# Modeling and Response Conditioning for Remote Perception Enhancement In Telemanipulation System

*By*


**JAYANTA KUMAR MUKHERJEE**

(Enrolment No.  ENGG12007040 20)

**Bhabha Atomic Research Centre, Mumbai 400 085**
**INDIA**


A thesis submitted to the Board of Studies in Engineering Sciences

In  partial fulfillment of  requirements

For the Degree of

**DOCTOR OF PHILOSOPHY**

*Of*

**HOMI BHABHA NATIONAL INSTITUTE, MUMBAI**

October 2015

# Homi Bhabha National Institute

## Recommendations of the Viva Voce Committee

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by **Shri Jayanta Kumar Mukherjee** entitled **"Modeling and Response Conditioning for Remote Perception Enhancement In Telemanipulation System"** and recommend that it may be accepted as fulfilling the thesis requirement for the award of Degree of Doctor of Philosophy.

Lalit Mohan Gant Gautam Kumar Dey

**Chairman** – Dr. L. M. Gantayet Prof. G. K. Dey **(Dean Academics)** **Date:**12/10/2015

BK Dutta

**Guide / Convener** – Prof. B. K. Dutta **Date:**12/10/2015

P Satyamurthy

**Member 1** – Dr. P. Satyamurthy **Date:**12/10/2015

**Member 2** – Prof. A. P. Tiwari **Date:**12/10/2015

Indra Narayan Kar

**External Examiner** – Prof. I. N. Kar **Date:**12/10/2015

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to HBNI.

I hereby certify that I have read this thesis prepared under my direction and recommend that it may be accepted as fulfilling the thesis requirement.

**Date:** 19.10.15

**Place:** TSC/HBNI

BK Dutta

Prof. B. K. Dutta

**(Guide)**

# STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under rules of the HBNI.



**Jayanta Kumar Mukherjee**

# DECLARATION

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree / diploma at this or any other Institution / University.

**Jayanta Kumar Mukherjee**

# List of Publications arising from the thesis

## Journal

### A. Published

1. J.K.Mukherjee: "A Phantom Sensory Framework for Enhancing Remote-perception in Tele-operated Systems" Sensor Journal IEEE (electronic version) Vol 14, Issue 9, Sept. 2014, page 2999-3007
   http://dx.doi.org/10.1109/JSEN.2014.2318893

2. J.K.Mukherjee: "Intelligent Transduction for Response Synthesis in Telemanipulation" International Journal on Smart Sensing and Intelligent Systems Vol. 6, No. 4, Sept. 2013

## B. Accepted for publication in Journal:

1. J.K.Mukherjee: **"**Physics Based Vicinity Emulation for Enhanced Perception in Tele-operation" Indian Science Academy – journal Springer PRAMANA
   http://link.springer.com/article/10.1007/s12043-015-0957-0

## C. Publications In International Refereed Conferences

1. J. K.Mukherjee: "Virtual Transducer for Augmented Perception" IEEE proceedings of the Sixth International Conference on Sensing Technology (ICST 2012), Kolkata India Dec 6-9, 2012, (*IEEE Explore*)

2. J. K.Mukherjee: "Synthesized Transduction for Proximity Sensing in Tele-operated Systems" IEEE proceedings of the Sixth International Conference on Sensing Technology (ICST 2012), Kolkata India December 6-9, 2012, (*IEEE Explore*)

3. J. K.Mukherjee: "A Vicinity Activation Model for Intelligent Interface in Tele-operation" IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction, Kharagpur, India, December 27-29, 2012 *(IEEE Explore)*

4. J. K. Mukherjee: "Autonomous Visualization for Mitigating Lack of Peripheral Vision in Remote Safe Teleoperation" in Springer Verlag proceedings of 23rd. International Conference on System Engineering ICSEng2014, University of Nevada, LasVegas USA, August 19-21, 2014

*This work is dedicated to my late parents*

*Sri Narendra Kumar Mukherjee*

*&*

*Smt Bhagabati Debi Mukherjee*

*Who encouraged quest for knowledge and inculcated human values in their children.*

# *Acknowledgments*

*This research could not have been done without the immense encouragement and words of advice from Dr. Srikumar Banerjee former chairman AEC, DAE . Dr. Shekhar Basu, Director BARC , encouraged the development of experimental setups, without which the work would have remained inconclusive. I am grateful to them.*

*I am thankful to the chairman Dr. L.M.Gantayet , members and the invitees of the progress review committee that supervised the doctoral work , for extensive interactions during seminars, progress reviews, general comprehensive via-voce and pre-synopsis examination. Valuable advices from Dr. P. Satyamurthy , member review committee, made the thesis more communicative. I thank him.*

*Many thanks are due to Professor B. K. Dutta, for giving time out of his busy schedule and responsibilities of Dean HBNI. I am immensely thankful to Dr. G .K. Dey, Dean Engineering Sciences, for facilitating various steps necessary for the Ph.D. degree.*

*I profusely thank the final reviewers of my thesis. I am immensely thankful to the editors & unknown reviewers of the journals 'IEEE Sensor' International Journal of Smart Sensing and Intelligent systems, Springer-Pramana and reviewers of 6$^{th}$ IEEE International conference on Sensors 2012, IEEE ICHCI 2012 and ICSEng2014-Nevada 2014 for objectively reviewing papers on various segments of the research work of this doctoral work and publishing them in peer forums.*

*I received appreciable co-operation towards setting up test and validation hardware from my colleagues Sri K.Y.V. Krishna, and my students Sri A. Wadnerkar, Sri Gaurav Baluni and Sri Gaurav Patel of the MIAS laboratory, BARC, my thanks to them.*

*My elders Shri P. K. Mukhopadhyay , shri P. K. Mukherjee and Smt. Monju Banerjee have been my source of encouragement. The research called for huge extra effort apart from my professional duties and demanded time out of my personal life deserved by my family. My daughters Miss Audrija, Miss Rituparna and my wife Jaya have been supportive and steady source of moral strength ; mere words of thanks are not enough to express my gratitude to them.*

***Jayanta Kumar Mukherjee***

# ABSTRACT

The man in loop telecontrol systems derive their versatility by harnessing human intelligence and using remote reach by robotic front ends. The telecontrol approaches' dependence on sensing is intricate as well as crucial since both 'low level interactive' and 'goal based autonomous working' of these systems depend on sensing. In autonomous working, the sensing is oriented to direct use in control loops while in interactive highly human controlled modes the sensing is human perception targeted. In general visual sensing is most relied upon.

In hot cell telemanipulation, the on-line viewing is highly constrained, causing acute difficulty to operator in executing free space to contact phase transition. The absence of peripheral view, which is main enabler of interactive motion planning, makes situations more difficult and these together force the operator to work conservatively by maintaining low speed of operation always.

In this work innovative method has been developed to assist the human operator in perceiving the pre-contact phase by haptics based effect formation and in achieving transition of free space move phase to contact phase without relying on on-line viewing. The noteworthy outcome is use of minimal online sensing and the resulting avoidance of sensor damage in radioactive work cells.

A haptic phenomenon based on simple physics emulation of fluidics has been devised based on a multiple pass cylinder. This phenomenon occurs only in parts' vicinity and is fully defined by local distance to the nearing part surface and robots own dynamics. A workspace attribute suiting high speed sensor emulation and

transduction technique for rendering of haptic phenomenon has been developed without any object shape dictated constraints on its working. The model's success in forming an effective 'haptic percept' and facilitating its incidence on the operator hand has been established. It has functional compatibility with long standoff campaign mode sensing of workspace. The basic model has been evolved to a object centric version that is compatible to a workspace model and can be implemented by an emulation of sensing at low computation cost.

A structured method for synthesizing the aforesaid effect has been developed by casting its implementation in a sensing–transduction modality that subtly imparts robot dynamics effects to the process. The work achieves a model for synthesizing dynamics sensitive response to operator action depending on remote workspace environment. The method permits easy conditioning of its response through well defined parameters of a simple multiple-pass fluidic cylinder 'MPC' and generates percept parameters for multimodal presentation to operator leading to enhanced perception. The MPC model has been successfully simplified in stages to achieve its execution-efficient implementation.

The generic sensing framework development has aimed at forming a versatile percept and achieved it through highly configurable sensor emulation ranging from single point to cluster sensors in a unified manner. It also supports the haptic perception of proximity by taking advantage of rich cross modal perception synthesis modality across haptic, auditory and synthesized viewing domains. The proximal haptic percept, conveying impending contact, has been evolved further to also support distal percept formation in synthesized view domain and mitigates the lack of peripheral view to an appreciable extent.

The innovative technique evolved in this research, works without using any prediction of robot state. Casting of the model in two parts, a scalar ‘basis drag’ and a phantom sensor framework, resulted in low computation load, a key to the approach’s success in real time function. The method works with current state data only. The fluidics based model offers full flexibility in conditioning the mediator process’s response to the operator. It avoids recourse to autonomous control, thereby leaves full control with operator and offers scalable automation friendly implementation for large industrial implementation as well as precision small application like tele-surgery.

The method’s co-working with classical bilateral position forward and force feedback type telemanipulation has been established and tested experimentally. The research has lead to development of generic method for mitigating limitations in teleworking that arise from lack of direct and peripheral vision in hot cells.

# Contents

# List of Figures

# List of Tables

# Nomenclature

| | |
|---|---|
| $A$ | Programmable gain for slave feedback $\tau_2$ |
| $A_i$ | Homogeneous transformation matrix for kinematics computations |
| $B$ | Damping factor |
| $d$ | Bypass pipe diameter |
| $DSS_{p,k}$ | Distance of sensor point $SS_{p,k}$ from rotation axis |
| $F$ | Velocity and position dependent force on piston |
| $F_{am}$ | Actuator force in master servo loop |
| $F_{as}$ | Actuator force in slave servo loop |
| $FAZ$ | Floating Analyzer Zone |
| $\mathcal{G}$ | Inter bypass Gap |
| $G$ | Programmable gain for synthesized torque $\tau_1$ |
| $IBA$ | Intermediate buffer array |
| $l$ | Bypass pipe length |
| $LAEP$ | Link Axis End Point |
| $LASP$ | Link Axis Start Point |
| $LSJ_n$ | Offset of segment from host link's parent joint |

| | |
|---|---|
| $LL$ | Link length |
| $M$ | Mass |
| $N_c$ | Number of contributing sensors |
| $OVA$ | 3D object voxel array for a single object at real pose in subspace |
| $OVA\_xoffset$ | Offset of object voxel array on x axis from workspace origin |
| $OVA\_yoffset$ | Offset of object voxel array on y axis from workspace origin |
| $OVA\_zoffset$ | Offset of object voxel array on z axis from workspace origin |
| $OVA\_xrange$ | Range of $OVA$ array along x axis |
| $OVA\_yrange$ | Range of $OVA$ array along y axis |
| $OVA\_zrange$ | Range of $OVA$ array along z axis |
| $r_p$ | Radial distance of section polygon point from link axis |
| $R$ | flow resistance |
| $\mathcal{R}$ | A point Robot |
| $RP$ | Response parameter for developing perceivable effect to operator |
| $SE$ | Shaping element |
| $SL$ | Segment length |
| $SOJ_m$ | Offset distance of segment on link from link parent joint $LJ_n$ |
| $SOJ_m(x)$ | x component of $SOJ_m$ |

| | |
|---|---|
| $SOJ_m(y)$ | y component of $SOJ_m$ |
| $SOJ_m(z)$ | z component of $SOJ_m$ |
| $SSA$ | Sensor shaping array |
| $TLJ_n$ | Torque on joint 'n' |
| $\mathrm{TTLJ}_\mathrm{n}^\mathrm{NRD}$ | Total drag torque on joint 'n' from non-orthogonal rotation |
| $TTLJ_n^{ORD}$ | Total drag toque on joint 'n' from orthogonal rotation |
| $q_i$ | Homogeneous transformation parameters for kinematic computations |
| $Q$ | Fluid discharge |
| $v$ | Piston velocity |
| $V_{(X,Y,Z)}$ | Voxel at coordinate $(x, y, z)$ of $WVA$ |
| $WVA$ | 3D voxelized workspace in Cartesian space in array form |
| $x_{index}$ | Integer array index x to $WVA$ |
| $y_{index}$ | Integer array index y to $WVA$ |
| $z_{index}$ | Integer array index z to $WVA$ |
| $\gamma_n$ | Effective flow resistance for n bypasses |
| $\lambda$ | Torque sensitivity |
| $\mu_e$ | Effective $\mu$ in memory based smoothing |
| $\mathrm{\mu_f}$ | Fluid viscosity |

| | |
|---|---|
| $\rho_n$ | Flow resistance of path n |
| $\tau_1$ | Torque from synthesized reaction |
| $\tau_2$ | Torque from BTF feedback |
| $\Psi_p$ | Angle of $r_p$ from reference direction |
| $\zeta$ | Rotation axis of upper joint causing rotation to host segment |

# Acronyms

| | |
|---|---|
| AABB | Axes Aligned Bounding Boxes |
| AJS | Active Joystick System |
| AJC | Active Joystick Controller |
| BTM | Bilateral Tele-controllers |
| B-rep | Boundary Representation |
| BV | Boundary Voxel |
| BVH | Bounding Volume Hierarchies |
| BWP | Buffer write pointer |
| CAD | Computer Aided Design |
| CADM | Computer Aided Design and Manufacturing |
| CCTV | Closed Circuit Television |
| CSG | Constructive Solid Geometry |
| DFAP | Dual Function Audio Percept |
| DOF | Degree of freedom |
| ES | Enhanced Sensor |
| FAZ | Floating analyser zone |

| | |
|---|---|
| FPC | Floating Point Computation |
| HAVP | Haptic Assisted Vicinity Percept |
| HDS | Hybrid Dynamical System |
| HIP | Haptic Interaction Point |
| HL | High limit |
| IBA | Intermediate buffer array |
| IDP | Intersection Determination Procedure |
| LL | Low limit |
| LUT | Lookup table |
| MLT | Master Loop Torque |
| MOI | Master Operator interface |
| MPC | Multiple Pass Cylinder |
| NRD | Non-orthogonal Rotation Drag |
| NURBS | Non- uniform Rational B-spline |
| OBB | Oriented Bounding boxes |
| ORD | Orthogonal Rotation Drag |
| OVA | Object Voxel Array |
| PAV | Principal approach vector |

| | |
|---|---|
| PB | Polygon Bottom |
| PBP | Polygon bottom point |
| PE | Position exchange |
| PEFF | Position exchange / force feedback |
| PF | Position forward |
| PFFB | Phantom force feed back |
| PFFF | Position forward/force feedback |
| PGD | Principal gradient direction |
| PGS | Principal gradient strength |
| PGV | Principal gradient vector |
| PP | Percept parameter |
| PT | Polygon Top |
| PTP | Polygon top point |
| RT | Range Test |
| SCTFB | Slave coordinator torque feedback |
| SDT | Surface Direction Test |
| SE | Shaping element |
| SEBO | Smallest Encapsulating Bounding Object |

| | |
|---|---|
| SEI | Slave interface |
| SESS | Smallest Encapsulating Sphere for Surface |
| SESO | Smallest Encapsulating Sphere Object |
| SFFB | Slave Force Feedback |
| SL | Segment Length |
| SLA | Sensor Local Array |
| SSA | Sensor Shaping Array |
| STH | Susceptibility to Hit |
| UMD | Unsafe motion detector |
| VADP | View Assisted Distal Percept |
| VF | Virtual fixtures |
| VIT | Vector Intersect Test |
| VVS | Virtual Viewer System |
| WVA | Workspace voxel array |

# CHAPTER -1

## INTRODUCTION

## AND

## OBJECTIVE

Telemanipulation is the interactive human control of a robotic manipulator, where the operator and the manipulator are at different locations. Telemanipulation can be used to accomplish a large variety of tasks that are too remote, highly scaled or pose hazard to direct human manipulation. It is particularly advantageous in unstructured environments where completely autonomous robotic systems cannot be used due to the limitations of artificial intelligence, limitations on in-place sensing and sensor-data interpretation. Telemanipulation has current and potential benefits in diverse applications such as space, undersea, hazardous nuclear, chemical, and biological environments, surgery, construction, mining, military and disaster management [1].

Telemanipulation in which a human operator manipulates a master robotic device, and a slave device emulates the behaviour of the master, with some form of haptic i.e. kinaesthetic force feedback to the operator are referred to as 'Bilateral Tele-Manipulation' BTM. Although haptic feedback has been used to improve performance of telemanipulated tasks, traditional telemanipulation systems are not able to provide any intelligent assistance to the human operator. This limitation is more acute in a large segment of application scenarios that pose critical limitations on sensor usage. One such example is radioactive hot-cell BTMs

Hot-cell BTMs are faced with unstructured environment and ill defined tasks like removal of components for repairing cell based equipments which are hardly amenable to automation and therefore the man-in-loop type telecontrol with operator outside a biological shield wall remain in demand .

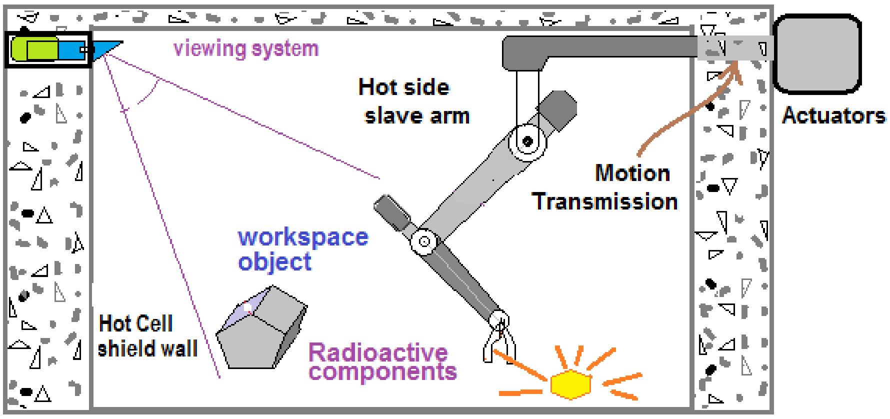



Figure 1.1 A typical slave side of bilateral telemanipulator for radioactive hotcell application with constrained viewing condition

The systems are equipped only with motion sensors like encoders coupled with servo actuators outside the cell and do not use body based sensors to avoid radiation damage.

The master and slave are similar in configuration with master construction of light nature as they are meant for command generation and force feedback on human operator hand. Remote execution of tasks is accomplished by low level mimicking of operator motion by slave side controller. The system works with operator's intelligence, who in non cell scenarios, perceives the physical workspace using visual feedback. However visual sensing is extremely limited in cell environment and covers only end-effector. Semiconductor cameras are placed away from active parts in cell at relatively safer locations with long standoff for reducing radiation damage [2].

The human vision has a large peripheral view that facilitates free space detection, motion coordination and general supportive perceptions of one's neighbourhood. The lack of such visual sensing and coverage of only end effecter in long standoff views, burdens the operator with fear of hitting work-space parts that are in reachable but not visible. This results in slow down, fatigue and reduced tele-control task throughput.

'Feeling of presence' refers to operator's feeling of being present in the remote environment. The preliminary reason for forming feel of presence arises from the belief, that presence is correlated with task performance in a positive, causal way.

In context of this scenario it is noted that in recent past long stand-off laser ranging instruments have been developed. A curious look at these suggest that through remote safe campaign mode sensing of parts in radioactive cell workspace a significant data on workspace that is stable spatial representative and usable as retained data can be generated to support run time operation. But it remains a challenge to use them effectively.

The objective here is to investigate and innovate means of mitigating lack of normal vision by eliciting synthesised response from system that is conditioned to form a perceived vicinity feel or sense of presence to operator, even without contact with workspace object. This can be viewed as 'operator aid'. It must cooperate with operator in achieving maximum free space usage in constrained space cells and for doing so enable the operator to form impact minimised contact with workspace object if desired. The perception route has to use sensing faculties other than normal vision. The implementation must function in conjunction with classical bilateral telemanipulation scheme.

The challenge lies in forming a unified proximal percept that is amenable to haptic rendering and achieving its multimodal i.e. auditory as well a visual incidences on user for integration into an unambiguous wholesome distal percept in human mind that gives a feel of presence to operator and works towards forming a unitary perceptual response. It must function as a supportive aid to the operator who remains in full control of the bilateral telemanipulation loop.

## Dissertation layout :

The Chapter 2 is devoted to literature review of published literature in related areas like classical BTM and newer haptics methods in diverse application domain. It identifies the need for an approach that is more of model assistance in nature than mediation oriented, so that controls remain with operator. It explores the development of synthesised haptics and its blending with bilateral telemanipulator. It makes a case for evolving a new technique. Techniques of object modeling have been explored.

Chapter 3 proposes a fluidics inspired model for haptic effect formation between object and robot body that can generate conditioned response to approach to objects based on speed. It is modeled in section 3.1. The model is evaluated for causing a haptic physical effect incidence on operator hand using an active joystick with force transducer (section 3.1.3). The model's effectiveness is established through an experimental setup as proof of concept in sections 3.1.3 to 3.1.9. The experiments establish model's versatility in generating perceivable haptics effects while nearing an object. The model's implementation route using the interpenetration of object, as used in past and available in published literature, has been experimented with using motion projection and the MPC action along it. The approaches based on interpenetration are visited in section 3.2. Difficult to achieve computation reduction has been identified in section 3.2.4 and 3.2.5.

A voxel based approach has been introduced in Chapter 4 to eliminate uncertain time demand faced by intersection search methods of chapter 3. A vigorous treatment of producing voxel type object model finds place in sections 4.2 and 4.3. In section 4.4 the MPC model function in voxelized space is done by motion estimation which is based on extrapolation from robot's previous location and current location. This future position estimation based approach, which is error prone, is eliminated by an innovative re-conceptualization of the MPC model in section 4.6. A new technique of modifying the object model itself is described in sections 4.6.3. MPC model simplification of viscous medium is achieved. The neighborhood grading model is improved further in section 4.7 for increasing layer width fidelity which is one of the most important factors. The conditions imposed by the basic MPC model and its implication on fluidic layers have been identified and high fidelity layer formation by fast shaping element has been devised by techniques borrowed from image processing domain. The new model's effectiveness as percept is discussed in section 4.10.

The chapter 5 is devoted to constructing a phantom sensor framework. The sensor grows from point type haptic sensor to a systematic robot body attached virtual sensor. Considerations for synthesising a percept from virtual sensor are discussed in 5.1. The treatment elaborates implementation of a phantom framework in section 5.2 and establishes its effectiveness as a co-working system to augment bilateral telemanipulation systems with feel of presence by haptic route. Its complete emulation in the MPC model activated virtual workspace has been experimentally established on a laboratory setup. Real-time sensing function has been coupled to an articulated arm and test results have been described in sections 5.2.2 and 5.2.3.

The Phantom framework is built using virtual workspace with voxelised spatial coding as detailed in Chapter 4 and assembled into a single virtual workspace, a virtual robot emulating a slave arm in the slave workspace and a subsystem to emulate MPC model for physical percept synthesis.

Section 5.4 is devoted to investigating co-working of phantom and the classical BTM. The effectiveness of MPC-Phantom system in presenting a haptic percept that is sensitive to the robot-workspace-part nearness and rate of robot approach has been demonstrated to occur prior to the object contact in workspace. The experimental results establishing phantom's success in creating the drag based percept are presented in section 5.4.6. Sections 5.5 presents phantom function as co-working 'haptic assisted vicinity percept' creator for bilateral telemanipulators of different types. Section 5.6 gives a comparison of their performance Sections 5.7 identifies a suitable sensor structuring approach and also discusses the estimation of forces acting on the virtual sensor equipped arm for different motions.

In chapter 6, two major augmentations of the generic sensor framework are undertaken to achieve Phantom version III. There are

a. Cluster sensor based Enhanced sensor like Netting sensor, slender link sensors, finger sensor and the unsafe motion sensor (section 6.4)
b. Cross-modal percept function for achieving diverse proximal as well as distal percept linkages.(sections 6.5 and 6.6)

The 'audio percept' used in aiding the HAVP has been described. The MPC model's dual interacting capability has been developed.

The cluster sensor's versatility has been demonstrated by developing a high speed impending contact direction detection in section 6.6.1 and 6.6.2. An enhanced close up synthetic viewer through special graphic attributes has been achieved in section 6.7. Finally a method of compensating lack of peripheral vision has been evolved from the MPC model generated parameters in section 6.8 and 6.9

In section 6.10, types of model error have been identified and remedial measure to mitigate error effect has been suggested in section 6.10. For model error mechanisms, reader is directed to Appendix-A where safe modelling of cell interior is treated in brief.

A comparison of the effectiveness of the three versions of phantom that progressively evolves percept in different domains from the same basic percept parameters developed by a single MPC model around a generic Cartesian space sensing of specially attributed workspace model is made in 6.11. The effectiveness of the Phantom based technique for co-working with different types of BTMs oriented to frugal on-body sensor use has been compared.

Finally in chapter 7 the thesis contribution is noted and suggestions are made for future work.

# CHAPTER 2

# LITERATURE SURVEY

Past researchers have focussed on improving different properties of a telemanipulator systems like transparency, fidelity, performance i.e. ease of work etc. The aspect of transparency is related to the capability of presenting same environment conditions faced by the slave to the operator of the master manipulator. An attribute known as ‘presence’ is supported by transparency but is beyond the functional domain of transparency as it is more evolved than the later. Feel of presence is not limited to contact impedance and attempts to make the operator aware of the slave environment. Our requirement of working without the traditional visual feedback implies that the sense of presence which normally is dominantly contributed by visual feedback has to be built through other means. Also the feel of environment itself or the functions instrumental in forming awareness of environment, need to be supported by alternative new method as well as strengthening of the adaptation of classical bilateral telemanipulation methods. By classical Bilateral Tele-controller ‘BTM’ we imply configuration that do not adapt online to the operator behaviour, the encountered remote environment or the current task.

