Full	
2D	3D
23,04	21,891
18,563	30,325
25,462	24,346
52,911	36,261
20,032	10,145
44,681	20,205
22,947	26,951
40,707	37,295

Test6

Joystick	
POV	
2D	3D
21,552	30,578
18,822	21,554
37,614	65,268
22,3	24,413
54,252	19,222
28,063	28,394
38,238	36,857
43,113	17,32

Test7

Joystick	
2D	
Full	POV
23,04	21,552
18,563	18,822
25,462	37,614
52,911	22,3
20,032	54,252
44,681	28,063
22,947	38,238
40,707	43,113

Test8

Joystick	
3D	
Full	POV
21,891	30,578
30,325	21,554
24,346	65,268
36,261	24,413
10,145	19,222
20,205	28,394
26,951	36,857
37,295	17,32

Test9

2D	
Full	
Haptic	Joystick
27,312	23,04
28,817	18,563
29,11	25,462
30,964	52,911
19,747	20,032
35,668	44,681
52,669	22,947
43,203	40,707

Test10

3D	
Full	
Haptic	Joystick
31,369	21,891
25,634	30,325
36,036	24,346
32,01	36,261
18,761	10,145
25,636	20,205
45,951	26,951
37,262	37,295

Test11

2D	
POV	
Haptic	Joystick
18,772	21,552
29,595	18,822
38,223	37,614
25,765	22,3
15,047	54,252
37,542	28,063
51,065	38,238
39,979	43,113

Test12

3D	
POV	
Haptic	Joystick
26,198	30,578
28,058	21,554
25,335	65,268
14,628	24,413
16,957	19,222
32,28	28,394
44,878	36,857
26,641	17,32

Wilcox Rank sum test:

Test 1:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and haptic.full.3D
V = 23, p-value = 0.5469
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.7001, p-value = 0.5469
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 2:

```
        Wilcoxon signed rank test

data:  haptic.pov.2D and haptic.pov.3D
V = 29, p-value = 0.1484
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 1.5403, p-value = 0.1484
alternative hypothesis: true mu is not equal to 0
```

r = 0,385075

Test 3:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and haptic.pov.2D
V = 24, p-value = 0.4609
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.8402, p-value = 0.4609
alternative hypothesis: true mu is not equal to 0
```

r = 0,21005

Test 4:

```
        Wilcoxon signed rank test

data:  haptic.full.3D and haptic.pov.3D
V = 28, p-value = 0.1953
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 1.4003, p-value = 0.1953
alternative hypothesis: true mu is not equal to 0
```

r = 0,350075

Test 5:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and haptic.full.3D
V = 23, p-value = 0.5469
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.7001, p-value = 0.5469
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 6:

```
        Wilcoxon signed rank test

data:  joystick.pov.2D and joystick.pov.3D
V = 16, p-value = 0.8438
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.2801, p-value = 0.8438
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 7:

```
        Wilcoxon signed rank test

data:  joystick.full.2D and joystick.pov.2D
V = 15, p-value = 0.7422
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.4201, p-value = 0.7422
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 8:

```
        Wilcoxon signed rank test

data:  joystick.full.3D and joystick.pov.3D
V = 16, p-value = 0.8438
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.2801, p-value = 0.8438
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 9:

```
        Wilcoxon signed rank test

data:  haptic.full.2D and joystick.full.2D
V = 23, p-value = 0.5469
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.7001, p-value = 0.5469
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 10:

```
        Wilcoxon signed rank test

data:  haptic.full.3D and joystick.full.3D
V = 30, p-value = 0.1094
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 1.6803, p-value = 0.1094
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 11:

```
        Wilcoxon signed rank test

data:  haptic.pov.2D and joystick.pov.2D
V = 23, p-value = 0.5469
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = 0.7001, p-value = 0.5469
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025

Test 12:

```
        Wilcoxon signed rank test

data:  haptic.pov.3D and joystick.pov.3D
V = 17, p-value = 0.9453
alternative hypothesis: true location shift is not equal to 0
        Exact Wilcoxon-Signed-Rank Test

data:  y by x (neg, pos)
         stratified by block
Z = -0.14, p-value = 0.9453
alternative hypothesis: true mu is not equal to 0
```

r = 0,175025