## 2.1 Generic Framework for Improving Bilateral Telecontrol

Hannaford [3] suggested a framework for improvement of master slave bilateral control scheme (Figure 2.1). The left and right segments are on master and slave enhancement frameworks. ‘Z’, ‘e’ and ‘f’ are impedances, efforts and flows and parameters with ‘^’ are estimated values . This suggests that adaptations are possible in two segments a; on Master Operator Interface ‘MOI’, b; on Slave Environment Interface ‘SEI’ by master estimator and slave estimator that exchange efforts & impedances.

The generic measurement and estimation based approach suggested in this work [3] has been pursued further in parts by a large number of researchers. Since several performance parameters like motion or force scaling, transparency, presence and robustness can be improved by modifying control aspects to achieve enhanced tele-control over classical bilateral tele-controllers, adaptations oriented to either MOI, SEI or combination of both has been attempted. By classical Bilateral Tele-controller ‘BTM’ we imply configuration that do not adapt online to the operator behaviour, the encountered remote environment or the current task. Some of these enhancements can be of value in forming solution to our requirements indicated in chapter 1.

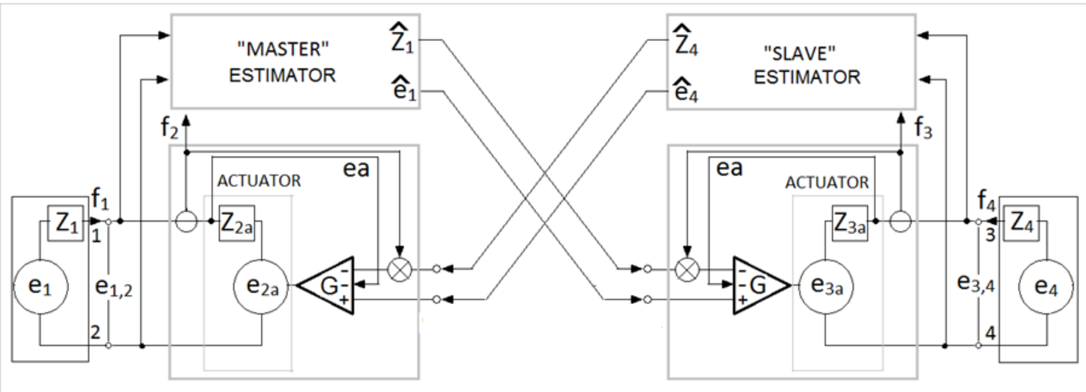



Figure 2.1 : Generic bilateral control enhancement framework suggested by Hanford

## 2.2 Feeling of presence

Forming ‘feeling of presence’ is a complex objective as it is subjective. It refers to the operator’s feeling of being present in the remote environment [4]. The preliminary reason supporting the notion of presence arises from the belief, that presence is correlated with task performance in a positive, causal way [5]. So by improving the feeling of presence, task performance is improved as well. Some studies [6] support this statement. A feasible conclusion drawn by Welch [5], says that a correlation between task performance and increased feeling of presence may depend on the work scenario and the task. For simple repeated tasks, a strong feeling of being present in the remote environment may not be needed as those tasks are done almost automatically. On the other hand for unknown tasks or in unstructured and unknown environments, a strong feeling of presence will help the operator to perform better on the task.

## 2.3 Slave Environment Interface (SEI) Related Techniques

These techniques are oriented to slave control loop adaptations for it to handle the change of control regime from position control to force control.

### 2.3.1 Impact Minimization Approaches

Much of the concern comes from the lack of environment awareness as with awareness (e.g. from visual feedback), the operator controls approach velocity. Therefore it is of interest to investigate ways of achieving the same by alternative method.

## 2.3.2 Assistance Functions for the Moment of Impact

Various researchers [7, 8, 9] have shown that robot using a stiff controller (high gain $k_p$) with force feedback often lacks asymptotic stability when going from free space to contact with a stiff object. The system may run unstable or a never-ending oscillation between free space and contact, referred as limit cycle, occurs. Since the goal, however, is to establish a constant interaction force between robot and object, the system has to be asymptotically stable with respect to a global equilibrium state even though it switches between free space and contact for a finite time. Some salient approaches are noted here.

### 2.3.2a Compliant Motion Controllers

Compliant motion controllers or a passive mechanical compliance are often used to reduce high interaction forces between robot and environment in autonomous robotic manipulation. If '$H$' is a transfer function of a spring-damper or mass-spring-damper system, and '$c$' is referred to as compliance gain then measured forces '$F$' at the end-effector may be transformed into position correction terms, $\delta x = HcF$. Modification of desired trajectory by '$\delta x$', is done so that the impact is smoothed. Depending on the measured force from the environment, the adapted slave dynamics are morphed to the biological systems inspired compliant controller. The resulting system has desired dynamics using the variable impedance controller on master side and avoids high interaction forces using the compliant controller on slave side.

### 2.3.2b Velocity Mapping Based Approach

Everett and Dubey [10] proposed a variable spatial velocity mapping, based on the estimated distance to the contact point using a laser-range finder. In their set up, velocities are sent from master to slave and that from slave to master are transformed using a varying, nonlinear Jacobian. For deriving the velocity mapping, a third-order spline is used to formulate a smooth fading from the commanded to the maximum allowable impact velocity within the deceleration distance '$d$'. During such regime of operation, the fidelity is reduced.

### 2.3.2c Position and Force Augmentation Based Approach

An approach to minimize the slave velocity at the moment of impact is presented in [11] assuming perfect knowledge about the object position. This method focuses on leaving as much control as possible with the operator. In position-force architecture, the master position enters the slave controller as reference command, while forces from the environment enter the controller on operator side. In this approach, the operator movement is estimated assuming a minimum-jerk motion with non-zero impact velocity. The estimated master trajectory is then used to reshape the approach trajectory on teleoperator site to enforce zero impact velocity. The human commanded trajectory is smoothly faded to the desired trajectory using a linear combination. A psychophysical study has shown that the objective of avoiding operator irritations and loss of feeling of presence can be achieved with this approach. However, sensor deficiencies are encountered in real applications and would need to be incorporated in the control design.

## 2.4 Master Operator Interface (MOI) Techniques

The main aim of these methods is to facilitate better task execution or an improvement in system performance by predicting human behaviour. Major challenge for controllers lies in modelling and recognizing the user's behaviour in free space motions and transportation tasks are the benefited types of tasks. In context of the present work the MOI has a role of rendering the 'haptics effects' to the operator.

## 2.5 Task Oriented techniques

These control concepts relate to the robot's function in performing task and may act on both MOI side as well as SEI side. In their approaches the intelligence of the robot recognises constraints that are formed in workspace by external methods that are oriented to performance of a single, a series of tasks or a set of tasks. As these are oriented to pursue a given goal (execution of task) these need formation again and again for different tasks.

### 2.5.1 Modified Mapping between Master and Slave

A general approach, as described by [12, 13, 14] consists in variable position or velocity mappings between master and slave devices. The direction and amount of scaling is derived based on an optimization criterion. Dubey et al. [12] present a velocity mapping for an improved task execution time in a pointing task as well as in a teleoperated docking task. A possible example is peg in hole insertion by manipulator of 'SCARA' type where after locating the end effector, motions in directions other than 'prismatic' move are scaled down from master to slave. Similar concept is used by Manocha et al. [13], they apply velocity scaling to pipe cutting such that large unwanted velocities along the pipe axis are avoided. A variable position mapping is presented in [14] with which a desired orientation can be maintained more easily and the probability of hard impact is reduced due to slower velocities near walls.

The major assumption in this approach is that the target is known exactly, and the locus from start to final goal position may be assumed to be a straight line. For application in an unknown environment, sensory information has to be used to estimate the target position. Furthermore, position or force deviations between master and slave devices can occur. The advantage is that in these approaches, the operator stays in full control of the teleoperation and is not explicitly pushed towards a certain trajectory.

## 2.5.2 Virtual Fixtures

Software based constraints known as virtual fixtures are used to reduce execution time and reduce error rates by avoiding moves into defined regions. Since the function is directly related to a task , it is task oriented and needs reworking with change of task. They either guide the operator's motion along a certain trajectory or path [15] or prevent penetration into undesired zones of the workspace [16, 17, 18]. For example its main objectives are avoidance of forbidden regions and guidance in minimally invasive surgery. For guiding virtual fixtures, on the other hand, task-relevant motions are supported, while deviations from the desired path are constrained [18]. In [19], virtual fixtures are proposed, which correct deviations from the desired path and guide the user towards the desired speed profile. For avoiding penetration into critical regions, virtual walls can be implemented. The effect of the compliance gain on task performance for different types of tasks is investigated in [15]. Higher degree of guidance is favourable for a predefined path following task, as the task performance increases considerably. On the other hand the operator should have a high influence on the slave actions such that a low level of compliance is overcome for off-path targeting or obstacle avoidance. Otherwise, the operator would have to work against the fixture, leading to reduced accuracy and increased execution time.

The main challenges for applying virtual fixtures are the choice of the right fixture, the optimal trade off between completely human-commanded and purely computer-controlled operation, and the recognition of task primitives. Larger segment of task control by computer disturbs the operator as he/she looses continuity of control over the slave robot. Their application needs experts to analyse these factors as they are task definition dependent.

### 2.5.3 Potential Field Methods

The potential field methods for navigation applications apply potential distribution on global (complete workspace) [20] basis which have influence on local potential as a varying background. Therefore they cannot be used for creating feel of presence. The 'repulsive potential from objects only' is limited by their being modeled for only convex objects of limited shapes. The technique is aimed at computational solution model for finding path or avoiding obstacles and is not suited for forming 'haptic' feedback. The 'force artifact' formation is the major detriment.

Potential fields provide a tool by which a robot can be guided autonomously towards a goal as well as allowing certain types of human inputs. Khatib [21] promoted potential fields for motion planning and obstacle avoidance. Potential fields have been applied to similar purposes for both mobile robot and multi-link robot path planning and obstacle avoidance [21, 22]. However, potential fields have problems of creating spurious local minima other than the goal. Potential functions are also limited by the shapes that can be represented as well as how closely the potential field can envelope a real object. Superquadratic potential functions, discussed by Khosla[22] alleviate this to some degree. The potential field being originating from a complex phenomena where net field at a given point is influenced by a number of originating source (even with same body) has been applied for simple objects or by enclosing them in simple but grossly approximate representation like enclosing shell of simple nature like sphere, cylinder etc.

Potential field based methods have drawbacks [23]. Simulated objects with spring action on surface too have been used earlier for force feedback [24] formation. In these approaches, broadly speaking, some form of force field (figure 2.2) exists. The methods are active in nature implying that at any location within field, forces shall

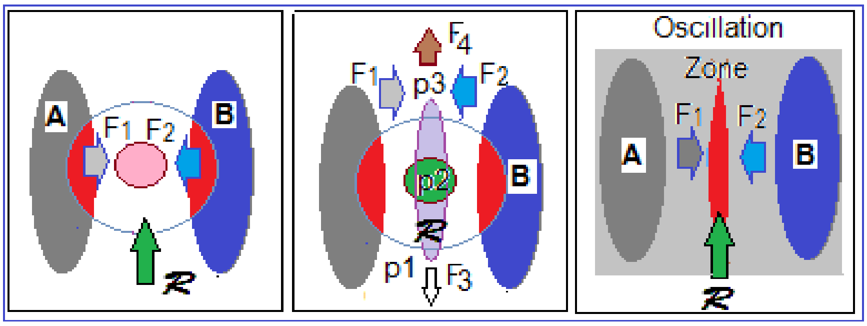



Figure 2.2: Robot moving through a force field faces complex forces $F_1$ $F_2$ $F_3$ etc.

act on a point robot irrespective of its velocity. The temporal variation in force is purely an attribute of spatial state of robot. The method is sensitive to robot position vis-a-vis the repelling object. It continues to push the master arm back out of the repulsion zone even if robot stops and attempts to maintain position in this zone.

Force feedback generated from closely placed objects or concave zones in object cause instability of control by subjecting the master arm to mutually counteracting forces. Force field is directed in nature and creates undesirable effects. For instance in figure 2.2 robot $R$ faces a push and pull effect in the periphery of force field at $p_1$ and $p_3$ respectively, leading to oscillatory reaction. Also force artifacts form at p3 and p1 that propel the robot in different directions against an operator's intention.

### 2.5.4 Model Mediated Techniques

In telecontrol system with long delays, received signals are out of phase compared to transmitted signals. Thereby, the causality between action and reaction gets lost and operators usually change their behaviour to a move-and-wait strategy. Note that

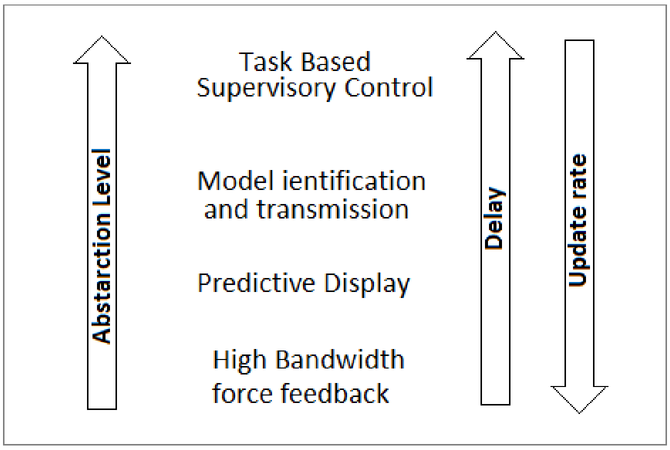



Figure 2.3: Forms of bilateral telecontrol and dependence of nature of data interchange on communication time.

model based mediation with low or no time delay pose different challenge. Virtual reality based teleoperation system, as described in [25, 26, 27, 28, 29] try to overcome this limitation by using models of several kinds . In a significant work graphically rendered object models are also used.

In a haptic teleoperation system, the incorporation of knowledge about the remote environment in the controller design can improve stability and performance. Model-mediated teleoperation adopts this idea by rendering an estimated model of the remote environment on local site instead of transmitting force/velocity flows. Thus, the user perceives locally generated forces corresponding to the estimated and transmitted model parameters

A significantly stronger feeling of perceived realism is reported in [11] when applying model-mediated teleoperation. In [20], a distance sensor is used to make this approach applicable in unknown remote environments. In both cases, the virtual forces can be provided to the operator on MOI side. For a high fidelity, the errors between virtual model and real environment have to be small, i.e. the estimation has to work properly.

For model-mediated teleoperation, the most significant improvements by model mediation is desired, if considerable time delays are present in the communication channel, as for example shown in [28]. The main problem of this approach arises from the estimation of the remote objects.

The main differences between the approaches lie in the algorithm for estimating the parameters of the environment model as well as in the updating procedure of the virtual model. Assuming some time delay in the communication channel, one way consists in generating virtual forces after the first impact between slave and object [27, 28]. The model mediation philosophy / approach are largely motivated to work in telecontrol scenarios with large delays. They use haptic models that are very simple and mostly assume a single point contact between robot and object. In work by Mitra et al. [27] concepts have been presented for a bilateral teleoperation scheme designed to operate under several seconds of delay . A hard ball fall on planar flat floor is used. The authors observe the followings-

**a**. Model complexity may need to be increased to identify contact stiffness, to allow for non-rigid interactions, as well as detecting contact surface normals, to allow operation in multiple dimensions.

**b**. For complex geometries in the slave environment, suggested approaches include method of using matched surface from known libraries to be used in realistic reaction computation for the operation under large delays. It may be noted that this does not work for no or very low delays.

These approaches succeed owing to delayed slave working whereby the master comes in virtual contact with virtual object while slave is in free space and not yet in contact with object.

## 2.6 Haptic Rendering for Virtual Contact

The fidelity of the simulated physical effect reproductions depend, among other factors, on the similarity of the behaviours of real and virtual objects. In the real world, solid objects cannot interpenetrate. Contact forces can be interpreted mathematically as constraint forces imposed by penetration constraints. Kinaesthetic feedback adds to feel of presence and has been proved to enhance task performance in applications such as virtual object assembly [30]. In particular, task completion time is shorter with kinesthetic feedback in docking operations.

Haptic rendering comprises two main tasks a: the computation of the position and orientation of the virtual probe grasped by the user; b: the computation of contact force that are fed back to the user. Some rendering methods [31] apply the position and/or orientation of the haptic device to the grasped probe. Collision detection between the grasped probe and the virtual objects is carried out and the computed collision response is applied to the grasped probe as a function of object separation or penetration depth. The synthesised contact force and/or torque are directly fed back to the user.

Any haptics based method shall need modelling of workspace object and robot body physical interaction by computational method. The success of the technique shall depend on the ease of detecting their interaction in virtual space which in turn depends on simplicity of representation of object in workspace without sacrificing fidelity of the model.

Object modelling have been pursued in several domains like Computer Aided Manufacturing 'CAM', Computer Aided Graphics and Machine Vision. A brief survey is undertaken in the following section. A major consideration should be the viability of model building by sampled data from object surface, as such possibility open route to forming model by a sensing campaign where in the sensors face harsh cell conditions (albeit mitigated to large extent). Since the purpose is to simulate interaction by use of model we first explore the interaction detection mechanism.

### 2.6.1 Mechanism of Interaction Detection

The 'virtual interaction' which is the general name used for model-model interaction, differ from real interaction in a sense that as opposed to the real interaction, penetration can occur in virtual interaction. 'The simplest query is to know whether two models touch. Some-times, it may be to find which parts (if any) touch, i.e. find their intersection or what is the minimum Euclidean distance between them? If incursion or penetration is occurring, what is the minimum translational distance required to separate them [32]? Different queries are pertinent to different applications. Distance information is useful for computing interaction forces and penalty functions in robot motion planning [33] and dynamic simulation [34, 35]. Intersection computation is important for physically-based modelling and animation systems which must know all contacts in order to compute the collision response.

## 2.6.2 Principles of Interaction Modelling

Typically, a haptic rendering algorithm is made of two parts: (a) collision detection and (b) collision response (see figure 2.4). As the user manipulates the probe of the haptic device, the new position and orientation of the haptic probe are acquired. Collisions with the virtual objects are detected and the interaction forces are computed using pre-programmed rules for collision response. These are conveyed to the user through the haptic device to provide him/her with the tactual representation of 3D objects and their surface details. User interacts with model at Haptic Interaction

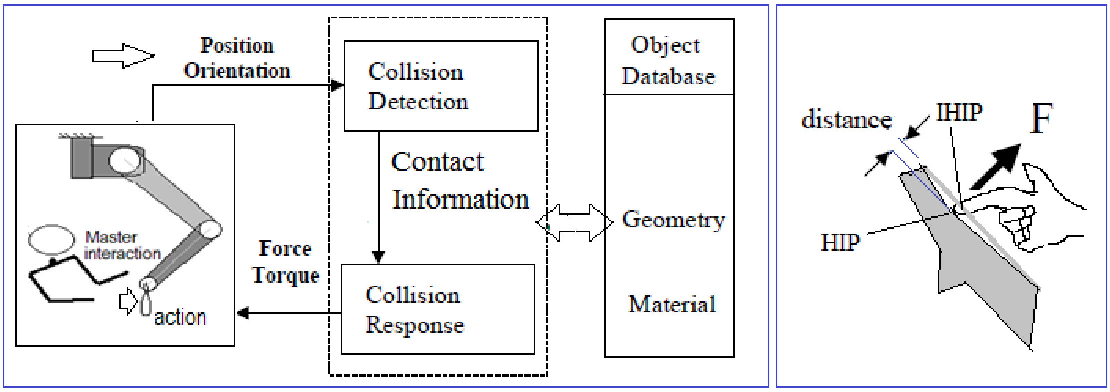



Figure 2.4: Haptic rendering of a 3D object in virtual environments. Computation modules calculate the direction and the magnitude of the reaction force

Point (HIP). As the probe freely explore the 3D space, the haptic device will be passive and will not reflect any force to the user until a contact occurs.

Since our virtual object has a finite stiffness, HIP will penetrate into the object at the contact point. Computation of the reaction force is feasible by assuming that it is proportional to the depth of penetration. Assuming no friction, the direction of this force will be along the surface normal as shown in figure 2.4.

A rigid virtual surface can be modelled as an elastic element. Then, the opposing force acting on the user during the interaction will be:

$$F = k\Delta x \tag{2.1}$$

Where, $k$ is the stiffness coefficient and $\Delta x$ is the depth of penetration.

A low stiffness coefficient would make the surface perceived soft, and vice versa. At first glance the basic recipe for haptic rendering of virtual objects seems easy to follow, but rendering of complex 3D surfaces and volumetric objects need more evolved algorithms than the one shown above. The existing techniques for haptic rendering with force display may be distinguished based on their probing object model which may be a: a point [36, 37, 38], b: a line segment [39]. However the type of interaction method used in simulations depends on the application.Using a point-based rendering technique, various classes of models, polyhedrons [36] and volumetric objects [38] have been successfully rendered.

A major compromise is on direction of reaction as the vector is aligned from HIP to nearest surface boundary. In some cases a god object [36] which tries to memorise the first contact point is used and simulated spring action between it and the HIP generates reaction outward.

### 2.6.3 Robot Spatial Position

An articulated arm, like the one of our interest, with $n$ joints will have '$n$' links. Rotation of the joints, starting from the base propagates to end effector. By this convention, joint $i$ connects link $i-1$ to link $i$. With the $i^{th}$ joint, a joint variable, denoted by $q_i$, the angle of rotation, is associated. To perform the kinematic analysis [40], a coordinate frame is attached rigidly to each link. In particular, we attach $o_i x_i y_i z_i$ to link $i$. The frame $o_0 x_0 y_0 z_0$, which is attached to the robot base, is referred to as the inertial frame. A homogeneous transformation matrix $A_i$ expresses the position and orientation of $o_i x_i y_i z_i$ with respect to $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. The matrix $A_i$ is not constant, but varies as the robot moves and its configuration changes. However, the assumption that all joints are either revolute or prismatic means that $A_i$ is a function of only a single joint variable, namely $q_i$. In other words, $A_i = A_i(q_i)$. Then the position and orientation of the end-effecter in the inertial frame are given by:

$$H = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} = T_{0n} = A_1(q_1) \cdot A_2(q_2) \cdots A_n(q_n) \qquad (2.2)$$

## 2.7 An Overview of Object Model Representation in Workspace

Feeling of presence in non-visual domain has to occur in proximity of object for it to have accurate contextual relationship. Therefore model accuracy is important. In hot cell applications, the interior of cell have machined components and well defined fixtures which are highly amenable to modelling as these are made using CAD-CAM routes and so have their CAD models readily available for modelling the workspace inside cell. Several parts of the cell have added components that are not well documented and in-situ reverse engineering approaches employing laser range data are used to construct model from object. The above two requirements need vastly different modelling routes. The modelling approach also needs to take the model usage into account. The advancement in modelling has been driven by graphics presentation of model for real appearance synthesis. These models need different data like texture, colour and several other optical properties.

## 2.7.1 Part Models for Workspace Modelling

As opposed to the aforementioned modelling approaches, the workspace modelling needs only space occupancy model which can be achieved by either solid model or bounding surface based models. Primary entity to be modelled is a surface as boundary between body and free space. Types of part modules appear in figure 2.5.

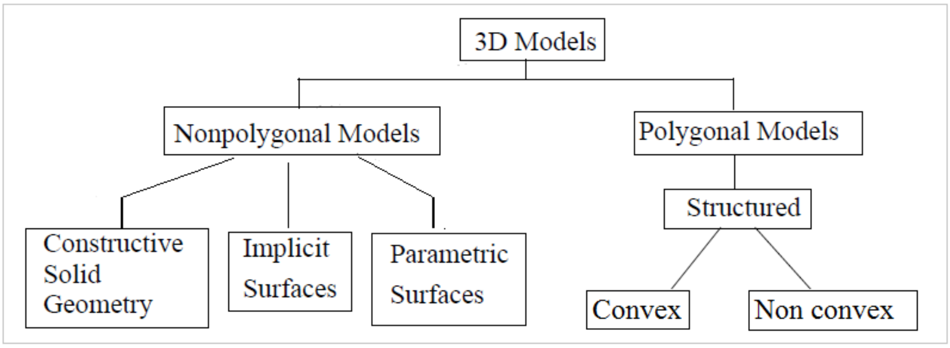



Figure 2.5: Types of part models

### 2.7.1a Model Properties:

Despite the different application contexts of object models in vision and graphics, some criteria apply to representations regardless of domain. In an early survey on object representation, Brown [41] lists ambiguity, conciseness, and uniqueness as some properties of object representation. *Ambiguity* measures the representation's inability to completely define the object in the model space; since converse is desired the measure *completeness* of the model is better measure. *Conciseness* represents how efficiently (compactly) the description defines the object. Finally, *uniqueness* is used to measure if there is more than one way to represent the same object given the construction methods of the representation. If the representation is unambiguous and unique then there is a one-to-one mapping from the object to the representation. The importance of these properties to the object representation strategy depends on the application context. In the case of object recognition applications, completeness and compactness are often sacrificed in favour of efficiency [42]. The pragmatic issue of performance often makes such compromises appropriate. This highlights the application dependence on the use of *complete* vs *discriminatory* models. Discriminatory models are most often used in object recognition. These representations are designed to be efficient for the task of matching, and not as a complete description of an object's geometry. By contrast, some computer graphics applications require complete models to accurately render a realistic synthetic image of the objects described.

#### 2.7.1b Wide Range of Representations

Generally the shapes of manufactured objects are easy to define geometrically as they are manufactured using well defined process like turning, extrusion and joining. These uses primarily following types of surfaces:

- Planar surfaces such as facets of cube object.

- Regular (or canonical) surfaces that include surfaces of revolution such as cylinders, cones, spheres, and torus, and ruled surfaces (linear in one direction) such as surfaces of extrusion.
- Freeform surfaces that allow more complex shapes to be represented via freeform surface modelling.

## 2.7.2 Mathematical Forms

Very compact mathematical representation of surfaces have evolved over the time, salient among them use following surface representations.

### 2.7.2a Implicit Surfaces:

Implicit surfaces are defined using implicit functions. They are defined with mappings from space to the real numbers, and the implicit surfaces are the loci of points where $f(x, y, z) = 0$. Such a function defines unambiguously what is inside the model, $f(x, y, z) < 0$; and what is outside, $f(x, y, z) > 0$. Consequently, implicit surfaces are generically closed manifolds, a desirable property.

### 2.7.2b Algebraic Implicit Surfaces:

If the function is polynomial in $x$, $y$, and $z$, then it is called algebraic, which includes the algebraic surfaces [43]. Implicit are also often used as the primitives in Constructive Solid Geometry 'CSG' systems. A special case of algebraic surfaces are the quadrics, which are the second-degree polynomials in $x$, $y$, and $z$. These can represent slabs, cones, spheres, and cylinders in a unified framework. A number of specialized algorithms have been developed for intersection computations between quadrics [44].

## 2.7.3 Free Form Surfaces

Definitions of free-form surfaces and objects are often intuitive rather than formal. “sculpted,” “free-flowing,” “piecewise-smooth,” are some synonyms of an object whose surfaces are not of a more easily recognized class such as planar or natural quadric surfaces. Hence, a *free-form object* is often assumed to be composed of one or more non-planar, non-quadric *surfaces.* A characterization was provided by Besl [45]: “a free-form surface has a well defined surface normal that is continuous almost everywhere except at vertices, edges and cusps”. Car bodies, ship hulls, airplanes and terrain maps are typical examples of free-form objects.

The representations discussed here are *complete* in that the geometric description is explicit, the entirety of a surface or object is described, and hence the synthetic images of the object in an arbitrary pose can be generated when the geometry description is coupled with a view specification.

### 2.7.3a Parametric Forms

Parametric surfaces are widely used in computer aided design and manufacturing (CADM) and other object modelling systems as they are mathematically complete, can be easily sampled. They support design by use of patches that can be joined as the patches’ continuity can be controlled at the joins. They pose representational power as they can be used to represent complex object geometries. Methods for generating parametric representations for data points taken from existing objects have been developed [46].

A generic parametric form for a 3D surface (a 2D manifold embedded in 3D) is

$$S(u,v) = \begin{bmatrix} x = f(u,v) \\ y = g(u,v) \\ z = h(u,v) \end{bmatrix} \tag{2.3}$$

The three functions $f(u,v), g(u,v)$ and $h(u,v)$ have as arguments the two parametric variables $(u,v)$. Without loss of generality the domain of $(u,v)$ can be restricted to be the unit square $[0,1] \times [0,1]$.

**Trimmed Nurbs Models**

Non-Uniform Rational B-Spline (NURBS) surfaces are highly compact and yet very expressive as a representation for modelling. A NURBS surface is a bivariate vector-valued piece-wise rational function of the form

$$S(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} P_{i,j} w_{i,j} B_{j,k_v}(v) B_{i,k_u}(u)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j} B_{j,k_v}(v) B_{i,k_u}(u)} \tag{2.4}$$

Where the $P_{i,j}$ form the control mesh, the $w_{i,j}$ are the weights, and the $B_{j,k_u}$ and $B_{j,k_v}$ are the basis functions defined on the knot vectors $u$ and $v$ for a surface of order $k_u$ in the $u$ direction and $k_v$ in the $v$ direction.

The various properties of a NURBS surface, including a local convex hull property, and the ability to evaluate surface points, normals and tangents along with its intuitive control characteristics make it a good choice for modelling and design.

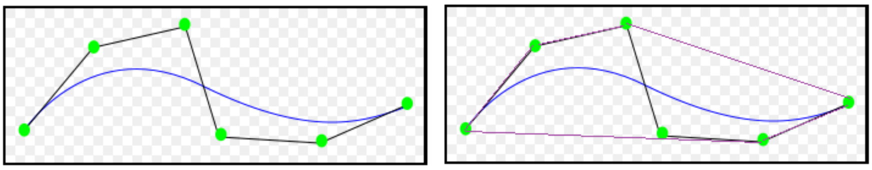


Figure 2.6: A 2D curve defined by NURB method and the convex hull of control points that contains the curve.

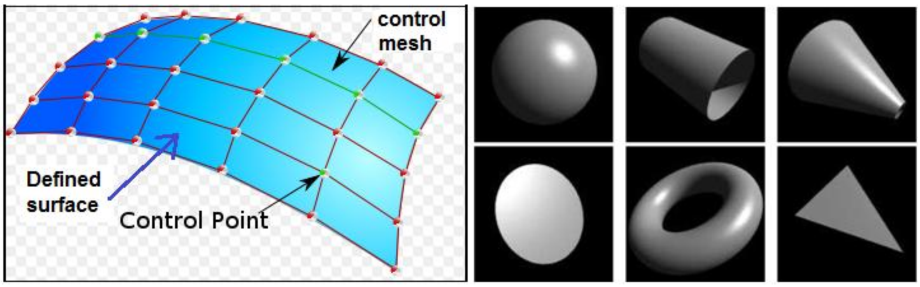



Figure 2.7 (Left): A Curved NURBS patch depicted as a mesh and as a shaded surface. The control polyhedron is visible Right: Quadric surfaces rendered by NURBS

It has been shown that natural quadrics (such as spheres, cylinders, and cones) admit exact representation as NURBS; Unconstrained interval and homogeneous coordinate makes this representation quite flexible [47]. Figure 2.7 shows a curved NURBS patch as a parametric mesh and a shaded surface, including the positions of vertices in its control point mesh.

This common descriptive form has been considered for use in computer vision systems, but Besl [45] explains why parametric representations are not widely used. In particular, it is difficult to make a surface defined on a parametric rectangle fit an arbitrary region on the surface of an object; this necessitates the use of trimming curves, which are not always unique and have to be maintained as a separate set of 3D vectors. Moreover, the homogeneous control points are also neither easily detectable nor unique. The completeness of parametric forms makes them useful as a source of an initial object specification, from which a polygonal mesh or other representations can be generated.

### 2.7.3b Generalized Cylinders

Generalized cylinders are defined by a space curve $A(s)$ (represents the axis of the primitive) and a cross-section contour $C(s, \theta)$ defined in the plane normal to the axis at s and defining the boundary of the primitive along the axis [48, 49]. Figure 2.8 shows an example of a free-form object represented by a generalized cylinder. To construct more complex objects (e.g., hierarchies such as animals), multiple generalized cylinders are used to represent their individual parts.

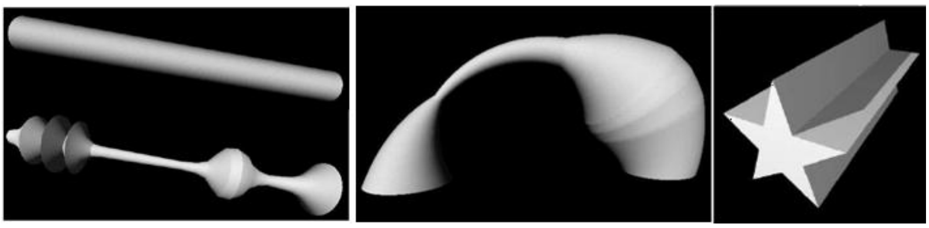


Figure 2.8: A generalised cylinder formed by varying diameter (left) along linear axis, (middle) along semicircle, A polygon as basis curve for extrusion (right).

Generalized cylinders are particularly attractive for representing elongated shapes where an axis is easy to define. In this case the axis of the primitive often provides an intuitive method to conceptualize the design of a object . In many cases, it may be impossible to define an axis whose cross sections contain only one closed contour.

### 2.7.3c Polygonal Meshes

A popular representation for 3D objects is the polygonal mesh.A shared vertex-list notation is common for such representations. Accordingly, an object is defined by a pair of ordered lists

$$\mathcal{O} = (\mathcal{P}, \mathcal{V}) \tag{2.5}$$

where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{Nv}\}$ is a list of $N_v$ three-dimensional vertices $\mathbf{v}i = (x_i, y_i, z_i)^T$, and $\mathcal{P}=\{p_1, \ldots p_{Np}\}$is a list of polygons, each specified as a list of vertex indices: $\mathbf{p}_i=\{\boldsymbol{v}_{i,1},\ldots, \boldsymbol{v}_{i,nvi}\}$.

If $nv_i$ =3 for all $i$, the mesh consists strictly of triangles. The guaranteed convexity of triangles allows simpler rendering algorithms to be used for the generation of synthetic images of models [50]. A variety of techniques (commonly called *polygonization* methods) exists for generating polygonal mesh approximations from other geometric primitives such as implicit surfaces [51] and parametric surfaces [52]. Polygonal meshes have a long history in computer graphics, but have also become increasingly popular as an object representation in computer vision. This increase in popularity is due to several factors including advances in computer storage capacity and processing power and a modest increase in the popularity of dense range sensors, which produce rectangular arrays of 3D points that can easily be triangulated into meshes. Meshes can faithfully approximate complex free-form objects to any desired accuracy given sufficient space to store the representation. While they are a faithful representation, they are also approximate and scale-dependent. Any higher-level surface characterization must be explicitly maintained along with the mesh.

### 2.7.4 Constructive Solid Geometry

Constructive Solid Geometry or CSG forms objects from primitives such as blocks, spheres, cylinders, cones, and tori, by combining them with set theoretic operations such as union, intersection, and set difference [53, 54]. One strength of the CSG representation is that it enables an intuitive design process of building shapes by means of cutting (intersection and set difference) and joining (union) simple shapes to form more complex ones. Figure 2.9 shows an object for which CSG is (probably) the only sensible modelling technique. Furthermore, an accurate boundary

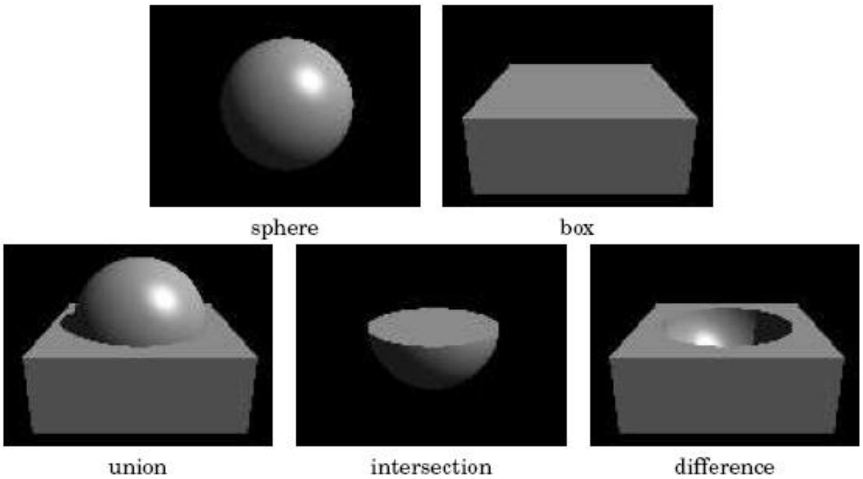



Figure 2.9: An object for which CSG is (probably) the only sensible modelling technique.

or surface representation, useful for rendering or interference computations, can be hard to compute directly from CSG representations [54]. It may be noted that the part models used in CSG can have surfaces built by any of the techniques described in earlier sections.

## 2.7.5 Articulated Objects

These belong to dynamic object class. In many cases dynamic objects can be decomposed into a set of rigid parts connected together by joints. This mechanical perspective on both machines and biological forms can closely approximate a large number of dynamic objects. For biological forms whose support structure (bone, cartilage, and exoskeleton) is rigid, the primary contributor to change in the shape and appearance of the object is the relative motion at the joints between parts of the form (body, limbs, or appendages). Articulated objects are typically studied by identifying the rigid parts and parameterizing the relationships between them. As mentioned in the previous sections, generalized cylinders have been used to recover and identify parts of articulated objects.

**Table 2.1:** 3D Object representations and their properties

| Types of representation | Complete | Local Control | Compact | Easily sampled | Easily fit | Global |
|---|---|---|---|---|---|---|
| Implicit | Yes | No | Yes | No | Yes | Yes |
| Parametric | Yes | Yes | Yes | Yes | No | Yes |
| Gen. cylinder | Yes | No | Yes | Yes | No | Yes |
| Mesh | Yes | Yes | No | Yes | Yes | No |

Table 2.1 summarizes some important aspects of 3D modelling technique as discussed above. Each strategy is characterized in terms of the accessibility of local shape information, compactness, controllability, completeness, ease of sampling, and ease of construction from sampled data by fitting in overall descriptor. A model that can be reoriented and scaled has global scope of usage.

# 2.8 Computations in Model Interaction

The requirement of model interaction arises from our aim to simulate haptics using interpenetration of objects.

## 2.8.1 Polygon-Polygon Interaction

The goal of a haptic rendering system is to generate forces that can be applied to a user's hand or arm to accurately produce a sense of contact with a virtual model. These forces are called restoring forces. The restoring forces resist penetration into the virtual model and so these are calculated using a wall model simulation. These response models often have a restoring force proportional to the penetration depth [55] and in the direction of the surface normal at the contact point.

### 2.8.2 Computation Reduction Approaches

When large number of bodies exists in space zone under consideration, a number of them need to be disqualified from candidate subset as contact may not occur with them. For this purpose object representation in hierarchies, where the initial level are less accurate but definitely conservative in missing a contact condition, use very compact data defining the shell, such as box or sphere, surrounding the object fully and employ simple comparison of position data for detecting interaction. Bounding Volume Hierarchies 'BVH' have proved successful in the acceleration of collision detection [56, 57, and 58]. Assuming that an object is described by a set of triangles TR, a BVH is a tree of 'Bounding Volumes' BVs, where each BV bounds a cluster of triangles. One determining factor in the design of a BVH is the selection of the type of BV. Often there is a trade-off among the tightness of the BV (ratio of actual volume or area occupied by objects in BV to the volume or area of BV) and the cost of the collision test between two BVs that depends on the complexity of BV defining procedure. Some of the common BVs, sorted approximately according to increasing query time, are: spheres [56], Axis-Aligned Bounding Boxes 'AABB' [57], Oriented Bounding Boxes 'OBB' [58]. BVHs of rigid bodies can be computed as a pre-processing step. Bounding box building from laser range data has also received attention [59].

# 2.9 Towards Synthesised Haptics and Its Blending with BTM

Computational geometry and physics emulation based physical part interaction has been a goal of research. A major breakthrough was the advent of computer-based Cartesian control for teleoperator systems, enabling a separation of the kinematic configurations and workspace [40]. The physics emulation can be used in robot interaction simulation. This area remains a challenge as efforts have largely gone towards developing approaches focussed to standalone contact haptics simulation.

The foregoing survey suggests that integration of haptics techniques with BTM system can be attempted in a workspace centric manner wherein the presence of objects is conveyed through some percept other than conventional percept like vision. Notable aspects in this context are the following.

1) Feeling of presence in non-visual or non-auditory domain has to occur in proximity of object for it to have accurate contextual relationship.
2) Owing to requirement above, the percept used for conveying such presence has to be sensitive to object shapes much more than that desired for obstacle avoidance.
3) The absence of delay implies that synthetic reaction needs to be in tight synchronism with physical contact forming process on slave. Since the first point contact can prevent further motion of slave arm, the commonly followed haptics simulation by penetration based force rendering is inapplicable in our problem.

# 2.10 A Case for New Technique Development

The overview in previous sections suggest that model mediation based approaches can be helpful in improving teleoperation by giving feedback ahead of contact

However the modalities of both of these are oriented to contact with body. Further a number of sensing approaches lack robustness. For example; the Laser range sensing referred here [12, 25] cannot be applied to multi-DOF operations. Same has been observed in [27] section 2.5.4, about model mediation. Limitations of the virtual fixtures and potential field approaches indicated in sections 2.5.2 and 2.5.3 are impediments in their versatile applications as generic solutions. The application of model mediation (haptic domain), in cases without delay between master and slave, is challenging as normally 'feeling of presence ahead of contact' isn't feasible. Predicted future robot state will be required to create such effect. The future state prediction calls for MOI based master motion prediction that are based on inaccurate human arm model and arm motion models. Furthermore such models have to be learnt 'on-line' as different operators may use the system at different time. Even the single operator behaviour too can largely vary owing to different reasons. There is a need for new approach development that attempts at forming an innovative framework for interworking of some of the concepts of sensing and model mediation and thereby support the operator with enhanced awareness of the remote environment. A holistic attempt encompassing the different domains, referred in section 2.5, can throw-up interworking related intricate aspects. Their treatment can lead to generic solution discovery.

Open literature have no noticeable reported work on forming a real-time robot dynamics sensitive perception that acts in the localized near vicinity region of environment objects using synthesized haptics and discourage the operator by rendering a passive percept in form of a yielding-opposition to his/her action but effectively leaves control with him/her.

The synthesized haptics poses challenges in-

a. Coupled working with bilateral classical telemanipulator, as the two work differently. The real arm doesn't allow interpenetration that is required by the haptics model for its working.
b. Rendering the effect of the synthetic haptics, as it has to share perception formation function with the feedback effects of the bilateral telemanipulator.

The cell based application being very specific, methods for haptic effect creation do not receive focussed efforts and attention to the inherent necessity of the methods being adaptive to sampled data based object model building, or assembly of the workspace with such models is less than adequate. Of particular interest is to constrain small errors from causing large erroneous rendering of haptics effects.

### 2.10.1 Model Requirement

**a.** The model should aim at creating a percept to the operator of telemanipulator for inducing some behaviour that can help or support a procedure of the normal work intended and pursued by the operator without taking away control from him/her. In recent times research has indicated that haptic and cross-modal methods have advantage [60] and the modelling approach must facilitate such cross modal function.

**b.** A major consideration should be the viability of model building by sampled data from object surface, as such possibility open route to forming model by a sensing campaign wherein the sensors face harsh cell conditions (albeit integrated to large extent) but are not on-body type.

In addition the non applicability of penetration based haptics reaction computation as indicated in earlier paragraph, suggests that estimation of effects of real physical interactions that naturally starts in the form of point contacts in most of the cases of contact initiation, has to be computed at such contact initiation withstanding demand on computational accuracy of the point contact i.e. the point contact positions and the physical contact direction (normal to the common tangent plane at the contact point). This may be unrealistic in real time applications. Further complications arise if multiple contacts occur as resultant reaction computation needs more computations and may be highly erroneous. Innovative method needs to be developed for circumventing the problem.

## 2.11 Motivation

The foregoing survey shows that the classical BTM control offers generic control solution with full control in the hands of the operator and is most versatile in nature resulting in its being favoured for unstructured and unplanned work needing human intelligence to solve a problem as it enables him to fully control the slave. The attributes like “presence”, if improved beyond that available in conventional BTM through haptic route by enhanced perceptual support, shall form a system wherein the operator feels aware of the environment and can himself modify actions to adapt to the operating regime between free space move phase and contact phase. This operator oriented support concept is a promising ‘aid’ for the operator of the classical BTM system.

Given the scenario, one is motivated to investigate and innovate means of forming a perceived vicinity feel or sense of presence, even without contact with workspace object. The ‘operator aid’ must cooperate with operator in achieving maximum control by the operator and assist the operator to form impact minimised contact with workspace object. The perception route has to mostly use sensing faculties other than direct vision. The implementation must function in conjunction with classical bilateral telemanipulation scheme.

# CHAPTER 3

## KINESTHETIC PERCEPTION OF CLOSENESS TO OBJECTS

## AND

## INNOVATIVE DYNAMICS SENSITIVE PERCEPT

The model mediated approach referred in chapter 2, employ the remote environment model to locally generate feedback effects. We intend to use a fluidics based virtual environment model as mediator model and so attempt to develop a suitable model in following sections..

# 3.1 Haptics Oriented Model Design

Our goal is to form perception of an object's presence by rendering some form of force in kinaesthetic domain to act as percept. Though notion of nearness has been used in several potential based approaches [21, 22] the objective of these approaches are to form a method of control of robot that takes it away from contact. The stress in these approaches is on forming some control law to avoid contact by treating them as obstacles [61]. We intend to develop a method for dissuading the operator to move toward object.

## 3.1.1 Fluidics Inspired Model

Consider fluid motion in a closed piston-cylinder assembly (figure 3.1). The cylinder has a side branch tube connected at tail of cylinder through which the fluid can flow back to the head end of the cylinder. The side assembly consists of one out let at.

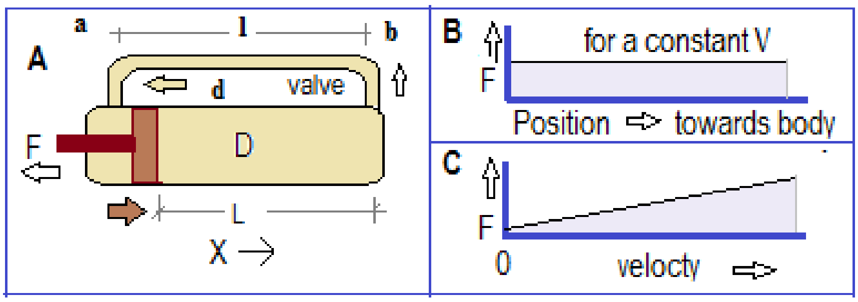



Figure 3.1: The single bypass cylinder behaviour.

head end and a single inlet at tail end. A piston pushes the fluid from head end which flows back into the cylinder through the side loop To simplify the model we made some assumptions such as no friction between cylinder wall and fluid. Further, we consider no turbulence conditions and the flow to be in laminar regime. Volume flow rate through the cylinder [62] is

$$Q = \frac{(v * \pi * D^2)}{4} \tag{3.1}$$

Where $v$ = piston velocity, $D$ = cylinder diameter

Flow resistance of pipe:

$$R = \frac{(128 * \mu_f * l)}{\pi * d^4} \tag{3.2}$$

Where $\mu_f$ = fluid viscosity, $d$ = pipe diameter, $l$ = pipe length

Force on piston:

$$F = \Delta P * A = Q^{\sim} * R \tag{3.3}$$

$$\text{Where } Q^{\sim} = \frac{(v * \pi * D^2)}{4} * \left(\frac{D}{d}\right)^2 \tag{3.4}$$

Where $\Delta P$= pressure difference between two sides of the piston, $Q'$ = volume flow rate through pipe, $R$ = Flow Resistance of the pipe, $A$ is area

$$F = v * \left(\frac{D}{d}\right)^2 * \left(\frac{128 * \mu_f * l}{\pi * d^4}\right) * \left(\frac{v * \pi * D^2}{4}\right) \tag{3.5}$$

$$F = v * C * \rho \tag{3.6}$$

Where constant $C = \left(\frac{D}{d}\right)^2$ and $\rho = \left(\frac{128 * \mu_f * l}{\pi * d^4}\right) * \left(\frac{v * \pi * D^2}{4}\right)$

$C$ and $\rho$ are functions of cylinder dimensions and fluid viscosity $\mu_f$ which are constant for a given set up, leaving the force $F$ as a function of velocity. The opposing force $F$ on cylinder varies linearly with the velocity of the piston (figure 3.1). An imaginary object with such assembly placed normal to its surface and tail end in contact, opposes motion towards it (figure 3.2). The width $W$ represents velocity of piston i.e. velocity of approach towards the body and $F$ represents the opposition force.

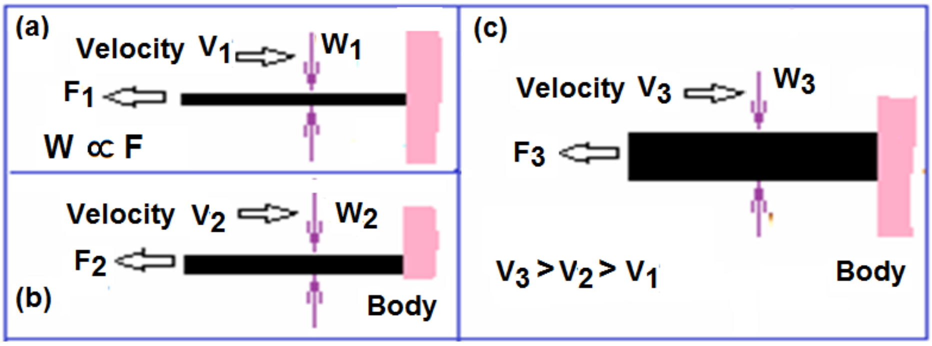



Figure 3.2: Opposing force to an approaching robot

Figures 3.2a, 3.2b and 3.2c use width ‘$W$’ of the approaching vector as depiction of required opposition force ‘$F$’ presented to an approaching robot with increasing velocity of approach.

### 3.1.1a Properties of the Model

The model generates more opposition to movement at higher velocity. The model dissuades the piston if approaching at high speed but conversely allows it to reach the body at very low speed. This ensures reaching without high impact to the object. Its function is same along full length ‘$L$’. A model that can simultaneously make the opposing force ‘$F$’ dependant on nearness to tail end can also serve the additional purpose of developing a kinesthetic feel of nearness to end for the pusher.

### 3.1.1b Creating Position Sensitivity in Model

Consider a cylinder with modified return path as depicted in figure 3.3. Major part of the return is of large diameter. Here $n$ segments of small diameter '$d$' and length '$l$' are connected in parallel. Thus $n$ paths with flow resistance $\rho_1$ to $\rho_n$ act as parallel

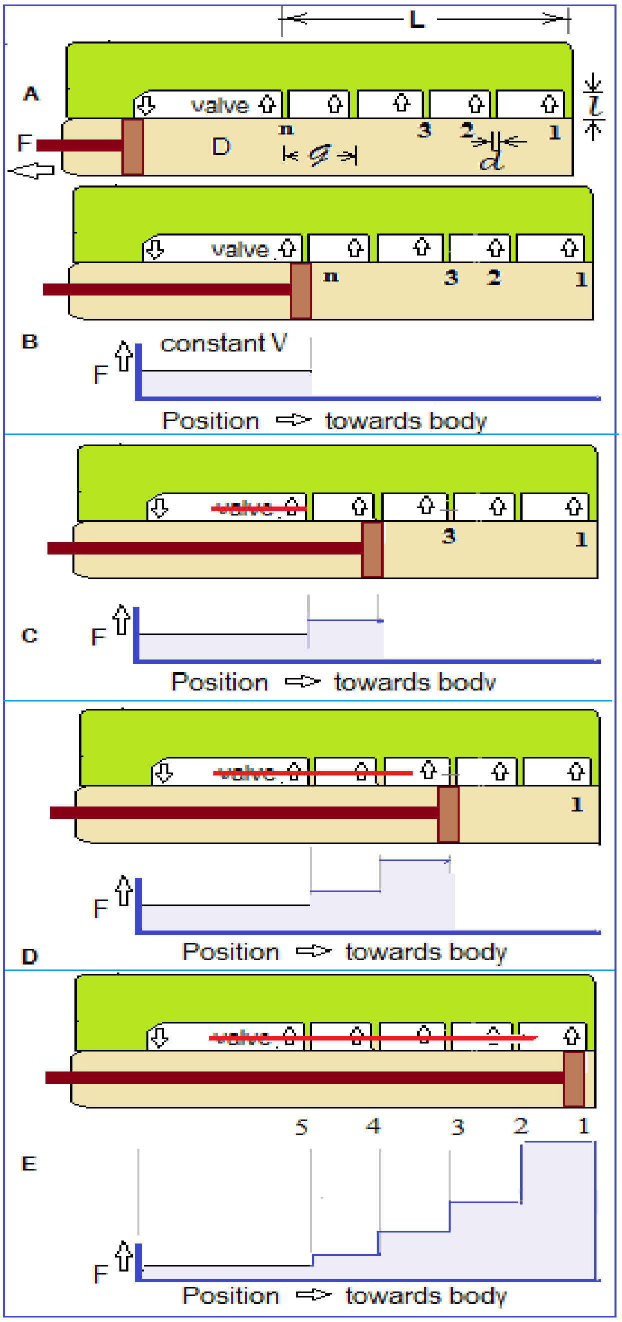



Figure 3.3: Force behaviour in Multiple Pass Cylinder – 'MPC'

return paths. Remaining part of loopback path has very large cross section compared to these and hence has ‘negligible resistance’ to flow. For a constant velocity ‘$V$’ of piston, ‘$F$’ is now dependent on the piston position in cylinder as available return paths change along the traverse. In case of travel from start to position 1, all n parallel paths are available (refer figure 3.3a and 3.3b).

$$F = v * \left(\frac{D}{d}\right)^2 * \left(\frac{1}{\frac{1}{\rho 1}+\frac{1}{\rho 2}+\frac{1}{\rho 3}+\cdots+\frac{1}{\rho n}}\right) \tag{3.7}$$

In present example $n$ is 5. As the piston moves inward in cylinder, return paths keep falling out of loop. At point 4 (figure 3.3c ) resistance to flow is

$$\gamma_4 = \frac{1}{\frac{1}{\rho 1}+\frac{1}{\rho 2}+\frac{1}{\rho 3}+\frac{1}{\rho 4}} * C \tag{3.8}$$

At point 3 (figure 3.3(d)) the effective $R$ increases to

$$\gamma_2 = \frac{1}{\frac{1}{\rho 1}+\frac{1}{\rho 2}} * C \tag{3.9}$$

the general relation is

$$\gamma_n = \frac{1}{\frac{1}{\rho 1}+\frac{1}{\rho 2}+\frac{1}{\rho 3}+\cdots+\frac{1}{\rho n}} * C \tag{3.10}$$

figure 3.3e shows the complete behaviour of $F$ for a constant velocity $v$ of the piston.An approaching piston faces opposition force $F$ that varies with distance remaining to tail end and its velocity $v$ (figure 3.4). Since C is constant we have a general equation

$$F = v * \gamma_n \tag{3.11}$$

### 3.1.2 Model Function

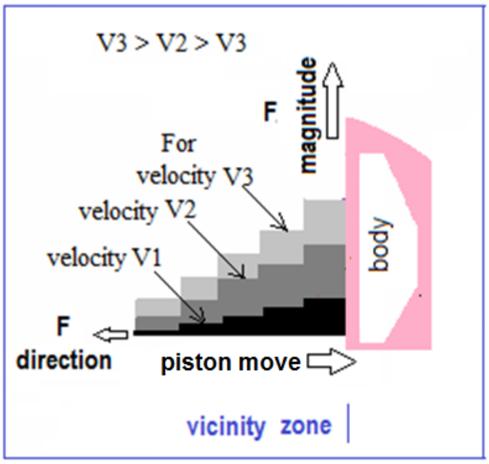



Figure 3.4: The multi-pass cylinder responds to speed as well as distance from object

The model enhances properties of single pass loop back cylinder by adding the capability to produce object vicinity based increase of $F$. A push action on piston at constant speed causes stepped increase at each bypass crossing. The effect is more enhanced for piston moving at higher velocity (figure 3.5). It retains the property of

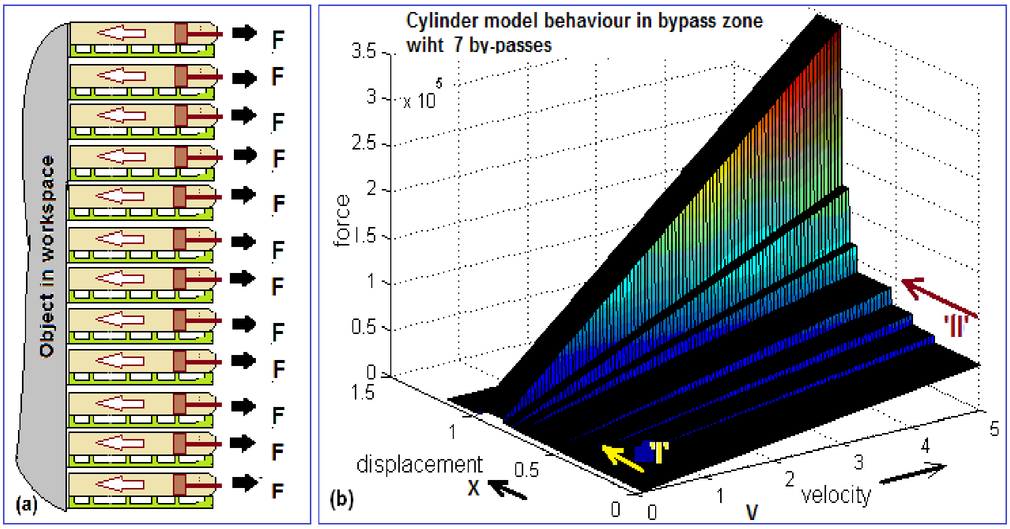



Figure 3.5: a; Conceptual piston-cylinder assembly attachment to an object's outer surface for causing reaction to approaching robot arm, b; $F$ developed by multi-pass cylinder and its dependence on piston velocity

velocity dependence of earlier version. The position dependant change in F can be used to generate perceivable increase of F with nearness to a body part while approaching it and the property can be used to ensure reaching body without high impact. We call this model Multiple Pass Cylinder or ‘MPC’

## 3.1.3 MPC Model Evaluation

The model can be imagined as an arrangement placed with the closed end of cylinder touching the specimen part and the axis of cylinder aligned with normal to the surface (figure 3.5a). A point robot, ‘$\mathcal{R}$’, approaching the specimen has to push the piston to reach the body surface. Such an arrangement achieves velocity based opposition to ‘$\mathcal{R}$’ move in neighbourhood of body. It hinders move along ‘II’ in figure 3.5b, which has high velocity, but permits approach at a very low speed, reducing risk of damage by its impact as evident for move along path ‘I’. $F$, computed by (3.7) is treated as Response Parameter ‘$R_P$’ for developing perceivable effect to operator. The stepped variations in $\gamma_n$ is preferred (3.10) by some users as the resulting sharp change in $F$ at positions 1,2,3 etc, in figure 3.5, cause easily perceivable feel of $F$. A smoothened $F$ too can be mathematically formulated by using interpolations to intermediate positions.

### 3.1.3a Force Rendering from the MPC Model

For evaluation of the model in the proposed form, figure 3.5, of use in telecontrol application, output of MPC emulation process is used with an experimental force transducer. The MPC function is formed in two phases. In pre-run phase, a desired model is formed by choosing $n$, $d$ and $l$. Resulting impedances $\gamma_n$ are held as an array (as in figure 3.6), facing a planar part in workspace. In this arrangement, $\gamma_1$ is

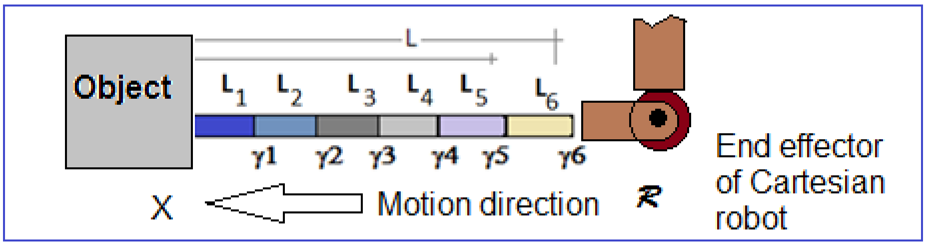



Figure 3.6: The use of $\gamma$ coding of distance in experiment and motion X convention.

the impedance closest to part surface and is assigned from the last segment of the cylinder and $\gamma_5$ is assigned 5th. from first segment (figure 3.6) and so on.

In run time the ‘$\mathcal{R}$’ position is computed from the kinematics model using the present joint status of slave arm. This ‘$\mathcal{R}$’ position is binned as per the ‘distance to object’, in workspace giving applicable ‘$L_n$’, and corresponding $\gamma_n$ value is identified and retrieved from array. The probe velocity ‘$v$’ is computed from discrete time stamped positions at time intervals or from velocity sensors on slave robot arm whose tool tip represents ‘$\mathcal{R}$’ (figure 3.6). Applicable Response Parameter ‘$R_P$’ is computed by (3.7). $R_P$ is converted in physical form by using servo motor operating in torque control mode for creating force-feel on Active Joystick (AJS) as opposition. Figure 3.7 outlines method for response parameter generation.

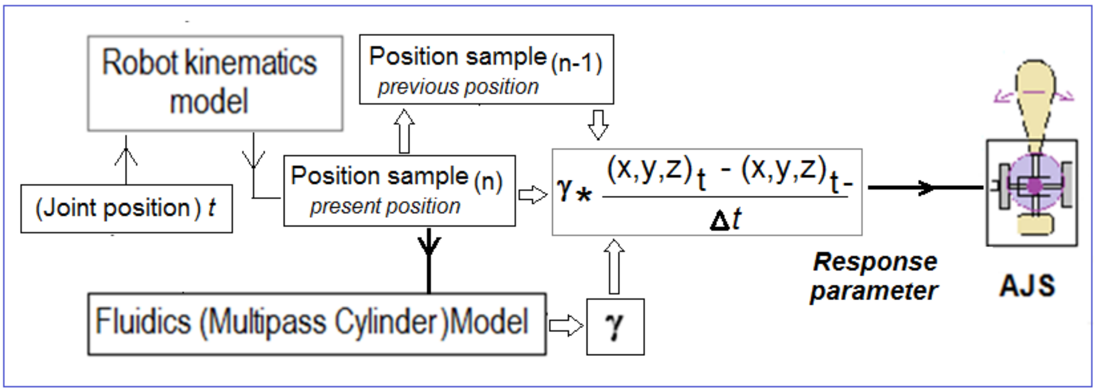



Figure 3.7: Experimental response parameter estimation for MPC model

In the evaluation experiments, the feedback is a force-feel created by an active joystick AJS, designed and developed for the experiments. AJS has 2 degrees of motion and renders force by using servomotors working in torque mode and coupled directly to the handle joint. The AJS is well balanced to appear passive without excitation of torque transducers. It has low inertia. (figure 3.8)

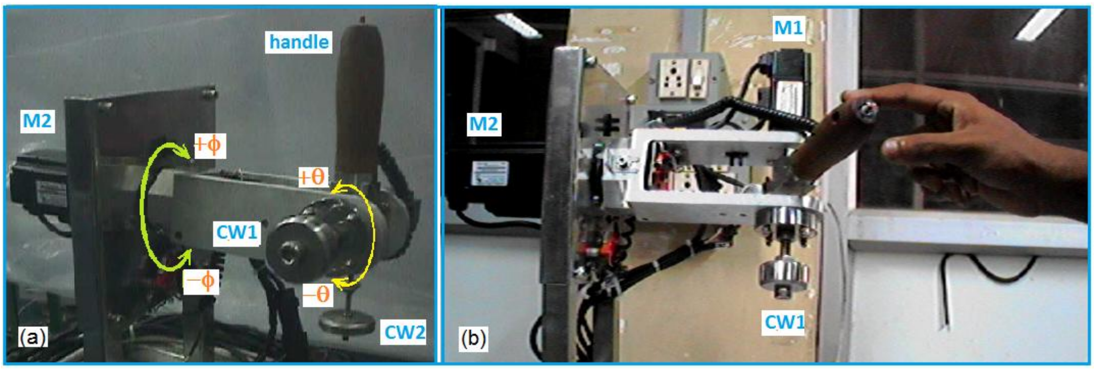



Figure 3.8( a and b): The balanced active joystick ‘AJS’ developed for validation of MPC model.

The torque demand signal vs. torque output of the motors is linear with programmable gain selection. It is dependent on the sensor output dynamic range. AJS has been formed using 100 W motor on $\theta$ shown as axis 1 and 400 W motor on $\phi$ axis shown as axis 2. The $\theta$ axis has 0.32 N-m torque capabilities up to 50 rps with torque constant 0.36N-m/A. and $\phi$ axis has torque 1.27 N-m up to 50 rps with torque constant 0.49N-m/A.

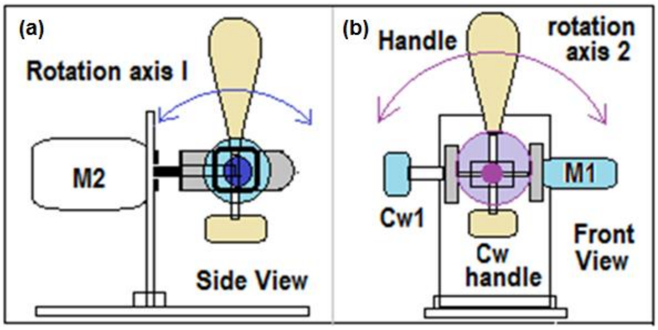



Figure 3.9(a and b): Rotation and balancing arrangement for the two axes.

### 3.1.3b Response Parameter Estimation and Rendering

The applicable $\gamma$ for the robot position is computed from the joint sensor values and robot kinematics model. The method followed is outlined in figure 3.7. For *RP* conversion to ‘force-on hand’, the servo motor and controller combination is configured in ‘torque-mode’ and operating sensitivity is designed as illustrated in figure 3.10 and figure 3.11. Note that the transducer formed by the motor and controller combination has very good linearity.

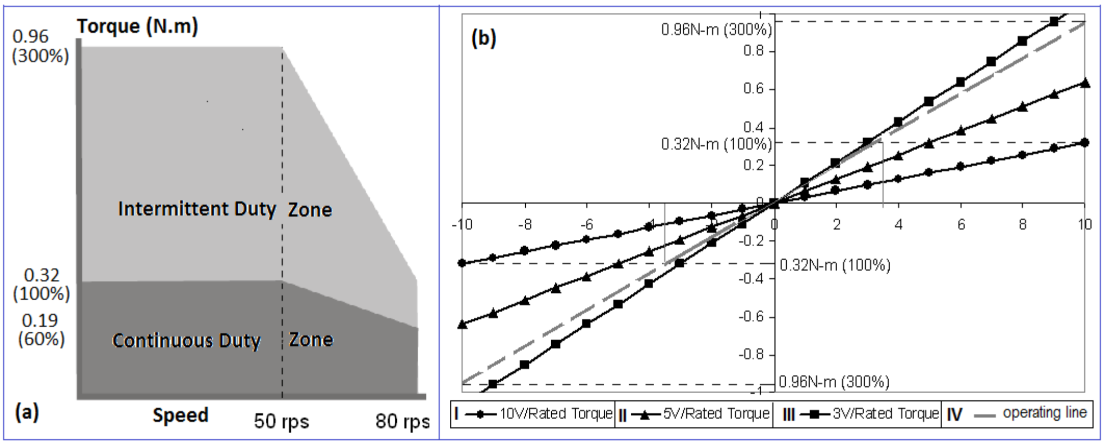



Figure. 3.10: a; $\theta$ axis command Voltage-Torque Characteristics, b; $\theta$ motor characteristics (torque control mode)

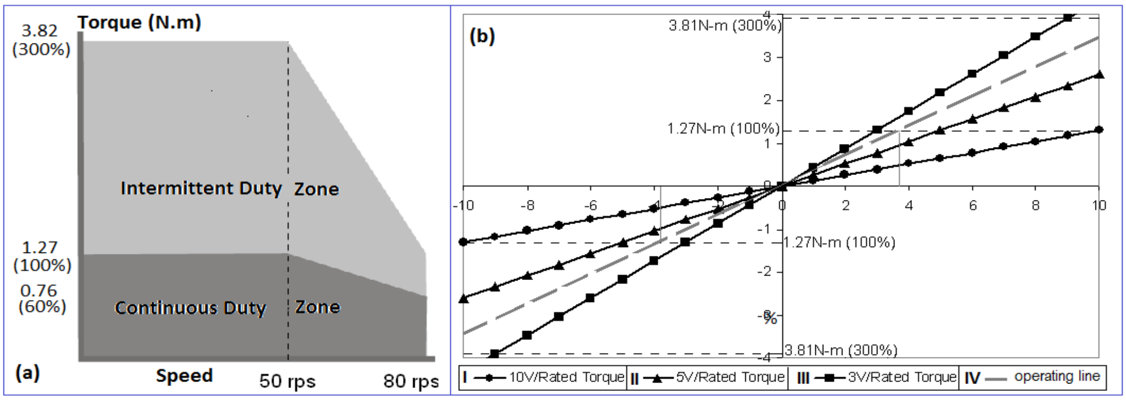



Figure. 3.11: a; $\phi$ axis command Voltage-Torque Characteristics, b; $\phi$ motor characteristics (torque control mode)

## 3.1.4 Experimental Verification of Model Performance

Experiments were carried out for differenty design of the MPC model as described below.

### 3.1.4a Experimental Setup

A 5 DOF robot's (figure 3.12) only translation stages x and y were used keeping z constant in experiments for $R_P$ generation. In the experiment , time stamping was used for unified referencing . The legend AJC represents joy-stick controller.

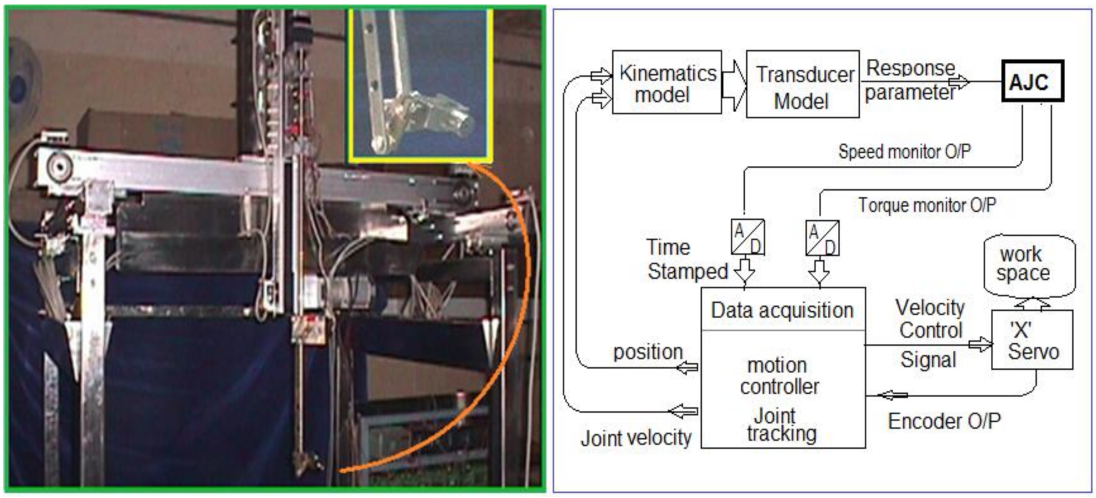



Figure 3.12 a; 5 DOF Cartesian robot experimental set-up, b; Signal flow in a single axis test setup. AJC- joystick controller.

In the experiments, slave motor (in 5DOF robot) worked on velocity control mode. Velocity control signal to the controller was formed by 14 bit accuracy D/A output and updated at 100 Hz. Shaft encoder tracking was achieved by the slave joint controller and time stamped position was acquired. For high speed approaches, the object surface was used as data coded limit and real object was removed for avoiding inadvertent damage during experiments.

## 3.1.5 MPC Model Variety

The MPC model can be differently designed by changing physical parameters of the bypasses . The interval between the bypasses plays an important role in forming different models of MPC. This is explored below.

### 3.1.5a Regular Size '$\mathcal{G}$' Based MPC Model

The probe robot arm was moved along a fixed 'Y' at different constant velocities which were sensed by the robot joint sensors. The $RP$ values generated by the model are shown in figure 3.13. $RP$, developed by system for same conditions but constant accelerations and constant deceleration are shown in figure 3.13b andfigure 3.13c.

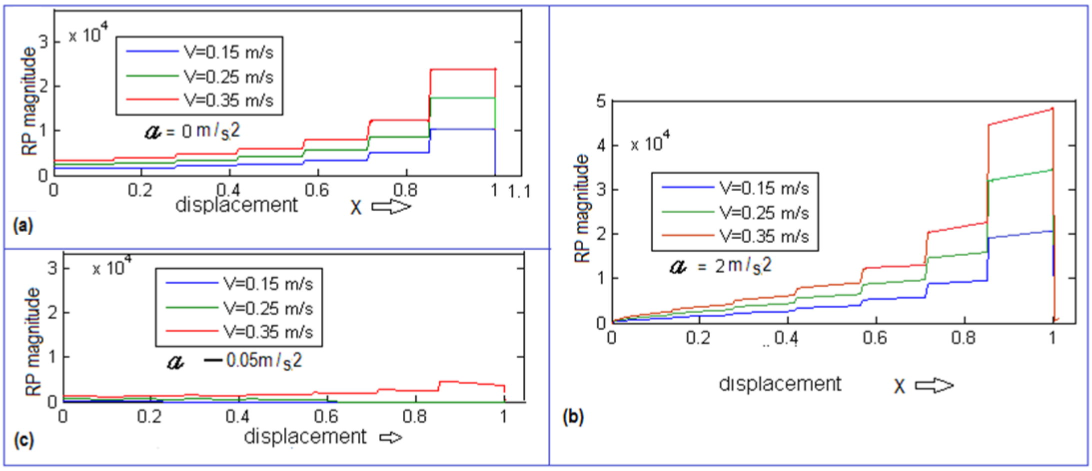



Figure 3.13: Response parameter generated by the model under controlled speed operation

### 3.1.5b Effect of Variation of gap $\mathcal{G}$ between Bypasses in MPC Model

The parameter '$\mathcal{G}$' i.e. gap (figure 3.3a) is mapped as $k$ (integer) numbers of voxel of length $V_x$. Therefore $k$ is scale factor between real workspace and model. For $k = 2$, fluid resistance $\gamma$ forms an array that appears as

$$[\gamma_1, \gamma_1, \gamma_2, \gamma_2, \gamma_3, \gamma_3, \gamma_4, \gamma_4, \cdots, \gamma_{n-1}, \gamma_{n-1}, \gamma_n, \gamma_n]$$

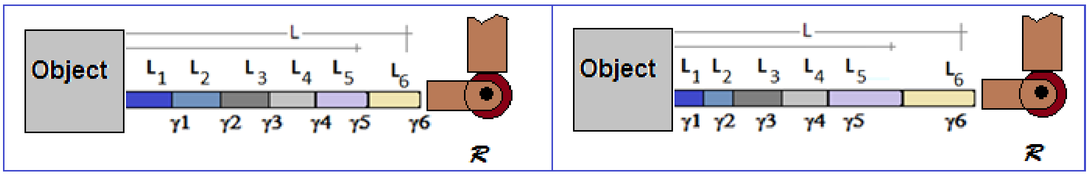



Figure 3.14: a; Coding for narrow ‘$\mathcal{G}$’, b; for unequal gap ‘$\mathcal{G}$’

profile is depicted by different colour code in figure 3.14a and is repeated for a set of $Y$ values for facilitating single axis motion along $X$ axis. If value of gap $\mathcal{G}$ is reduced then $k$ reduces. Lowest value can be $k\ = 1$. The coding is shown in figure 3.14a.

### 3.1.5c Performance of the Transducer

The servo controller of AJS motors provides instantaneous torque proportional signals at monitoring output.

$$\tau = \lambda * RP \tag{3.12}$$

Where, $\lambda$ is the torque sensitivity to control signal for AJS

For a nominal holding position at 75mm from the AJS shaft, the force appears as in figure 3.15 for ‘$\mathcal{R}$’ motion executed by controller at zero acceleration, constant acceleration and constant deceleration.

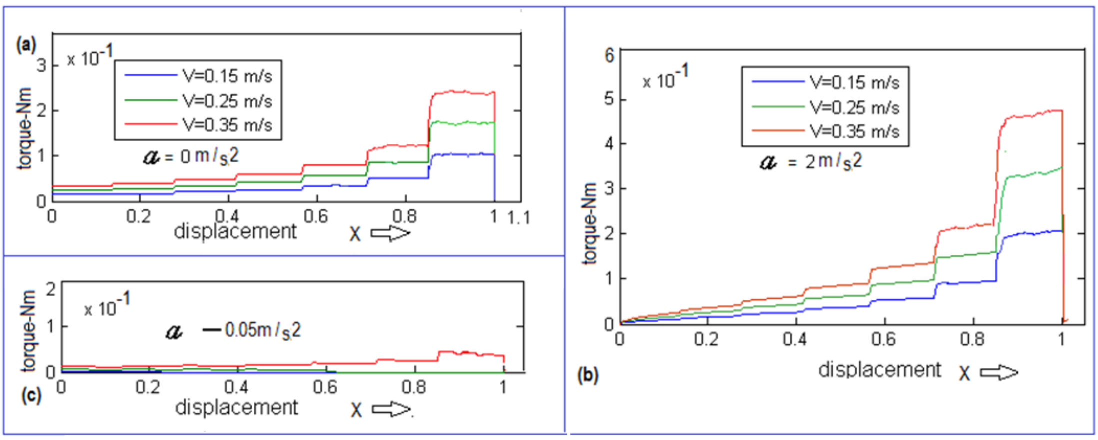



Figure 3.15: Transducer performance ($\theta$ axis) for constant $\mathcal{G}$ and fixed, accelerating and decelerating motions. In experiments, $x\ = 10cm$. (Plots are for fraction of $x$)

### 3.1.6 MPC Evaluation for Narrow Gaps and Varying Gaps

The rendered effect from unequal gap design (figure 3.14) is depicted in figure 3.16b. The advantage of narrow gap is evident in closer vicinity where a responsive operator reduces speed causing '$\mathcal{R}$' slowing down but bypasses being nearer still senses slowdown owing to frequent change of $\gamma$ rather than change of $v$ which may not be appreciable over short travel. An interesting way of enhancing closeness perception is to have gap $\mathcal{G}$ of different sizes. At periphery of vicinity $\mathcal{G}$ can be

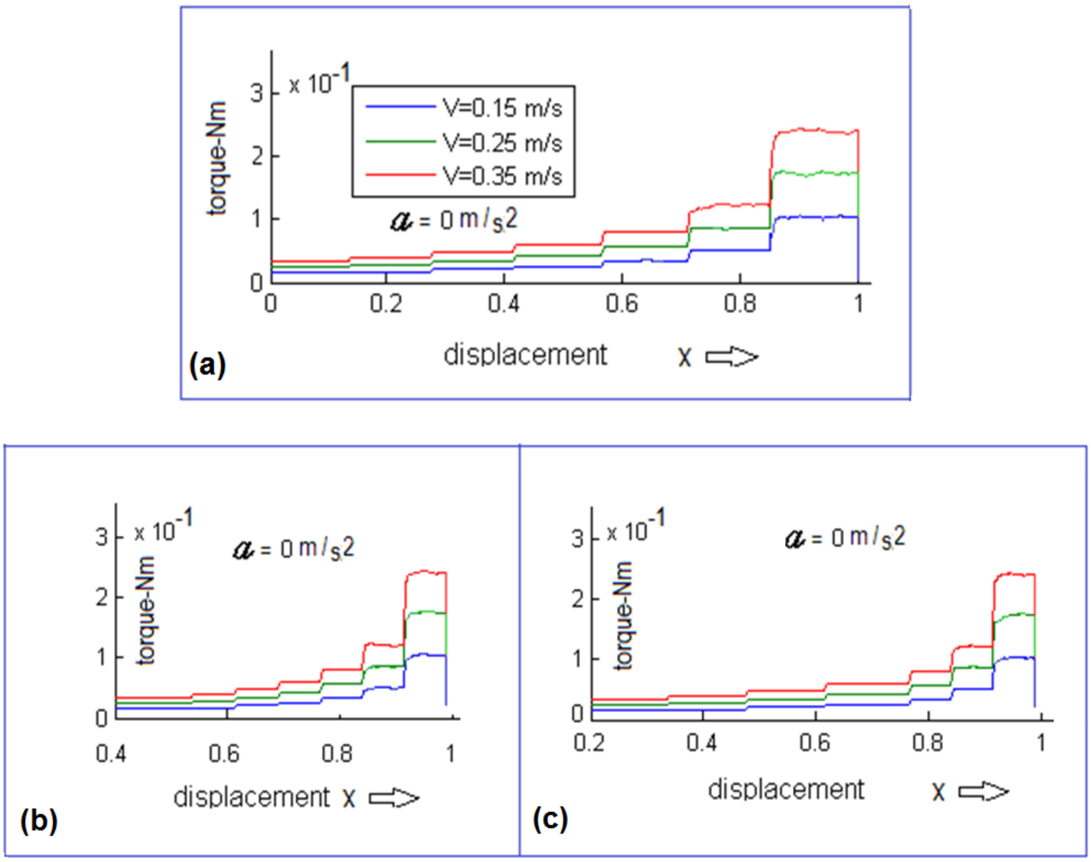



Figure 3.16: Effect of changing $G$ on force feedback, a; regular gap with $k = 2$, b; regular gap with $k = 1$, c; initially $k = 2$ and later $k = 1$. (In experiments, $x = 10cm$. Plots are for fraction of $x$)

longer and close to object it can be reduced. For such case the '$\boldsymbol{\gamma_n}$' coding for the same part is as shown in figure 3.14b. The resulting $\boldsymbol{RP}$ is shown in figure 3.16c. As '$\mathcal{R}$' approaches closer to the object the transducer produces more frequent $\boldsymbol{RP}$ change giving feel of increasing proximity by increasing frequency of step changes on AJS.

## 3.1.7 Observations from MPC Evaluation.

The AJS produces stepped force opposing operator hand which increases in magnitude with advance of '$\mathcal{R}$' towards object. For higher velocity of approach, at same relative distances from object surface it produces higher force magnitude step. On frequency scale the faster move generates increasing frequency of steps. A design with lower $G$ near cylinder end offers relatively faster step occurrence at low velocity also and enhances perception of approach to object more effectively. The $RP$ in experiments produced perceivable opposition force to operator's hand placed at nominal distance of 75 mm from rotation axes in manner expected using $M_1$ on 100% dynamic torque range. For constant speed move, generated by computer controlled servo, appreciable stepped changes in force were produced for low speeds when high torque gains were set. But for speeds above 0.3 m/sec. force was high enough for reasonably strong push back to operator. Response parameter developed here works well (Table 3.1). The variety of the response makes its rendering very effective for conveying the proximity based object presence to AJS operator. Individual's sensitivity being different, range tuning is desirable and selection of '$RP$ to force ratio' by adjusting torque gain of $M_1$or $M_2$ as per their sensitivity. Experimenters perceived the approach phenomenon in spite of not being allowed to see the robot and workspace.

**Table 3.1**: Performance of MPC driven percept rendering using AJS

| Motor | Mode | Control signal (volts) | Torque Usage | Experimental observation on force rendering |
|---|---|---|---|---|
| M1 | Vicinity at normal speed<br><br>High Velocity | 0 – 3.6 (+/-)<br><br>3.6 - 8 (+/-) | Normal demand<br><br>Overdrive 250% | Effective force feedback dependant on vicinity. AJS operation is low inertia, quick response. Transducer action very good. The perception developed is well suited for delicate manipulation. |
| M2 | Vicinity at normal speed<br><br>High Velocity | 0 – 3.8 (+/-)<br><br>3.8 - 7 (+/-) | Normal demand<br><br>Overdrive 200% | Effective force feedback dependant on vicinity with wider force range. Transducer action very good. AJS feels heavier on hand and can resist operator motion apart from only forming perception |

## 3.1.8 Observations on Model Function

The function of ‘virtual transduction’ is to create a feedback parameter for use at master end based on telerobot’s proximity to parts in workspace and its own dynamics as depicted in figure 3.5. While the basic model is virtual, the rendered effect achieved by the transduction has to be real. The function of MPC model is to create a kinaesthetic feedback parameter for use at master end based on telerobot’s proximity to workspace parts and its own dynamics. The method must mitigate abrupt feedback to the operator on surface contact, it must offer programmability of the modelled parameter in an unambiguous manner, opposition must appear in close vicinity of specimen surface and must depend on closeness to body. The functional requirements dictate that it should be suitable to real time working and must leave the slave side control unaltered.

The simple model offers following features

i. It is well suited for transduction and rendering haptic effect on a force feedback joystick.
ii. The model has sensitivity to position that is of discrete nature.
iii. A constant speed move generates distinct force steps.
iv. The force steps, nonlinearly rise in magnitude when closing in to object and are more distinct near object.
v. The step weights are amplified by speed of piston motion.
vi. The step interval on time axis is constant for a fixed speed but reduces with higher speed. In effect the step frequency increases with speed of piston motion
vii. The model offers flexibility of sizing the gap '$G$', allowing a frequency based change of step rate. A frequency modulation steps wherein a piston sees more frequent force steps closer to object when moving at a fixed speed.

# 3.2 Viability of MPC Model Application

In figure 3.17 an articulated arm is treated as example. The joint configuration is illustrated by stick model. It has $\phi$ rotation around vertical rotation axis and $\theta_1$, $\theta_2$, $\theta_3$ rotations around tilt axes that cause rotation of links. The length of the vectors formed between the points on previous state of arm and corresponding present instance of the respective points are the move vectors$P_1P_1'$, $P_2P_2'$, $\ldots P_uP_u'$.

These approximate motion directions of the points on body when extrapolated, intersect the body at $P_n$".

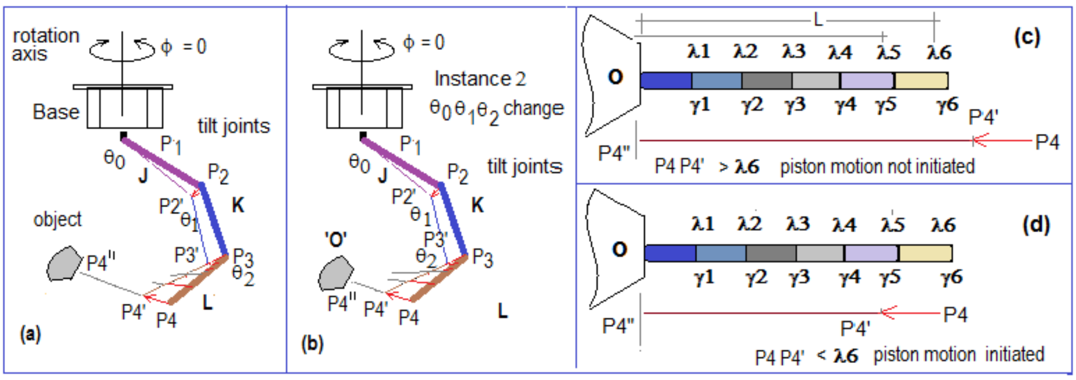



Figure 3.17: (a, b) Move vectors on various points on an articulated arm; (c, d) The MPC model on a linear approach to the object 'O'

If $P_n' P_n$" length is equal to piston stroke length along the cylinder i.e. $L$, the model action needs to be emulated . Binning of $P_n' P_n$" is done between $\lambda_6$ to $\lambda_1$ and applicable $\gamma_n$ is found from the model design based on $d, D, l, \mathcal{G}$ and $\mu_f$.

The velocity is computed by:-

$$V = \frac{|P_n P_n'|}{\Delta T} \tag{3.13}$$

Where $\Delta T$ is encoder sampling interval. The $F$ is computed by (3.7)

Therefore, the model application is primarily viable, but in the case below, $P_n P_n$' does not hit object '$O$' but grazes past it and in such case also the vicinity based drag must act on the robot. Therefore there is a need to probe the condition by using transverse vectors '$TD$' normal to $P_n' P_n$" and the distance $TD$ as shown in the diagram determines applicable $\lambda_n$ by binning $TD$ instead of $P_n' P_n$". The procedure has to be followed in all direction as shown in figure 3.18d.

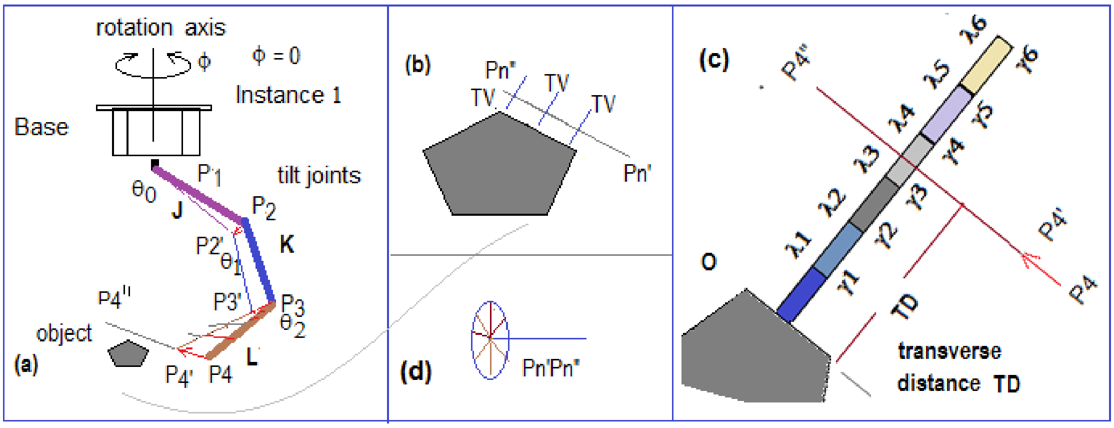



Figure 3.18: Cases needing different treatment

### 3.2.1 Probe Penetration Based Approach

Alternatively objects modelled in any manner referred earlier can be intersected by a probe vector and applicable $\gamma$ can be estimated. For planar objects and other objects approximated by planar surfaces at the point of penetration, the applicable MPC model can be estimated by projecting a vector in direction of its move with length representing the next move distance. But large number of surfaces has to be tested.

### 3.2.2 Search Scope Control

The concepts of section 2.8.2 can be used in vector - object intersection for limiting the inclusion of object or surface entities within search process [63, 64]. The method can be seen as 'Search Scope' control. It can be done by use of 'Smallest Encapsulating Sphere Object' *'SESO'* with radius '$r$' figure 3.19. It achieves reduction by excluding objects (and their component surfaces) that are farther than '$r$' from current location $P_n$ through simple test of comparing object centre distance from $P_n$.

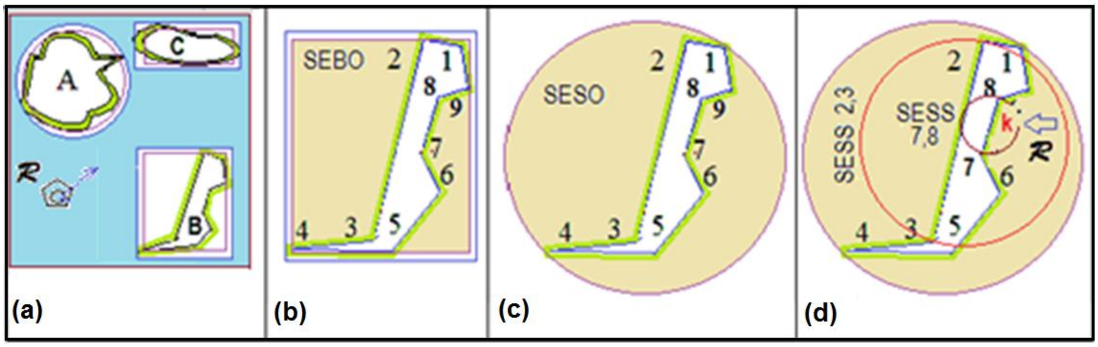



Figure 3.19: a; Workspace in 2D, b; smallest bounding box , c; bounding sphere; and d; SESS for surface 2,3 and for 7,8 .

### 3.2.3 Probing 'CAD' Generated Objects

Several tests have been chained to reduce computation cost of testing vector penetration in a part. These are following. 'Search scope' control extension to Stereo Lithographic surfaces that are planar triangular surfaces called 'STL' by using 'Smallest Encapsulating Sphere for Surface' patch *'SESS'*. The working is similar to SESO but it leads to data enlargement problem as each surface triangle must have an associated CG and radius stored. Data structure also needs redefinition of object CAD files for this data inclusion or attachments of these data as linked data. SESS for large surface is big. In a typical situation (figure 3.19d) *'$\mathcal{R}$'* approaching towards the object sees two SESS formed by surfaces 2,3 and by 7,8. But SESS 2,3 being on the other side, will not be intersected in reality whereas surfaces 7,8 will be. The basis of this test is that since the $P_n'$ is on robot body, it is naturally outside any object and so any surface with exposed side's normal, not opposite to $PP_n'$ cannot be reached without crossing the object body and so is not reachable. We refer to this test as Surface Direction Test 'SDT'. Surface direction is coded as order of the three nodes in STL triangle data-set in the triangle list. In this work surface direction is used to eliminate surface 2-3 from SESS at position '$k$' as it is not facing move vector $P_n' P_n''$ (figure 3.17). The test is named 'surface direction test' SDT. Algorithm is similar to detecting occluded surface in graphics rendering [43]. It uses angle between the vector $P_n' P_n''$ and STL surface normal '$n$' to test whether the exposed side is towards $P$ or not.

### 3.2.4 Intersection Determination and Computation Reduction

A ray (infinite length) in probe direction '$\overrightarrow{pq}$' is tested for intersection with STL's plane '$S$', formed by nodes $S1$,$S2$ and $S3$. Except for the case when ray is parallel to this plane an intersection occurs (figure 3.20a) somewhere in space at $p''$.

For a penetration to be 'true' two conditions are necessary. i) $p''$ should be within the STL triangle $S1S2S3$ and ii) the distance of this intersection from $p$ i.e $\|\overrightarrow{pp''}\|$ should be $< L_6$ , for the MPC to touch or penetrate the object to which the STL belongs. $L_6$ is the probe length that is equal to longest MPC piston stroke. Only intersection determination with plane '$S$' is called intersection determination procedure 'IDP' in further discussions.

**Range Test :** The second test listed as (ii) above is 'Range test' (RT) and is done first to reduce computation in case a failure was to happen.

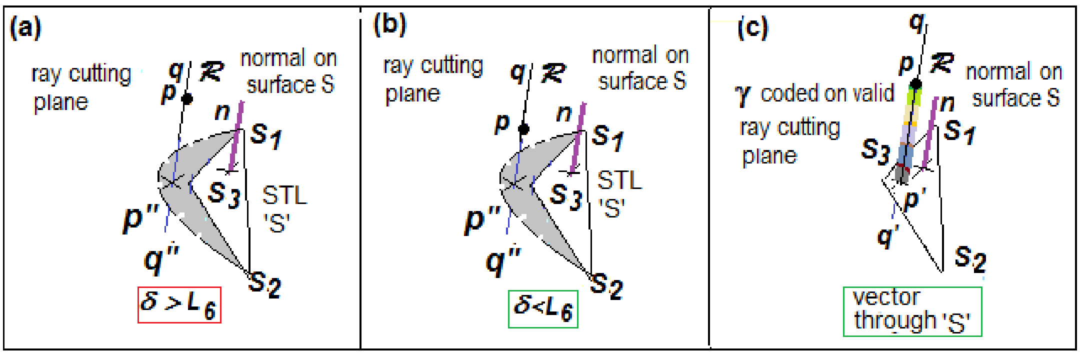



Figure 3.20: Optimization by test order selection – minimizing late fails. The graded bar is aligned along move vector $pq$ which intersects plane of STC on $p''$

**Table 3.2**: Computations in vector- STL object interaction

| Sr no. | Procedure | Operation Performed | Floating Point Operations Count |
|---|---|---|---|
| 1 | Within SESO **"SESO"** Input: $N_{max}$ Output: $N_q$ | a) Non-Optimized (3 S, 3 M, 2 A, 1 SR, 1C)<br>b) Optimized (3 S, 3 M, 2 A, 1 C) | $(12{\sim}35) * N_{max}$ $N_{max}$ = total objects in Workspace. **"SESO (Load)"** |
| 2 | Within SESS **"SESS"** Input: $N_q$ Output: $N_{tq}$ | a) Non-Optimized (3 S, 3 M, 2 A, 1 SR, 1C)<br>b) Optimized (3 S, 3 M, 2 A, 1 C) | (12~35)* $\sum_{i=1}^{N_q} N_{tmax}\,(i)$ $N_{tmax}$ (i) is number of triangles in an $i^{\text{th}}$ object within the object CG radius. "**SESS (Load)**" |
| 3 | Surfaces Facing Observer "**SDT**" Input: $N_{tq}$ Output: $N_{ft}$ | a) Non-Optimized (3 S, 3 M, 2 A, 2 SR, 1 D, 1 T, 1 C)<br>b) Optimized (3 S, 3 M, 2 A, 1 C) | (12~82)* $N_{tq}$ where $N_{tq}$ is number of triangles within the surface CG radius "**SDT (Load)**" |
| 4 | "**IDP**" Input: $N_{ft}$ Output: $N_{ray}$ | (12 M, 9 A, 1 D, 1C) | (57)*$N_{ft}$ where $N_{ft}$ is number of triangles facing vector. "**IDP (Load)**" |
| 5 | Range Test "**RT**" Input: $N_{ray}$ Output: $N_{rayq}$ | 4M 5S 1 C | (14)* $N_{ray}$ where $N_{ray}$ is number of plane intersect within radial distance from observer "**RT (Load)**" |
| 6 | Vector Intersect Test "**VIT**" Input: $N_{rayq}$ Output: $N_{vec}$ | (24M, 18S, 13A, 1 L, 2C) | (82)* $N_{rayq}$ where $N_{rayq}$ is number of triangles having intersection with the observer probe length. "**VIT (Load)**" |

**Notes:** This table assumes there are '$N_{max}$' numbers of objects in the workspace. Here, IDP=Intersection Determination Procedure, S=Subtraction (1 FPC), A=Addition (1 FPC), E=Exponent, SR=Square Root (23 FPC), M=Multiplication (2 FPC), T=Trigonometric (1FPC, considering lookup tables), C=Comparator (1 FPC), D=Division (=23 FPC), L =Logical Operation (1 FPC), CG=Centre of Gravity, '$N_{max}$' = Total numbers of objects in the workspace.

**Vector Intersection Test :** Vector intersection test 'VIT' finds whether the intersection point $q''$ is within triangle STL on surface '$S$' (figure 3.20c). This computation is costly (case 6, Table 3.2). A failure may happen even after this test if the virtual cylinder falls short of touching '$S$'. Therefore a wiser approach is to change order of testing and first test $\delta < L_6$ condition (figure 3.20b) and then test 'inside $S$' condition. It is termed vector intersect test here. It can result in a minimum cost equal to that of ray test and maximum cost equal to that of vector test.

In summary the determination of probe contact with surface comprises IDP+VIT+RT. Ordering VIT and RT is important. Ordering IDP followed by RT and then VIT is beneficial. Process IDP+RT is called vector intersection procedure (VIP).These methods can be seen as filters (figure 3.21) that terminate the test loop early by predicting non-intersection condition and save time. The time '$t_s$' represents time within which sensing should be complete and procedure should start for the next position $P_{n+1}$. Time $t_R$ is available to react in true vicinity. For reducing computations, reusable data like 'bounding box extents, centre and radius of bounding spheres etc. are computed once at start up and associated with part models as additional persistent data.

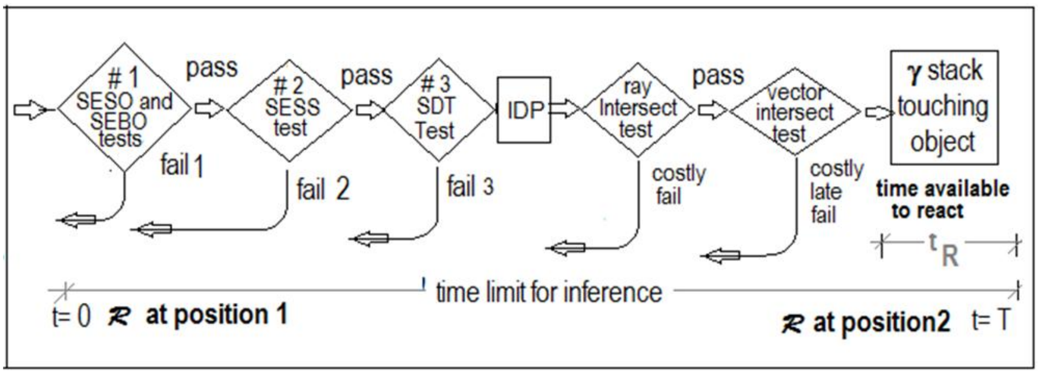



Figure 3.21: Filters for computation reduction in vector object intersection

## 3.2.5 Tests for Assessment of Vector Object Intersection Optimization:

A software test environment has been built to apply the filtering methods on line and access the parameters indicating computations passing through the filter processes at various locations of a point robot on selected locus . Objects (CAD in STL forms) and their pose is user definable and locus too can be chosen as 3D vector. The setup operates through interactive console.

### 3.2.5 a Test set-up:

In the test set up a complex section cylinder comprising 221 STL triangles is used to form a work space scenario by placing two instances of the same object (figure.3.22). The object instance $B$ is rotated 90$^o$ around $Z$ (vertical axis) from pose of object $A$. The point robot $'\mathcal{R}'$ is moved from $\boldsymbol{a}$ to $\boldsymbol{h}$ . The point robot has a sense vector starting from the point and emanates in direction of move and has a length '$L_6$'. We refer this as '$p$'. As explained in earlier section the vicinity computing process avoids finding intersection of $p$ with the objects $A$ and $B$ directly and applies filtering chain (figure 3.21) as it would otherwise require test of intersection with 442 STL surfaces. First it checks the objects using 'SESO' test by test of presence of $'\mathcal{R}'$ sufficiently near spheres embodying $A$ and $B$. For this $\{ r_0 + L_6 > \delta\}$ test is used. Where '$r_0$' is radius of sphere enclosing the object, and '$\delta$' is computed Euclidian distance from container sphere centre to the position of $'\mathcal{R}'$.

### 3.2.5 b Experiments:

The setup executes probe interaction at each location of *'$\mathcal{R}$'* by subjecting the procedure to the filter (figure 3.21) and the tests described above. It graphically show the various parameters namely SESO, SESS, SDT, IDP(Intersection Determination Procedure), RT query, VIT query and total computation. The display cluster is aligned to show all these parameter as bar-graph on sequence of operation.

### 3.2.5 c : Computation Loads in Scope Control Approach:

Consider two objects A and B in workspace as shown in figure 3.22. Consider an object following a path from point ‘a’ to ‘h’.

The number of computation for profile path ‘a’-‘h’ along with their load is tabulated in Table 3.3 and 3.4 for un-optimized and optimized cases.

The optimised version in table 3.4 achieve significant reduction in computation for achieving same results as done by the processes in table 3.3. Maximum gains result in case ‘e’ for the test case where the test location is near multiple objets.

The major advantage come from following -

1. In SESO and SESS computation optimization is achieved by comparing Euclidean distance square rather than computing distance which involves square root (23 FPC) as extra load.

2. In SDT computation optimization is done by calculating dot product between normal to $\boldsymbol{S}$ and vector $PnPn''$ which is an intermediate step of calculating the angle between vectors and comparing the result rather than computation of exact angle. Only those faces which have $dotproduct <= 0$ are considered. The actual angle magnitude calculation that costs (3M, 2A, 1SR) is avoided.

**Table3.3:** Types of tests and their loads at different position of '$\mathcal{R}$' with un-optimized methods

| Position | Cost by SESO | | Cost by SESS | | Cost by SDT | | Cost by IDP | | Cost by RT | | Cost by VIT | | Σ Load |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SESO Query | Units (35U) | SESS Query | Units (12U) | SDT Query | Units (12U) | IDP Query | Units (57U) | RT Query | Units (14U) | VIT Query | Units (82U) | |
| a | 2 | 70 | 220 | 7700 | 16 | 1312 | 8 | 456 | 4 | 56 | 2 | 164 | 9758 |
| b | 2 | 70 | 220 | 7700 | 14 | 1148 | 4 | 228 | 2 | 28 | 0 | 0 | 9174 |
| c | 2 | 70 | 220 | 7700 | 8 | 656 | 0 | 0 | 0 | 0 | 0 | 0 | 8426 |
| d | 2 | 70 | 220 | 7700 | 2 | 164 | 0 | 0 | 0 | 0 | 0 | 0 | 7934 |
| e | 2 | 70 | 440 | 15400 | 17 | 1394 | 6 | 342 | 6 | 84 | 0 | 0 | 17290 |
| f | 2 | 70 | 220 | 7700 | 11 | 902 | 11 | 627 | 11 | 154 | 0 | 0 | 9453 |
| g | 2 | 70 | 220 | 7700 | 17 | 1394 | 16 | 912 | 16 | 224 | 4 | 328 | 10628 |

** Column header indicates type of tests performed and their respective loads*

Note: SESO = Smallest Encapsulating Sphere for Object,
SESS = Smallest Encapsulating Sphere for Surface,
SDT = Surface Direction Test,
IDP= Intersection Determination Procedure,
RT = Range Test,
VIT = Vector Intersection Test.
U = Unit load

**Table3.4:** Types of tests and their loads at different position of ‘$\mathcal{R}$’ with optimized methods

| **Position** | **Cost by SESO** | | **Cost by SESS** | | **Cost by SDT** | | **Cost by IDP** | | **Cost by RT** | | **Cost by VIT** | | **Σ Load** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SESO Query** | **Units (12U)** | **SESS Query** | **Units (12U)** | **SDT Query** | **Units (12U)** | **IDP Query** | **Units (57U)** | **RT Query** | **Units (14U)** | **VIT Query** | **Units (82U)** | |
| a | 2 | 24 | 220 | 2640 | 16 | 192 | 8 | 456 | 4 | 56 | 2 | 164 | 3532 |
| b | 2 | 24 | 220 | 2640 | 14 | 168 | 4 | 228 | 2 | 28 | 0 | 0 | 3088 |
| c | 2 | 24 | 220 | 2640 | 8 | 96 | 0 | 0 | 0 | 0 | 0 | 0 | 2760 |
| d | 2 | 24 | 220 | 2640 | 2 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 2688 |
| e | 2 | 24 | 440 | 5280 | 17 | 204 | 6 | 342 | 6 | 84 | 0 | 0 | 5934 |
| f | 2 | 24 | 220 | 2640 | 11 | 132 | 11 | 627 | 11 | 154 | 0 | 0 | 3577 |
| g | 2 | 24 | 220 | 2640 | 17 | 204 | 16 | 912 | 16 | 224 | 4 | 328 | 4332 |

** Column header indicates type of tests performed and their respective loads*

Note: SESO = Smallest Encapsulating Sphere for Object,
SESS = Smallest Encapsulating Sphere for Surface,
SDT = Surface Direction Test,
IDP= Intersection Determination Procedure,
RT = Range Test,
VIT = Vector Intersection Test.
U = Unit load

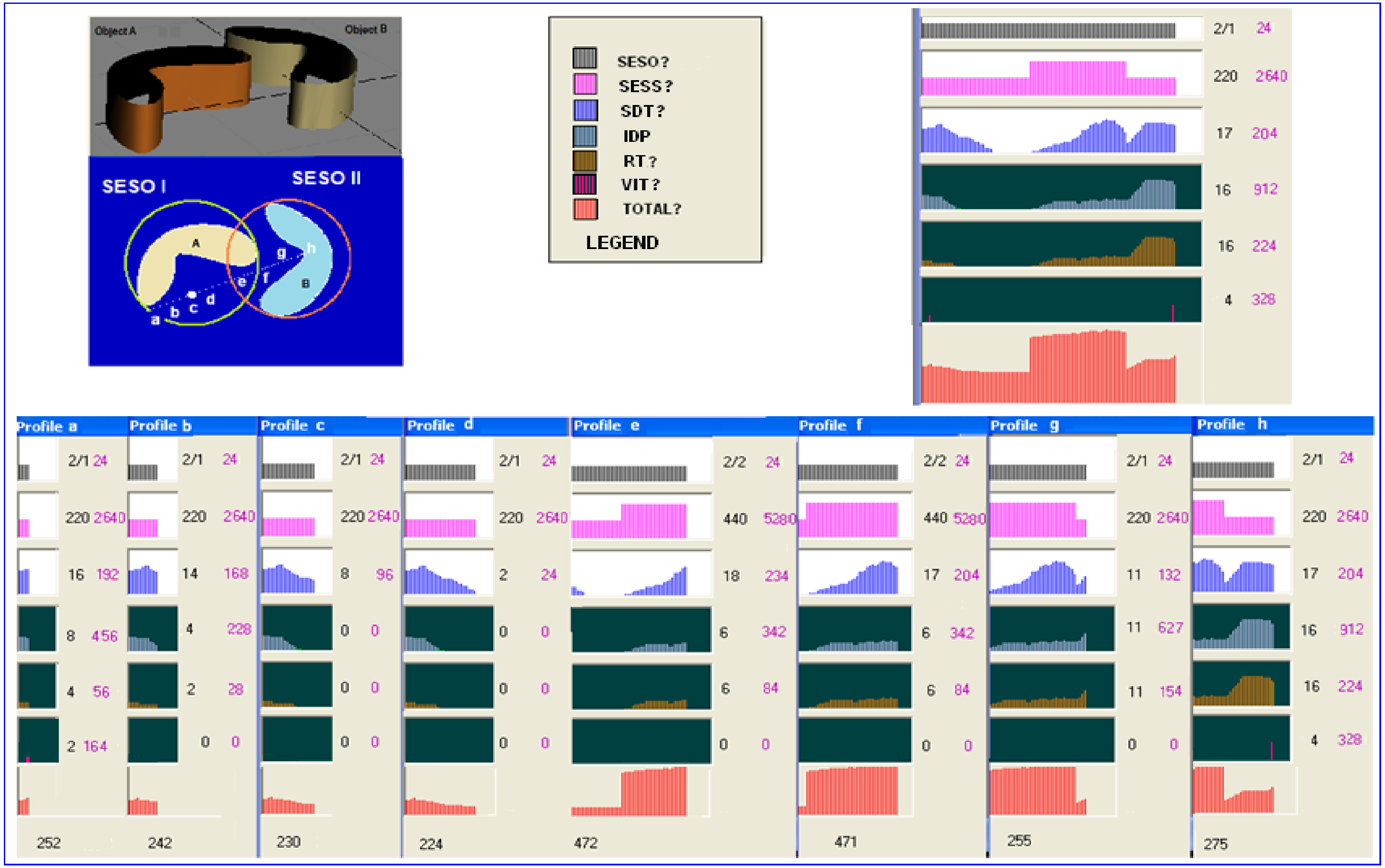



Figure 3.22: Experimental results showing computation loads for vector intersection with STL CAD modelled objects in workspace.

Notable details are as following ;

i. First row in the figure 3.22 shows the total no of objects in the workspace undergoing SESO test i.e. to find out the number of bounding entities which are in range of point ‘$p$’.

ii. The SESO test gives the number of objects which will subsequently undergo SESS test. The SESS test will take all the surfaces of the objects in SESO’s within range into account and finds out which STL surface’s are within point ‘$p$’ range. The number of triangles undergoing SESS test is shown in second row.

iii. After SESS test only few triangles remain which need to undergo SDT (Surface Direction Test) and only those surfaces facing the vector are considered for further computation. Other faces are discarded. The third row shows the number of triangles undergoing SDT.

iv. The fourth row shows the number of intersection point calculated on the surface planes using IDP (Intersection Determination Procedure).

v. The fifth row shows the surfaces undergoing Range Test (RT) where the surfaces which are not in the ‘$L_6$’ range of the point ‘$p$’ are discarded.

vi. The surfaces which pass the Range Test (RT) are then checked for Vector Intersection Test (VIT) where the intersection point’s inclusion within STL triangle is tested. Based on VIT test the hit is qualified. The surfaces undergoing VIT test are shown in sixth row.

vii. The Last row shows the sum of all the tests performed.

### 3.2.6 Results and Discussions

The results of above experiments are analysed to form an insight on their usefulness towards MPC model application. A detailed display in figure 3.23 uses computation quantum (along axis Z) for each filter procedure along filter axis (X) at each test point appearing along location axis (Y). 'Filter' axis corresponds to rows of the figure 3.22. Other axis corresponds to positions a to h and magnitude shows quantum of computation for different types of tests at the location of '$\mathcal{R}$' while moving on test path.

#### 3.2.6a Salient Observations

As we see the profile the following are noted:

1. For profile 'a'-'d' the second row shows no change as we can see in the figure 3.22 that up to position 'd' the point '$p$' is in the vicinity of only one object i.e. '$A$'.
2. In position 'e', there is a sudden jump in the number of candidate STLs for SESS test as we can see now the point '$p$' is in the bounding sphere of two objects $A$ and $B$. Hence the total computations rise.
3. The SDT profile in third row abruptly comes down to zero after point 'd' because the surfaces which were initially facing the observer ray have gone past the point '$p$'. So the total computations also reduce.
4. Sixth row, which shows the VIT computation, indicate high load as VIT is computation expensive. Out of all the points from 'a' to 'h' VIT was invoked only at 'a' and 'h'. Note that otherwise all STL triangles in all the objects of workspace would have required VIT test or at least IDP and RT.

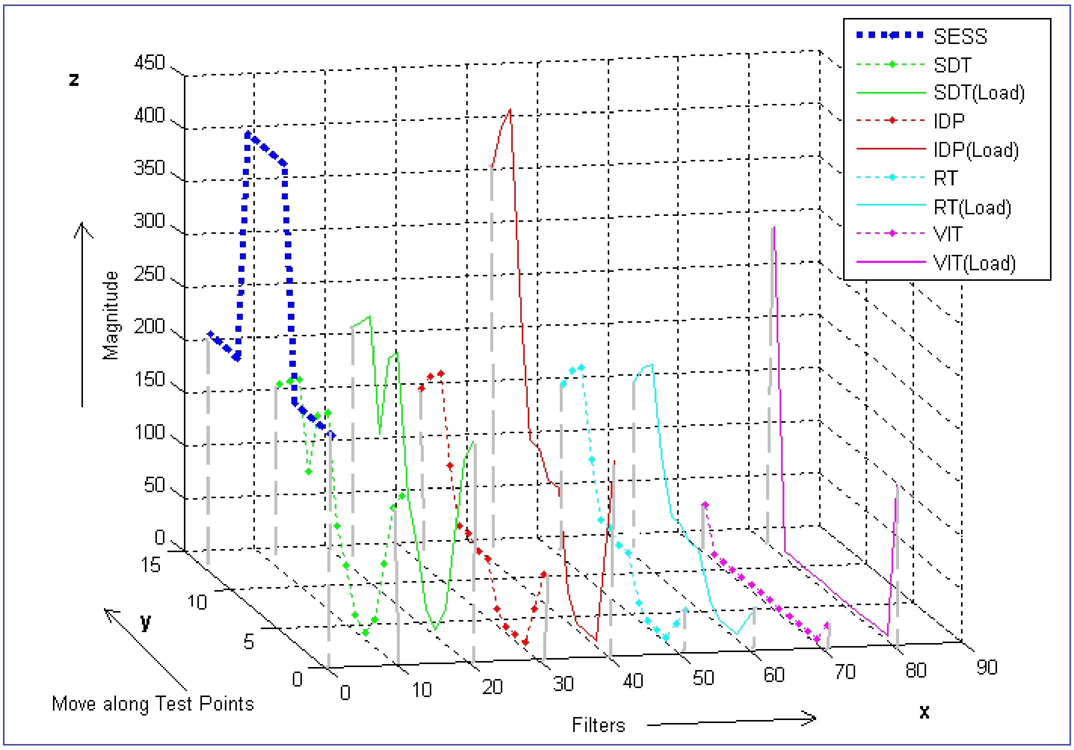



Figure 3.23: Plots of computation minimization filter in tests. Legend gives test type plotted along *'x'* axis as 'filter'. Test position sequence is shown along 'y' corresponds to the test locations *a* to *h*.

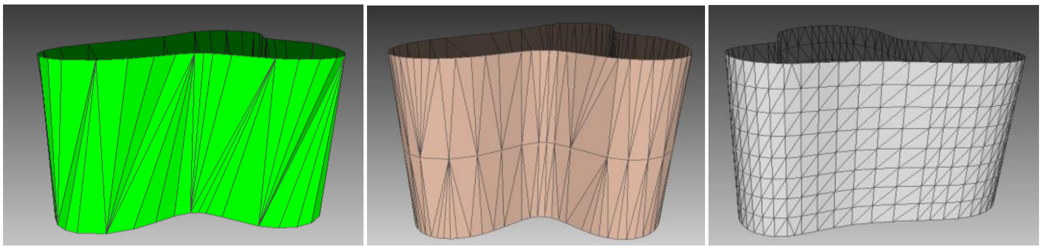
The test position of '$\mathcal{R}$' is along Y axis of the plot. At each location 'a' to 'h', the query for vector $P_n P_n'$ intersect with workspace object proceeds along $X$ axis with dotted and solid pairs of same color show the number of tests and their computation load respectively. The gaps on axis 'x' is used only for showing sliced view.

The number of tests performed at location ('a' to 'h') reduces along $X$ axis. Note that at 'a' and at 'h' (near $grid = 0$ and $grid = 13$) a few STL triangles eventually reach STL hit stage. On other points, STLs are filtered out. Note that 'IDP' is not a test but a mandatory ray-plane intersects computation that has to be done once a STL triangle qualifies SESS and SDT. Therefore it is important to have homogenous sided STLs.

### 3.2.6 b On Performance of Search Scope Control approaches

SEBO, SESO tests are equal to number of objects but cost of test is small, But objects with large number of STL surface segments such as those with NURBS surfaces bring in large SESS candidates needing elimination by SDT or RIT (Ray Intersect Test). Both are computationally costly The basic criteria is that free space inside SESO or SEBO should be minimised (figure 3.19b, 3.19c), mutual overlap should be minimised.

Note that the depicted phenomena is for single vector '$p$', but a large number of vectors emanating from '$\mathcal{R}$' at entire front hemisphere has to be considered and generally are more than 32 in number. The computations are as many times of that plotted here. Real time performance failure happens as computation load grows to extremely high figures near object when time overshoot can result in hit.



(a) (b) (c)

Figure 3.24: STL model of complex cylinder with same accuracy, a; has heterogeneous side triangles, b; better STLs than (a), c; homogeneous STL in large numbers.

The effectiveness of SESS lies in homogeneity of STL sides' size and their magnitude. In example figure 3.24 model error is same as the top surface polygon is converted to same polygon in all the cases and the object being cylinder, does not have any effect of STL triangle heights along cylinder axis. The (b) version needs double the numbers of triangles and corresponding data size. The representation in figure3.24c has higher number of STLs than first. The SESS are smaller and so fewer of them qualify for further tests when tested at locations of '$p$' farther than the small radius of their SESS. They qualify for the costlier intersection test much later during point robot approach towards the wall. Therefore reducing total numbers of STL surfaces does not definitely yield overall advantage as number of STL test increase while intersection test reduce.

As a consequence, for case figure 3.24c, 8 times the STLs shall be subjected to SESS test as compared to that for case 'a', but only a fraction will be subjected to ray test. However a good fraction of the STL qualifying in case 'a' will be eliminated by 'SDT'. On the other hand those STL qualifying from case 'c' may not be filtered much unless the object is very thin and SESS of opposite side also qualify.

# 3.3 Observations on Model Implementation

The optimisations described in the earlier section need pre-processing of STL model to attach data for SESS and reconstruct the data in different form. Further the effectiveness is not guaranteed to yield real time performance owing to search dependent approach which are inherently non deterministic. The filters do not help when the probe is in concave area near object and large STL's qualify for further test. The main function of filters is to free up a processor from one vector computation to take up another in multiple vector tests running on parallel threads on a computer.

# Closure:

The MPC model conforms to haptic design criteria well, by using a physical model whose haptics model is easy to understand and adequate information is available. The model succeeds in rendering the proximity and the robot-dynamics effects as kinaesthetically perceivable effect on operator hand.The model is apparently very compact in terms of computation load but its attempted implementation routes are not viable owing to uncertainty in computation load. Various techniques from the domain of obstacle avoidance are promising. The task ahead is to identify suitable modelling technique that can circumvent the presently encountered problem. Of particular interest is to evolve techniques that reduce search or eliminate search. Chapter 4 is devoted on overcoming this limitation.

# CHAPTER 4

## ‘VOXEL’ MODELLED OBJECTS

## FOR MPC IMPLEMENTATION

## AND

## MODEL RE-CONCEPTUALIZATION

Historically, use of discrete volume elements in haptics modelling has been reported in literature. Discrete type part-model can be formed by dividing the space occupied by it into unit cells (or volumes) [65]. To check for collisions, one has to examine if the cell is shared by other objects. For an environment where objects are of uniform size, this is a rather ideal algorithm and especially suitable for parallelization. Overmars has shown that using a hash table to look up an entry and storage space, one can perform the point location queries in constant time [65].

# 4.1 Voxel Based Model

In relatively recent works on graphics, the discrete volume model has taken shape for medical imaging and virtual reality domains owing to the high capacity desktop computer availability. A voxel is essentially a volume element in a regular grid in three-dimensional space. Word 'Voxel' is a combination of "volume" and "pixel" which is basic element in image processing.

## 4.1.1 Motivation to Use 'Voxel' Models

Let us visualize a workspace digitized in volume bits called 'voxel', where every voxel is owned by some object or free part of workspace. In the figure 4.1, the cube

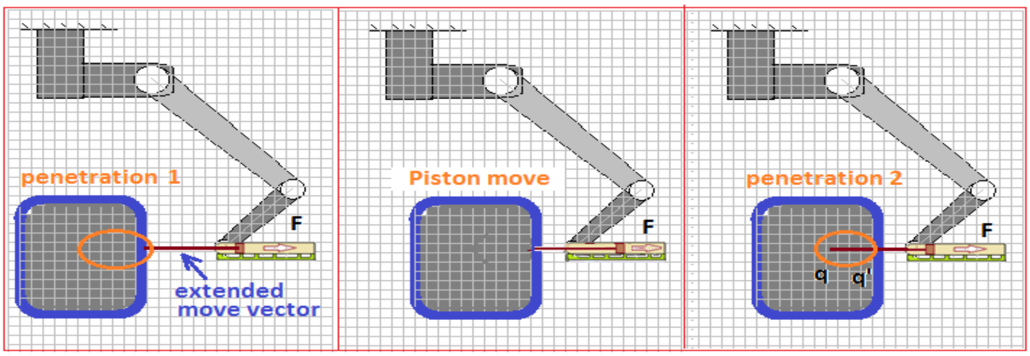



Figure 4.1: Estimating MPC piston move against object by simpler method.

object has gray voxels and free space are white voxels. For a push against physical object in figure 4.1(a), the piston moves in MPC model as in (b). For a virtual voxel model the piston pusher rod can be visualized to penetrate the object at $q$ and reach $q'$. The depth $qq'$ is a measure for piston travel. Hence MPC model can be applied by counting grey voxels on $qq'$ vector.

### 4.1.2 Demands on Force Modeling

The force values applicable for robot object interaction are used in the force servo loop and hence the force location and orientation computation must run at high speed. As a consequence the update rate puts a constraint on the complexity of the algorithms that can be used for finding the closest point between robot and a part and thereby restricts the types of models to the simpler ones that could be used to meet the rendering requirement. This in turn jeopardizes wide scale automation of the method as for achieving generality as almost all types of CAD object models need to be accepted as input and used for MPC model implementation. The Voxel based route has a promise to overcome this.

## 4.2 Voxel Representation

A voxel represents a value on a regular grid in three-dimensional space. Voxel is a combination of "volume" and "pixel" where pixel is a combination of "picture" and "element". As with pixels in a bitmap, voxels themselves do not typically have their position (their coordinates) explicitly encoded along with their values. Instead, the position of a voxel is inferred based upon its position relative to other voxels (i.e., its position in the data structure that makes up a single volumetric image). In contrast to pixels and voxels, points and polygons are often explicitly represented by the coordinates of their vertices. A direct consequence of this difference is that polygons are able to efficiently represent simple 3D structures with lots of empty or homogeneously filled space, while voxels are good at representing regularly sampled spaces that are non-homogeneously filled. Voxel oriented approaches have delivered very encouraging results in 3D graphics [66,67].

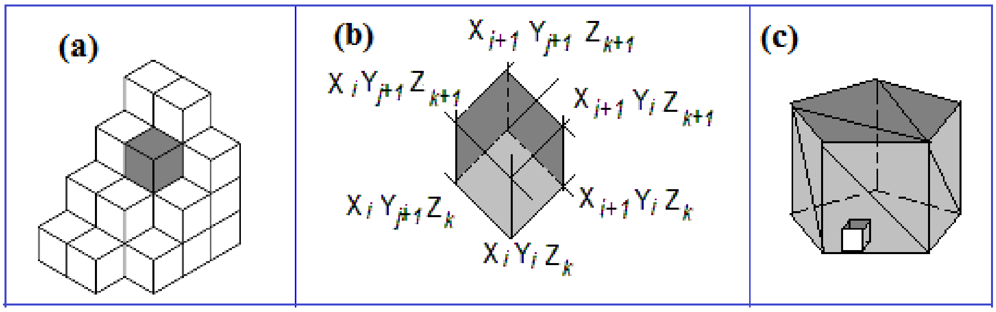



Figure 4.2 a; voxel relation to other voxels, b; voxel definition, c; surface voxel of hexagon section cylinder object.

### 4.2.1 Defining a Voxel Model

In voxel based approaches workspace is represented as 3 dimensional array of equal sized cube shaped volume elements with integer coordinate locations $(X,Y,Z)$ that are referred as 'Voxel' and denoted as $V_{(X,Y,Z)}$in further descriptions [68]. A voxel $V_{(X,Y,Z)}$ represents cube with diagonal nodes $X_i, Y_j, Z_k$ and $X_{i+1}, Y_{j+1}, Z_{k+1}$ as shown in figure 4.2a. A voxel has physical size referred as $voxel\ size(x)$ , $voxel\ size(y)$ and $voxel\ size(z)$ along $X$ , $Y$ and $Z$ directions rspectively in units of length as per physical space to model mapping. The indices range from 0 to a max integer value $X_{max}, Y_{max}, Z_{max}$ suitable for representing full extent of workspace with desired granularity. The voxels was assigned 'potential' property for path finding problem in [69]. Here we use them to assign other properties. For representation of spatial occupancy of space by parts in the workspace, we adhere to the following interpretation. A grid position in workspace where no object exists is represented by free voxel. A voxel through which a triangular surface passes is boundary voxel (figure 4.2c), a voxel inside object is occupied voxel. Typical voxel data is a record (figure 4.3) with last field as signed integer. Note the first attribute is for pointing in 3D linear array of voxel data and not included as record and the object index has utility in back-linking to object and may not be used in memory minimized versions.

| | | |
|---|---|---|
| Grid Pt. X,Y,Z | Object index | Type - Free / Boundary, Body |

Figure 4.3: Data associated with a voxel at grid point $X$, $Y$, $Z$ in 3D voxel array.

The voxel model of object need huge data to represent an object and being discrete elemental volume representation are prone to errors when subjected to rotation transforms and can form holes in thin object models. For reducing memory requirement Octree structures have been proposed. It is a hierarchical data structures for representing solid modes. Reddy and Rubin [70] proposed the Octree as one of three representations for solid objects. In figure 4.4, from a rectangular block (figure 4.4a), white volume parts formed by binary subdivision of different order of the full

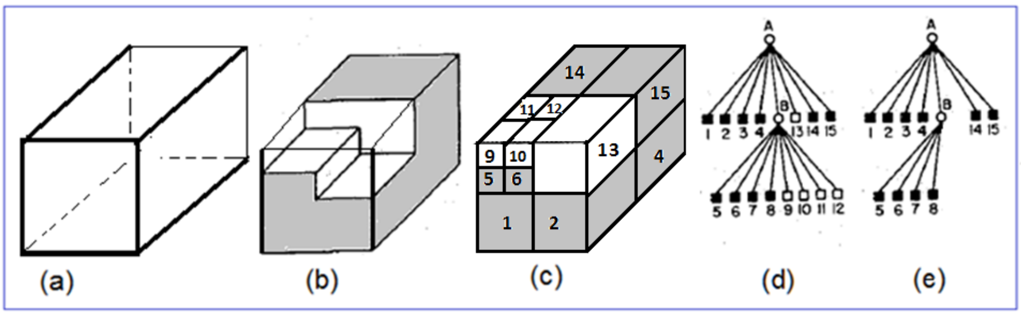



Figure 4.4: An object (b) is formed from (a) and structured in hierarchical model based on octree shown in (d) and (e)

block have been removed (figure 4.4b). Resulting part is modelled as filled and vacant blocks (figure 4.4c), shown as filled, and vacant nodes in tree structured data by octree subdivision (figure 4.4d). The memory reduction is achieved by not including empty nodes. Note that cubes at each level of tree are of same size.

The original formulation of the quadtree [71] encodes it as a tree structure that uses pointers. This requires additional overhead to encode the internal nodes of the tree. The hierarchical data structures causes' difficulty in adjacent voxel access if they do not appear on same layer and bottom to top search for a common parent node is necessary as in case of node 6 and 2 in figure 4.4 above. The reducing cost of hardware enables high memory use and so fixed size linear three dimensional arrays is preferred in this work, as it permit fast and straight forward access of geographically adjacent data.

## 4.2.2 Data structure for Voxel Models

The voxel model is built in three dimensional linear array with all voxel of same size and data record as indicated in figure 4.3. No metadata should be used. This criterion is important owing to tight real time execution needs which must avoid any process spawning for accessing voxel data. Also the data access must be amenable to fast orderly array based accessing. As meta data access can depend on the encoding employed by the data formation source, minimization of access time suggests that the voxel data formation should be 'native–accessible' needing no decoding or spawning process. Drawing analogy from 'pixel' it should be amenable to array based access, albeit of higher dimension and in 'object' based approach in computer processing it must be an object (software) sans any method and comprise of only property.

### 4.2.3 Voxel Model Formation

The voxel model formation methods are modeling method dependent. For all methods a subspace with its extent sufficient for fully containing the object is formed in a Cartesian space that is suited for workspace representation. This is achieved by creating an Object Voxel Array 'OVA' (3D) in Cartesian space with virtual grid at spacing decided by voxel size The part model is placed in it as per its actual pose and position in workspace by suitable translation and rotations of part model. We call it part instancing.

#### 4.2.3a Voxel Model Formation Approach for Implicit Models

For implicit defined models, the part model is instanced in OVA. Then using the part defining mathematical (refer section 3.3.2) equation, a grid $x.y.z$ is evaluated to test, $if\ \ f\ <\ 0$, If true then the voxel is inside the part body, otherwise it is outside. The test is applied to all the grid points inside OVA.

#### 4.2.3b Voxel model formation approach for bounding surface models

For all the methods that use some form of bounding surfaces, the voxelisation is done in two passes. In first pass the surface voxels are identified by surface and OVA grid intersection to finds if a surface of the object passes through the voxel and labels it as boundary voxel. In second pass, the inner voxels are identified by scan line approach, popular in image processing methods, as the surface voxels yield closed contours in single voxel thick object slices parallel to $XY$, $YZ$ or $ZX$ plains.

#### 4.2.3c Voxel Model Formation for CSG Based Part Models

The CSG standard primitives are the sphere, the cylinder, the cone, the parallelepiped, the triangular prism, and the torus. These can be objects formed by bounding surfaces of types treated in Chapter 3. If the component details are exposed by the modeling method for the individual parts, in pose and position at which they are merged, these can be voxelized. After this, only logical operations are done on the voxel groups of the respective parts.

Boolean operation is applicable on these voxel sets. The subtraction method changes the voxel property of the voxel corresponding to the removed part to empty from occupied. The difficulty arises in relative positioning of parts of the object before logical operations, as being voxelized, the part's position has to be only at discrete grid value. An alternative route is to form intermediate mesh form from CSG and use method developed hereafter as generalized solution.

# 4.3 Universal Voxel Models for Automatic Formation

Objects of almost any complexity can be designed or reengineered from a natural or manmade form by modern CAD systems. A method to make voxel models from them gives a universal route for forming voxel representation of objects.

## 4.3.1 Voxel Model Formation from CAD Designed Parts

Mesh representations of objects provide a very effective intermediate rpresentation for building voxel models of objects. These are composed of an array of small faces. In the native modeling method mesh building is feasible. For example in sphere formation by polar method, use of $\theta$, $\phi$ with constant steps and constant radius, generates points on surface that form regular mesh. Similarly in generalized cylinder the smooth 2D closed curve representing cross section can be approximated by polygon of '$n$' sides and mesh is formed by array of this array.

#### 4.3.1a Mesh models and Precision

Quadrilaterals formed by the adjacent node sets form planar facets and with facets extremely fine, the surface representation will generally be good. Such model will be very data heavy. The optimum resolution will depend on the usage of the model.

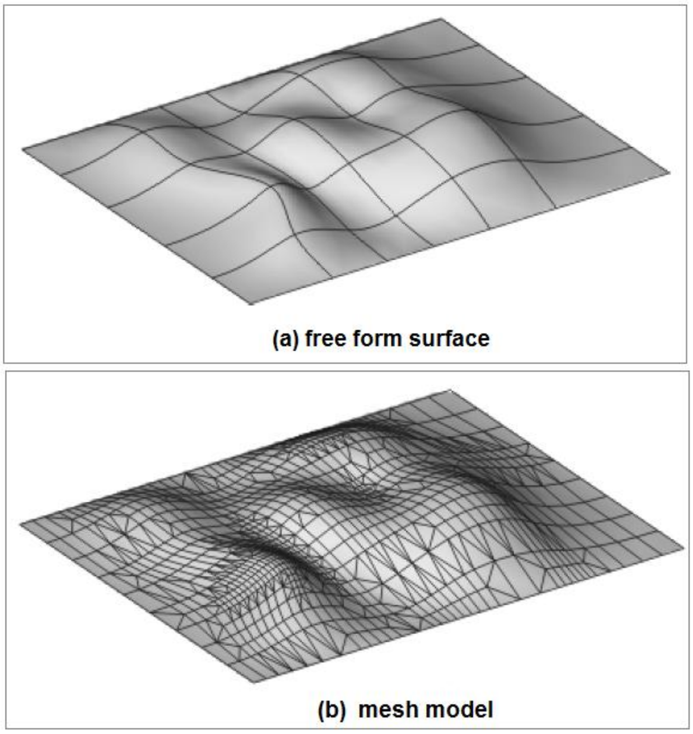



Figure 4.5: Free form surface (a) and its mesh model (b)

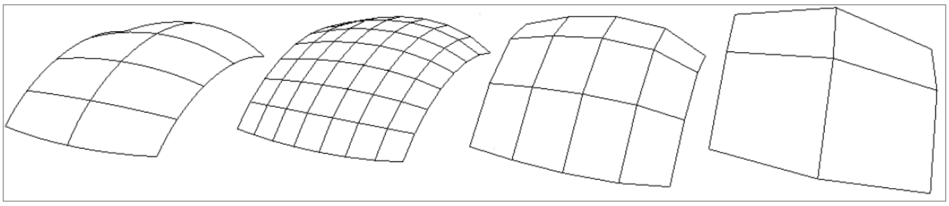


Figure 4.6: The initially smooth NURBS-surface (on the left) is converted into a facetted polygon mesh. According to the chosen accuracy, the user can define different level of detail.

If the individual facets in a mesh model are too coarse or there is high angular deviation between them, the object model will lack precision. Mesh precision may be thought of as the maximum difference allowed between the facetted mesh representation of the surfaces and the smooth surfaces themselves.

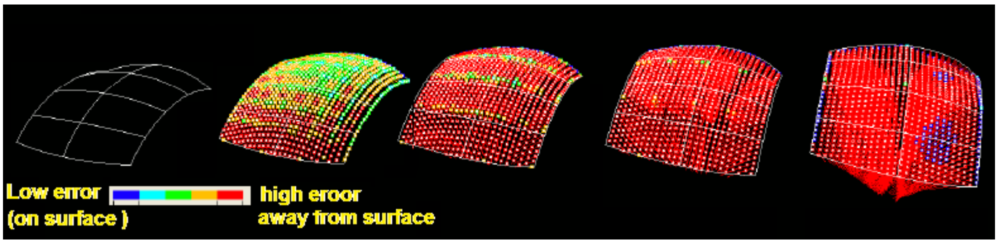



Figure 4.7: The relative error is low in smaller mesh as shown by higher population of blue and green portions. For larger mesh error is shown by red population

## 4.3.2 Forming Voxel Based Model for STL Objects

3D CAD models can be made with many, different software which do not expose their internal data structures and metadata from which the models are either rendered or document is prepared. However nearly all 3D CAD systems can export an STL[72] and most can import them. These represent the surface(s) of the object(s) in the form of one or more polygon meshes. The polygon meshes in an STL file are entirely composed of triangular faces, edges and vertices. Further, the node order in data-set indicates their orientation (inside/outside). For objects composed of entirely planar surfaces, this is not really a problem, as the facets will correspond exactly with the surfaces. For curved surfaces, the triangles will necessarily not lie entirely on the surface, and thus the degree of approximation is important.

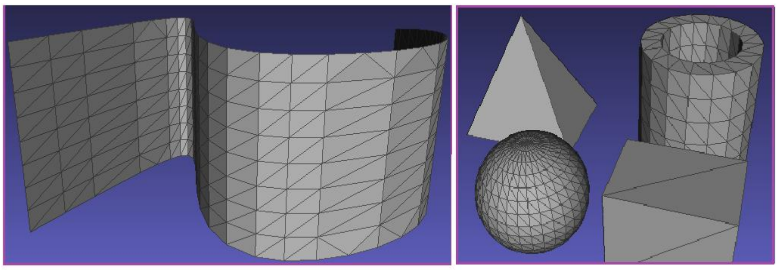


Figure 4.8: Triangular planar STL surfaces from CAD models

### 4.3.2a Surface Voxel Identification

For CAD models in STL version, data representing the 3D shape of object is associated to workspace geometry as per its pose and position in work space using node set rotations and translation. This also eliminates hole forming problem that can occur if voxelized model is rotated [73].

Triangular surface segments in CAD-STL model need to be computationally intersected by grid lines of Cartesian coordinate system. Intersection of triangular plane (STL element) with grid vector is more precise if intersecting vector is not

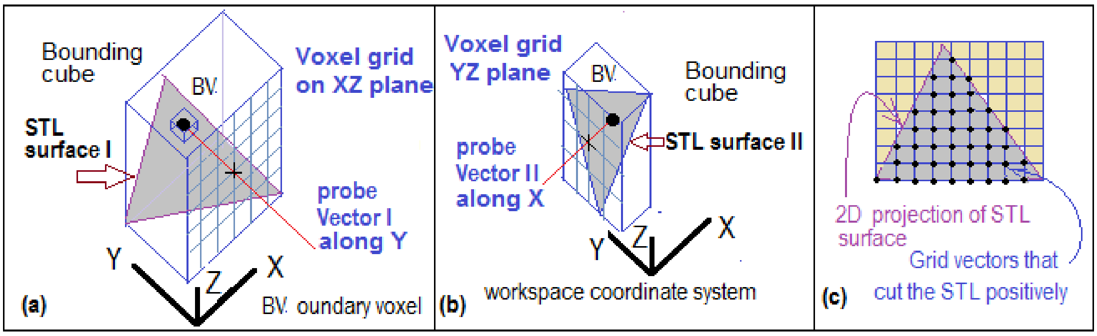



Figure 4.9: Minimizing probe vectors for optimized intersection.

parallel or near parallel to surface. Generally grid vectors of any orientation i.e. either $X, Y$ or $Z$ can be used to intersect the STL triangle but to ensure intersection by grid vector of appropriate orientation, minimum bounding box is formed with three nodes and largest face is found. Grid line normal to this face and contained by the projected face is used (figure 4.9a and 4.9b) to intersect the triangle for Boundary Voxel (BV) determination [74, 75]. Mark use of vectors of different direction in figure 4.9a and b. For computation, only grid vectors existing within the projected triangle face are used to avoid redundant vector–plane intersection test which eventually fail after consuming computation time. Reliable intersection tests are necessary [68]. In cases where the voxelisation is done as a pre-process operation to real time robot run phase, the aforementioned process is repeated for all the 3 projection faces. This is equivalent to STL face intersection by gridlines parallel to $X, Y$ and $Z$ primary axes in 'OR' mode wherein a voxel at an $x, y, z$ location is marked occupied in OVA if the STL triangle is intersected by any of them. This ensures that all voxels on STL surface near edges are marked as occupied voxel.

### 4.3.3 Voxel Model of a Single Object

The object's voxelized model is held in 3D array of records (figure 4.3) called object voxel array $OVA\ (x, y, z)$.

#### 4.3.3a 'OVA', the Voxel Holding Array for an Object

An Object Voxel Array '$OVA$' is formed in Cartesian space to contain all occupied voxels by a single object. Subspace from the full workspace bound as 'axis aligned bonding box' around a single chosen object is formed by forming 3D voxel array and is referred as object voxel array $OVA(x, y, z)$. OVA has associated extent definition data. $Min\,XYZ$, $Max\,XYZ$ define its location in the overall Cartesian workspace $WVA$.

#### 4.3.3b OVA Formation Procedure:

I. Extract $x_{min}, y_{min}, z_{min}$ and $x_{max}, y_{max}, z_{max}$ (diagonal point coordinates of $OVA$) by scanning the node list in the 'Object STL File'.

II. Calculate $OVA\_offset$ as

$$OVA\_xoffset = \left[\frac{x_{min}}{voxel\ size(x)}\right]$$

$$OVA\_yoffset = \left[\frac{y_{min}}{voxel\ size(y)}\right]$$

$$OVA\_zoffset = \left[\frac{z_{min}}{voxel\ size(z)}\right]$$

III. Calculate ranges of $x, y$ and $z$

$$OVA\_xrange = \left[\frac{x_{max}}{voxel\ \ size\,(x)}\right] - OVA\_xoffset$$
$$OVA\_yrange = \left[\frac{y_{max}}{voxel\ \ size\,(y)}\right] - OVA\_yoffset$$
$$OVA\_zrange = \left[\frac{z_{max}}{voxel\ \ size\,(z)}\right] - OVA\_zoffset$$

These are used for array definitions.

IV. Procedure for voxel coding :

Voxel data is formed by using voxel record (figure 4.3) in trim form in a 3 dimensional array $Voxel\_Data(x, y, z)$. This procedure works on STL object definition file.

i. Take 'STL' triangle node tuple from object definition data and project points $P_1, P_2, P_3$ on $xy, yz, zx$ planes

ii. Find the largest projected 2D triangle.

iii. Use grid lines normal to the projected plane and within 2D projected triangle for intersection (figure 4.9c).

iv. $(x_c, y_c, z_c)$ is typical intersection of one gridline with this triangle.

v. Voxel indices for OVA array access and voxel record update

$$x_{index} \ = \left[\frac{x_c}{voxel\ \ size\,(x)}\right] - OVA\_xoffset$$

$$y_{index} \ = \left[\frac{y_c}{voxel\ \ size\,(\mathrm{y})}\right] - OVA\_yoffset$$

$$z_{index} \ = \left[\frac{z_c}{voxel\ \ size\,(z)}\right] - OVA\_zoffset$$

vi. Modify $Type.Voxel\_Data(x_{index}, y_{index}, z_{index})$ to boundary (figure 4.3)

vii. Repeat steps (iii) to (vi) for all the gridlines within the triangle.

viii. Repeat steps (i) to (vii) for all the 'STL's in object definition.

Note: In above procedure [.] represents integer operator.

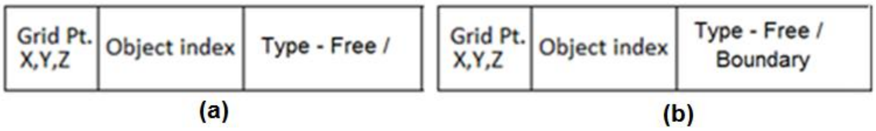



Figure 4.10: (Left) input data to this procedure; (right) output from the procedure.

Now the OVA array contains all surface voxels for the object as 'boundary'. The slices parallel to any principal plane, i.e. $xy, yz, zx$ consist of boundary voxel surrounding voxel corresponding to object body. Such slice of OVA is analogous to pixel image and scan line algorithms [75] can be applied to fill the body voxel properties in the OVA held object model. Complete coverage of object is a trivial repetition of bit plane process to cover complete object.

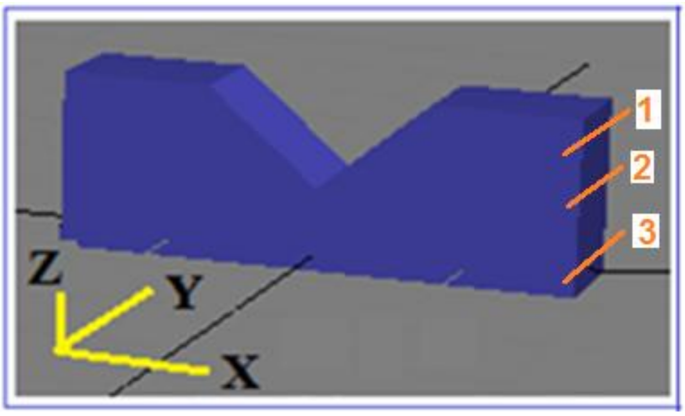



Figure 4.11a Wedge object as voxelisation specimen

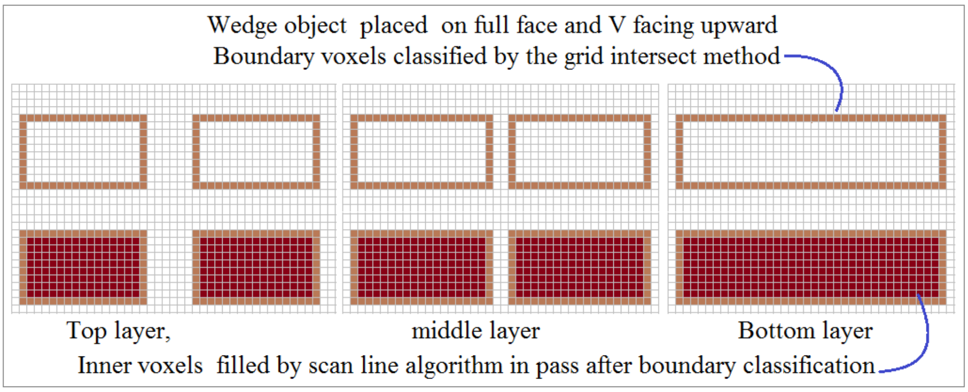



Figure 4.11b: Use of scan line technique to fill interior voxel on object slices with boundary voxels at level 1 ,2 and 3 of figure 4.11a.

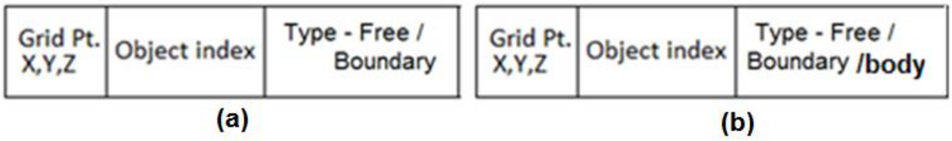



Figure 4.12a; input data to this module, b; output from the method.

### 4.3.3c Test of Voxel Formation Method

Two objects of different type, one with planar surface (wedge) and one with curved surface (cone) were converted to voxel version, shown in figure 4.13 and 4.14. The results show the correctness of the procedure as these are mapped in 3d space and shown by their graphic display. For the cone object the diameter change from bottom to top is depicted in the results of voxelization. A section through the vertex is also plotted to test the fidelity of conversion completely.

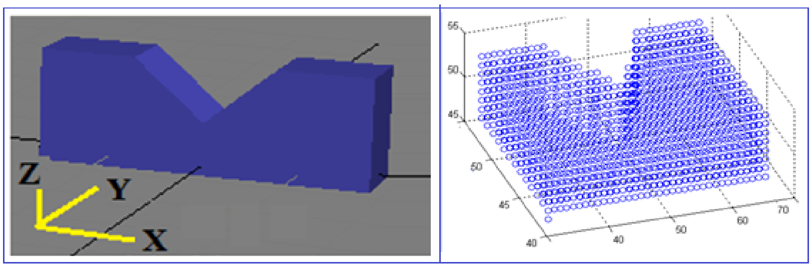



Figure 4.13: The wedge object CAD model(figure 4.11a) converted to its voxel model, in OVA as a result of object voxelisation

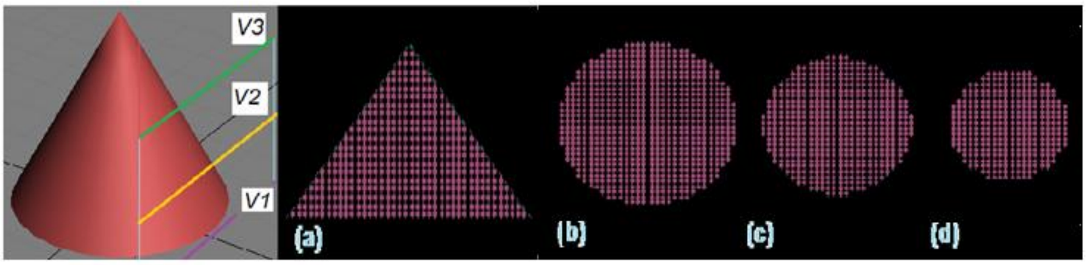



Figure 4.14: Voxelization results for a cone (top left) shown vertical section through its apex (a), as sections (b) to (d) at various height starting from base shown in (a).

# 4.4 The MPC Model Function in Voxelised Space

The $\gamma_{(n)}$ formation effectively needs computation of distances from nearest surface (figure 4.15). For forming $F$ only $\gamma_n$ is required (equation 3.11) only. $v$ is dynamic parameter associated to robot and remaining is property of part neighbourhood in workspace. It can be treated as graded γ bar affixed to an object's external STL surface as normal (figure 4.16). For dense placement this appears like porcupine's spikes.

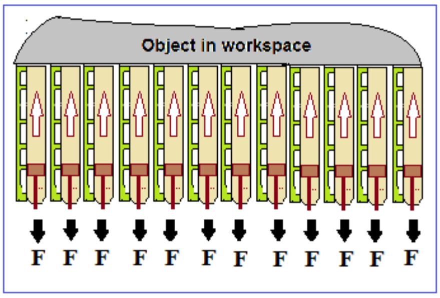



Figure 4.15: Object centric MPC application concept

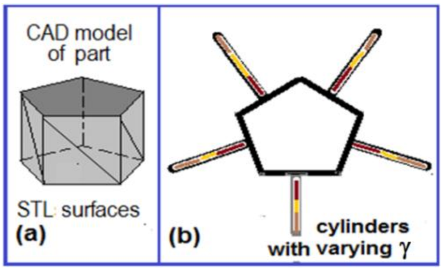



Figure 4.16: MPC model attachment to surface and degenerate cases

The $\gamma_{(n)}$ can be treated as added property of STL. The need for search for the nearest STL patch shall persist while finding applicable $\gamma$ bar. This must be eliminated for the system to sense rapidly and achieve deterministic response. Imagine a dense placement of the bars around object with (figure 4.17).

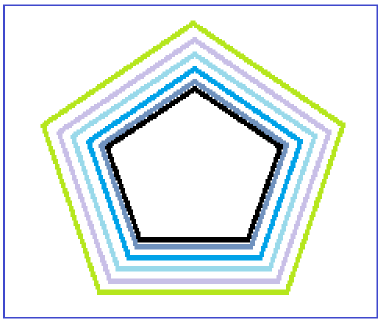


Figure 4.17: An alternative approach to implement object centric MPC

The appearance suggests that the arrangement can be treated as layers of $\gamma_{(n)}$ coded space and so motivates us for developing object centric volume coding method. Shape representation of object has been addressed by 'virtual reality' researchers [76]. Volume representations in graphics methods [66] too are relevant.

## 4.5 Ranked Neighborhood Formation by Voxel Model

Another spatial distribution centric view at figure 4.17 suggests that the space surrounding a workspace object be imparted property as per a ranked neighbourhood criterion in which space touching the object surface from outside is ranked as closest (ranked 1) and space touching the rank 1 surface from outside is ranked 2. Such ranking can be imparted till an order '$q$' which is sufficient to represent the '$\gamma$' layer.

## 4.5.1 Neighborhood Segmentation

In general if progressively larger shells are formed around objects and the voxels inside these shells are assigned respective '$\gamma$', a penetrating object can very quickly find applicable '$\gamma$'.

### 4.5.1a Use of CSG Techniques for Forming Neighbourhood:

The model representation by CSG method has been treated in chapter 3. Its use of set-theoretic Boolean expression of several types to combine primitive objects suggests a way to form surrounding space of an object. One way of forming it is to scale up an object and voxelise it. Then subtract voxels of the original object from the same. The remaining voxels can be attributed the layer assignment. Successive repeat of the process can give multiple layers. In figure 4.18 b and d, layer voxels belong to coloured neighbour zone. The method works well for parts that are 3D symmetric to the centre point and free of concavity i.e. the original part should not have been formed by subtraction.

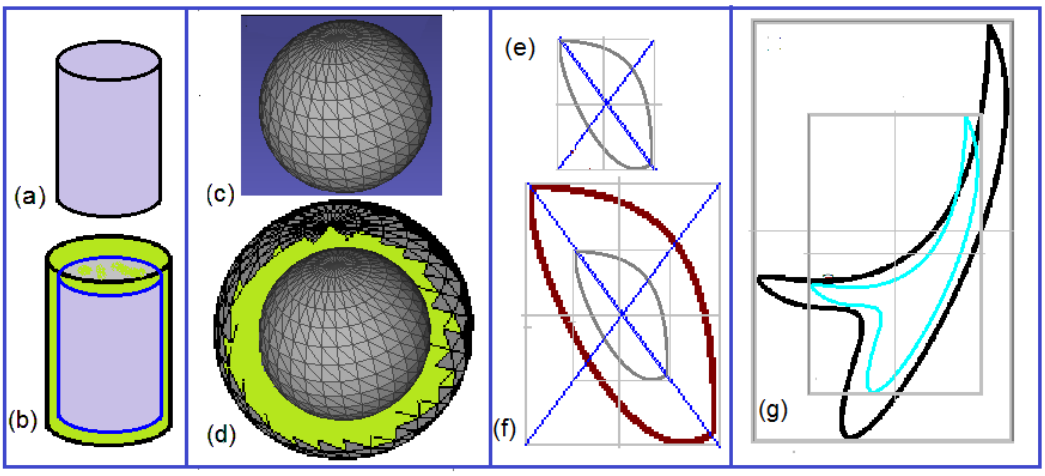



Figure 4.18: Shell formation around object by CSG method. Pairs (a)(b), (c)(d) show neighbor space formation by subtraction, pair (e)(f) and (g) show method failure.

Asymmetric and concave parts fail to produce valid layer voxel (figure 4.18f and 4.18g) by CSG based methods unless the part is decomposed into all convex only objects and objects that are subtracted are shrunk rather than expanded. Commercially available CADs do not provide the CSG formation support to be used out-of the CAD environment. Bezier surfaces, lofted surfaces and other higher order polynomial based construction methods used in designing the component parts in CSG assembly, also need to be addressed. Scaling up native curved surfaces is difficult and methods using their mesh approximation fail as errors increase in scale up. Therefore neighbour creation task must be addressed in part design stage itself. Devising generalised method applicable to all CAD systems is not feasible through this route. The shell width depends on application scenario and is not available at time of component design as only later the parts find place in workspace. Since parts in workspaces are prebuilt as hardware and only models are inserted in workspace or are modelled by *in-situ* sensing using dense or sparse range sensing on outer faces, the CSG based approach is impractical.

For objects with surface defined by NURBS [50, 43] building larger objet of some size is complex. The trims that limit the NURB to the boundary of the patch needs to be computed again for the scaled up object that itself has a different NURB description as compared to the original object. Building exactly parallel outer surface is not possible for all the objects. The concave parts of curved objects need different treatment.

#### 4.5.1b STL Based Bloated Object for Neighborhood Formation

The intermediate model like the STL [77] is supported by all commercial as well as open source CAD systems. Though an algorithm can be formed for building a larger version of such model as described below, it can lead to unacceptable error in some cases. The approach works by forming parallel planar surfaces using the STL surface normal going outward from STL surface. By using increasing distance between the parallel surfaces, multiple shell scan can be formed.

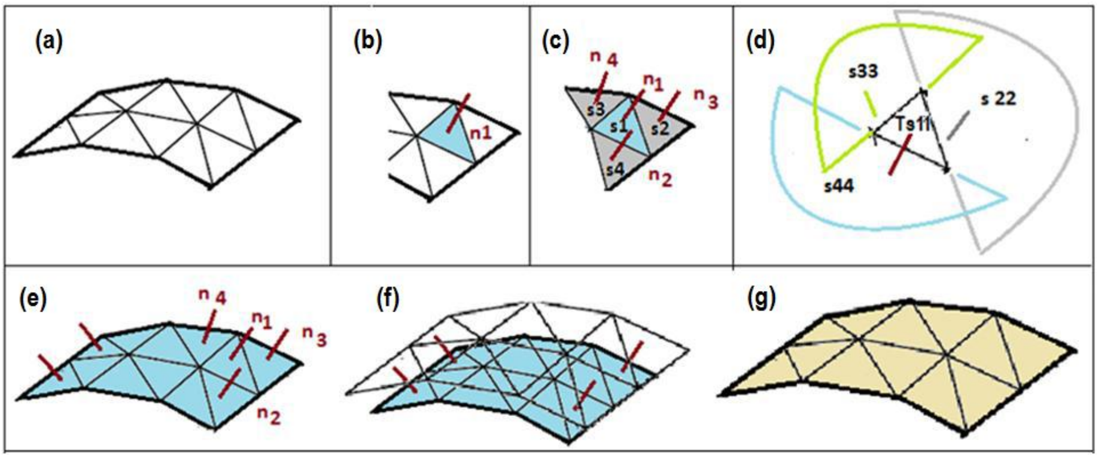



Figure 4.19: Shell formation around mesh represented STL standard objects by using surface normals and parallel planes.

**Proposed Method for forming larger object from STL**

Refer figure 4.19 for this procedure.

a. Compute Surface normal for finding $\boldsymbol{n_j}$toa surface $S_j$

b. Form vector as per distance between object along $\boldsymbol{n_j}$ at STL triangle center

c. Compute plane at this new vector end point parallel to old surface $\boldsymbol{S_j}$ say $\boldsymbol{S_{jj}}$.

d. Repeat a, b, c for three adjacent surfaces $S_2$ $S_3$ $S_4$ to form $S_{22}, S_{33}, S_{44}$ Compute intersections of $S_{11}$ with $S_{22}$ and find intersection edge.

e. Repeat to form triangular surface $TS_{11}$(figure 4.19d)

f. Repeat for all surfaces $S(2)$ to $S(n)$ to cover entire objects STL list (figure 4.19e, 4.19e f )

### 4.5.1c Drawbacks of the Method

In this method, error magnification occurs as original STL itself has approximation.

.

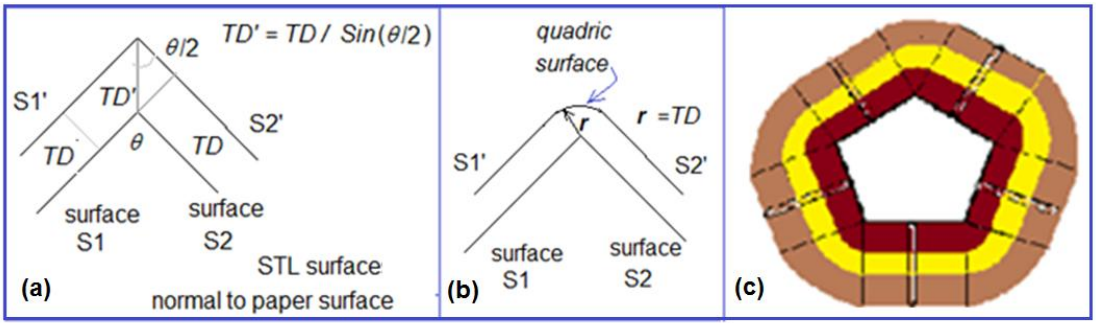



Figure 4.20: Shell formation error (a) and its correction (b) around mesh represented STL standard objects. c;The method seen in conjunction with figure 4.16

Residual error between STL and original surface increases further as new approximating triangles are larger. Error in distance is prominent at the STL nodes and it is higher for surface meeting at lower $\theta$ values (figure 4.20). Difference between $TD$ and $TD'$ increases at corners with reducing $\theta$ resulting in higher layer thickness. For the outer surface to have same distance from the corner a spherical surface segment is required. The corner needs blended surface fitting and new polygon (triangle) formation with appropriate error control. It is computationally costly and violates the requirement of using planar triangular patch only as constituent surface entity. Replacing these small quadric surfaces needs introduction of more triangular surfaces and for objects with large node numbers data explosion is possible. Hence this method does not gain support.

# 4.6 Model Simplification for Improved Implementation

The foregoing discussions emphasise the complexities in model implementation. The figure 4.20c suggests that the MPC arrangement can be seen as continuum of space with common quality zones. This encourages us to look at the MPC model in different way for the purpose of its implementation in workspace.

## 4.6.1 Recasting MPC Model

Levin and Shefer [78] observe "When developing models of a virtual sensory percept such as contact with a rigid surface, the goal should not be to model most accurately the physical qualities of the real distal stimulus, but rather to provide a perceptually adequate model of the proximal stimulus. When a perceptually adequate model of the proximal stimulus is provided, the user can then make the correct perceptual hypothesis and the appropriate distal stimulus will be inferred. Of course, a strong understanding of the perceptual qualities of the percept being modelled is a basic requirement for the perceptual design of a convincing sensation. Their study examines the rigid surface contact problem and attempts to ascertain which perceptual features are important. They observe that the sensation of viscous opposition caused while dropping in fluid can be connected to presence of bottom of container provided the subject is aware that such pools are shallow".

## 4.6.2 Physics Inspired Model

An alternative look at the MPC model reveals a very important physics phenomenon that has same effect as the MPC transduction.

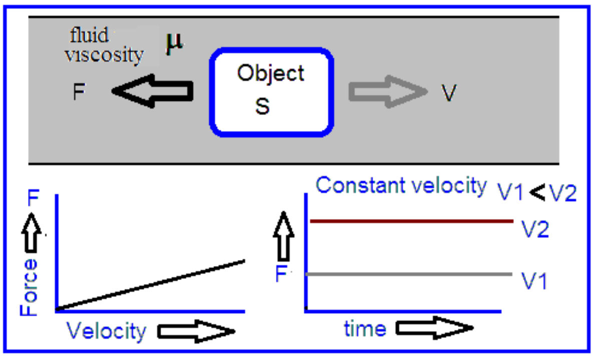


Figure 4.21: Effect of fluid viscosity on submerged body motion

A body with surface area $S$ in contact with fluid of viscosity $\mu$ faces an opposing drag force $F$ (figure 4.21) given by:-

$$F = \mu * S * V \tag{4.1}$$

Let us visualise a case where an object '$O$' is surrounded by fluids of different viscosities in manner indicated in figure 4.22. Point robot assuming they have finite non zero surfaces, shall face viscous drag that vary as $\mathcal{R}$ approaches '$O$'. $\mathcal{R}_1$ and $\mathcal{R}_2$ with same $S$ and velocities face same opposing forces. The viscosities in figure 4.22 have discrete values but it can be continuously changing too as function of distance.

$$\mu = f(d) \tag{4.2}$$

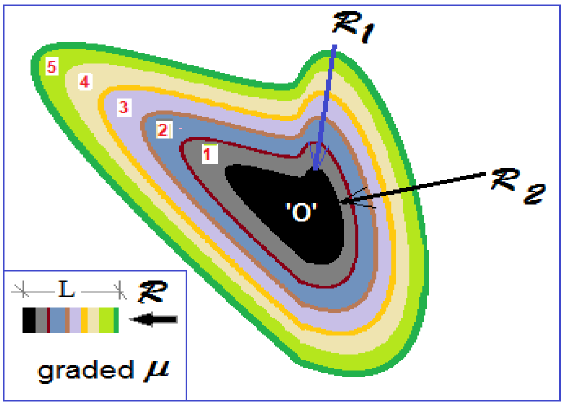


Figure 4.22: An object surrounded by varying viscosity fluid

Here '$d$' is distance from object.

For the function to assume higher value of $\mu$ in concave zones, we write

$$\mu = f(d_{\min}) \tag{4.3}$$

Here '$d_{min}$' is minimum distance from object.

Force '$F$' acting on point robot $\mathcal{R}$ located at position $P(x, y, z)$ is expressed by modifying equation (4.1) to

$$F_p = \mu_p * S * V \tag{4.4}$$

Where S is surface area of $\mathcal{R}$, $V$ is its velocity and direction of $\boldsymbol{F}$ is opposite to $\boldsymbol{V}$,

$$F_p = -1 * f(d_{min}) * S * V \tag{4.5}$$

A discrete version is considered with '$m$' discrete viscosity $\mu$ layers (figure 4.23).

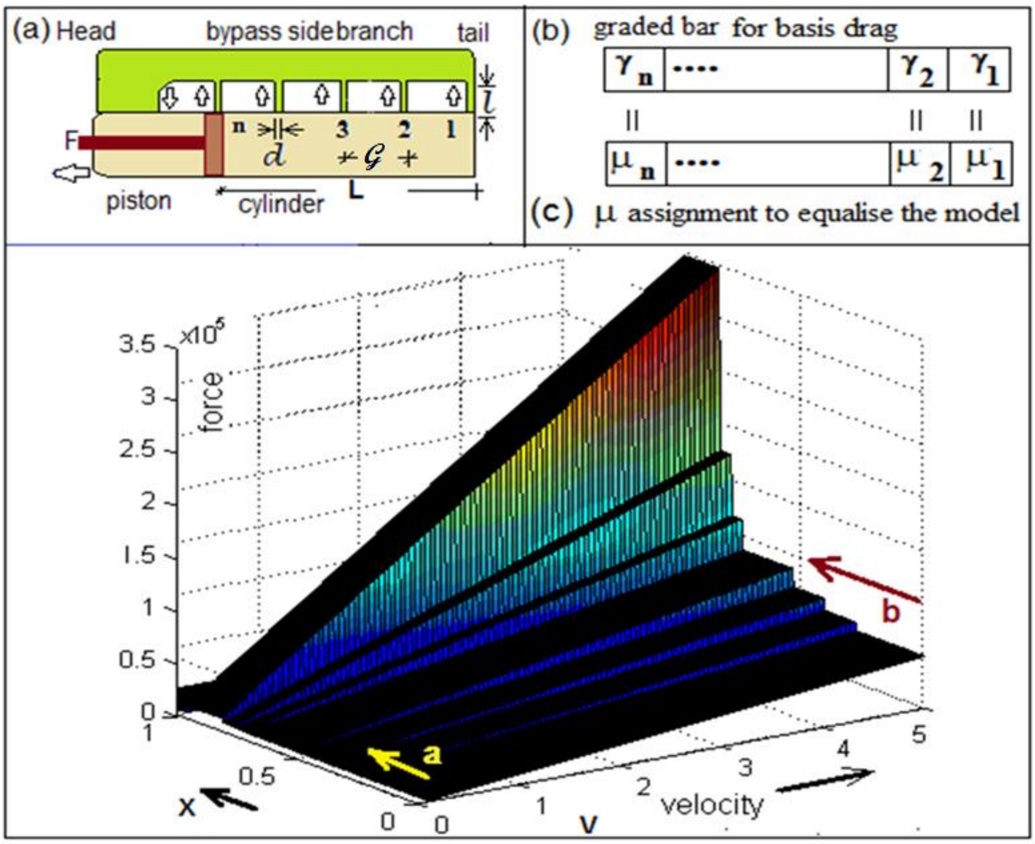



Figure 4.23: Multi-viscosity fluid generated drag force behaviour for the $\mu_{(n)} = \gamma_{(n)}$ as function of speed of approach and degree of closeness to object.

The $\mu$ are chosen with $\mu_1$ (of a predetermined high magnitude) and subsequent layers with graded viscosity such that $\mu_n < \mu_{(n-1)}$ for $n = 2$ to '$m$'. Any $\mathcal{R}$ moving at constant speed faces stepped increase in drag over the layers (figure 4.23). A slow moving $\mathcal{R}$ follows path along '$a$' sees very low drag. The drag effect is more enhanced for $\mathcal{R}$ moving at higher velocity as it moves along path '$b$'. The scheme retains the property of velocity dependence of drag. The position dependant change in $F$ can be used to generate perceivable vicinity effect with increasing nearness to a part while approaching it and the property can be used to present a kinesthetic effect on operators. Note that the resultant effect is similar to the drag effect formed by the MPC transducer. Therefore this multi-valued $\mu$ layering around a part is capable of replacing the MPC model provided the $\mu$ values are computed from the MPC model. The assignment of $\gamma$ to $\mu$ following $\mu_{(n)} = \gamma_{(n)}$ relationship converts the model to perform as shown below. For the Force creation setup like the one demonstrate in chapter 2 it can deliver same effect as MPC

$$F = -V * \gamma_{(n)} \tag{4.6}$$

$$F = -V * \mu_{(n)} \tag{4.7}$$

In above equations for the graded viscous fluid effect to be same as that of MPC

$\mu_{(n)} = \gamma_{(n)}$ has to be established by design

## 4.6.3 Conversion to $\mu$ based MPC Model

### 4.6.3a A new algorithm for 'Ranked Neighbor' forming

Section 4.5.1 suggests that object enlarging can give rise to new zones that can be treated as immediate neighbor of the object body. Taking this logic ahead further enlarging can add spatial zones as neighbor of object in form like figure 4.22.

Borrowing from morphology techniques, we can solve the problem. Morphology is a technique for the analysis and processing of geometrical structures based on set theory, lattice theory and, topology. It is most commonly applied to digital images, [79] but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures.

Topological and geometrical concepts such as size, shape, and connectivity, and geodesic distance, were introduced by this technique. These are used in image processing for erosion, dilation, opening and closing. The technique was originally developed for binary images.

The dilation of $A$ by the structuring element $B$ is defined by:

$$A \oplus B = \bigcup_{b \epsilon B} A_b \tag{4.8}$$

For avoiding reworking on the part model, we attempt at post processing type approach that starts with voxelised part models formed in earlier section.

Let us imagine a small cube formed by $(2m + 1) \times (2m + 1) \times (2m + 1)$ voxels so that it has centre voxel around which it is axis symmetric. This cube has property that when it is placed on surface of object it tries to move towards object centre till its central voxel coincides with a BV. In this state, all the voxels which are outside the object body but inside the cube can be classified as neighbour. If all the BVs of a part are visited by the cube, then entire object will have neighbour voxels which is order 1 (or rank 1). Subsequently if the cube's property is changed to react in similar fashion when it comes in contact with neighbour rank 1 voxel rather than surface voxel, another layer of next layer (rank2) can be formed. The process is detailed below to form an algorithm for voxelized 3D objects. Word 'rank' and 'order' have been used interchangeably.

Subspace of cube shape around a single chosen object is formed from the 3D voxel array. We build it by using the earlier OVA which is placed in a larger array $OVA'$ [80]. Its size is based on the extent of desired neighbourhood i.e. '$q$' from the surface. All voxels belonging to other object falling within this subspace are reset to 'empty' type. An enlarged OVA$'$ is formed to hold the layers of vicinity space voxels along with the object, by forming array corresponding to entire coded vicinity.

The range of $OVA'$ is

$OVA'\_xrange$ is $(OVA\_xoffset - m * q)$ to $(OVA\_xoffset + OVA\_xrange + m * q)$

$OVA'\_yrange$ is $(OVA\_yoffset - m * q)$ to $(OVA\_yoffset + OVA\_yrange + m * q)$

$OVA'\_zrange$ is $(OVA\_zoffset - m * q)$ to $(OVA\_zoffset + OVA\_zrange + m * q)$

$OVA'$ array is initialized to hold voxel records with the fields type as null.

Old $OVA$ records are copied to $OVA'$ as follows

$$for\ (i = 0\ to\ OVA\_xrange\ ;\ j = 0\ to\ OVA\_yrange\ ;\ k = 0\ to\ OVA\_zrange\ )$$
$$OVA'(i + m * q, j + m * q, k + m * q) = OVA(i, j, k)$$

To maintain same convention we now write $OVA'$ as $OVA$.

The $OVA$ is defined by $OVA(x, y, z)$ array and following properties

$$\{OVA\_xoffset, OVA\_yoffset, OVA\_zoffset,$$
$$and$$
$$OVA\_xrange, OVA\_yrange, OVA\_zrange\}$$

### 4.6.3b Method for Ranked Neighbor Layer Forming

The floating analyzer zone 'FAZ' is formed as 8 connected neighboring voxel set around a center voxel $FZ(\ i, j, k)$ (figure 4.24). It has $2m + 1$ layers of $(2m + 1) \times (2m + 1)$ voxel grids. This 3D zone is indexed by indices $i, j, k$ varying in range

$(-m\ to\ +m)$ along $x, y$ and $z$ directions respectively. Size of FAZ is determined by integer '$m$, around the center voxel $FZ(\ i = 0, j = 0, k = 0)$.

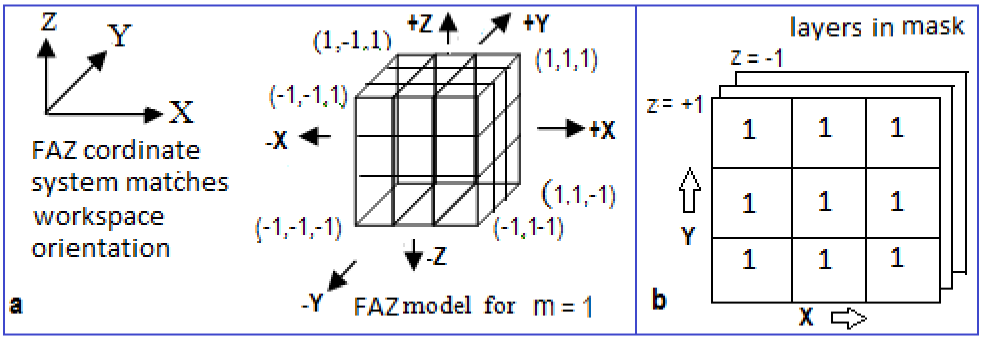



Figure 4.24: a; Floating analyzer zone and b; mask values of SE

Object voxel array $OVA_{(x,y,z)}$ is scanned. On finding a voxel with property as 'boundary', $FZ$ is aligned with $FZ_{(i=0,j=0,k=0)}$ at $OVA_{(x,y,z)}$. A search within local neighbourhood at $OVA_{(x,y,z)}$ corresponding to $FZ$ limits is carried out and for a voxel with property 'empty' the type field is a converted to 'boundary' and a parameter value '1' is assigned. For example, when $FZ$ has $m = 1$ then all $OVA_{(x,y,z)}$ voxels with 'empty' property existing within the $(-1, -1, -1)$ to $(+1, +1, +1)$ limits of $i, j, k$ are changed (figure 4.25b). This process is continued over the entire $x, y, z$ index space of $OVA_{(x,y,z)}$. The resulting $OVA$ with new 'neighbour' type in the type field and $n = 1$ in 'order' field is shown in figure 4.25a. Colour coded 'inner' (blue) and 'boundary' (gray) voxel in one horizontal layer of original $OVA$ appears in figure 4.25b. After above process, 'neighbour' order '1' (black) is also created (figure 4.25c). This completes one level of boundary grading. For growing the graded neighbour zone, the process above is repeated in phase II on processed $OVA$ by searching voxels in $OVA$ modified array data (figure 4.25a) that have property 'neighbour' and order 'n=1' and converting all 'empty' voxels within $FZ$ to 'neighbour' order 2. Repeating this modification on $OVA$ for '$n$' times, creates voxelised object with '$n$' graded neighbours surrounding it. All objects in workspace are treated similarly by forming their object voxel array $OVA$s and creating multi-graded neighbour coding in them. The number of $OVA$ created so, equals objects.

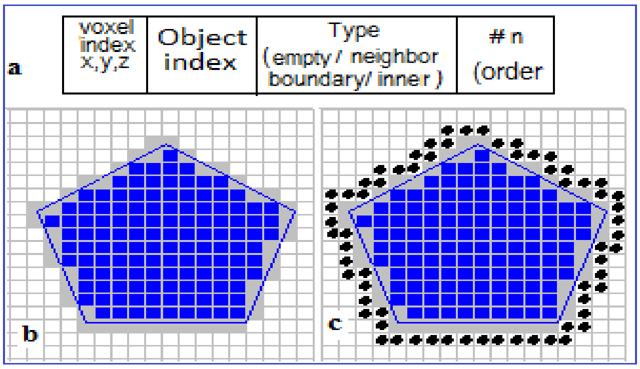



Figure 4.25: a; Data record associated with a voxel; b; the object and c; neighbour processing results on the object

# 4.7 Performance of Voxelised Neighbour Forming Technique

Two aspects of the techniques are tested for performance a: trueness of layer formation around objects for holding viscosity association and b: viscosity seen by $\mathcal{R}$ and drag generation for its movement in the neighbourhood. For appreciating the result in printed form, coloured planar slices of 3D voxelised spaces are shown.

## 4.7.1 Fidelity of Layer Forming Algorithm

Voxelised $\mu$ layers formed by using FAZ of $5 \times 5 \times 5$, $7 \times 7 \times 7$ and $9 \times 9 \times 9$ (left to right) for circular (top) and generalised cylinder of complex shape (figure

4.26). Bigger FE has utility in forming $\mu$ layers of higher thickness to cover a wide zone around object with fewer layers. The outer layers show notable change in the shape of neighbourhood and their lack of trueness when compared to the original object appearing as solid core at centre. This can be attributed to the $FAZ$ shape that is only axis-symmetric on $X, Y, Z$ axes but not iso-symmetric around $FZ_{(0,0,0)}$.

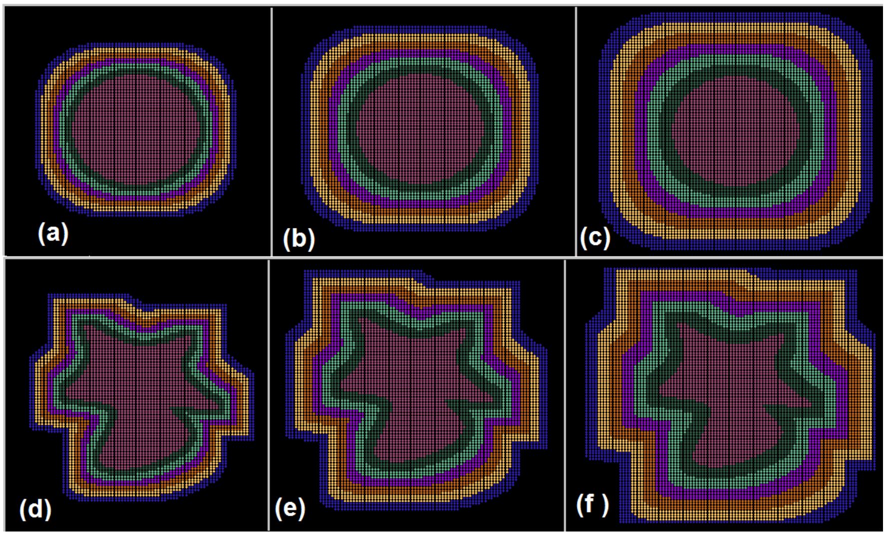



Figure 4.26: (a,b,c) and (d, e, f) show the results of neighbour layer forming for circular and generalised complex section cylinders with concavity on surface.

### 4.7.2 Improved Layering Algorithm for Higher Fidelity

The layers are required to maintain equal thickness around the object body irrespective of object shape. Equal thickness layer is high fidelity layer. This needs reshaping of $FAZ$. In order to effectively shape $FAZ$ in iso-symmetric form the $FAZ$ is now modified to hold binary data at its cells (figure 4.24b). The value of $SE\ (i, j, k)$ may be '1' or '0' for forming shaping element other than cube while maintaining high efficiency array based ordered processing. A '0' inhibits processing at the location. The algorithm of previous section is modified by permitting the boundary voxel conversion to neighbour type only where the SE element value is '1'. For giving the SE a spherical shape only those cells of SE that are within radius of size 'm' from the SE centre $SE_{(0,0,0)}$ are loaded with '1' value at the beginning of the process and others remain '0'.

### 4.7.3 Experimental Results with Spherical Shaping Element for Improved Layer Fidelity

An FE of sphere shape was formed in $5 \times 5 \times 5$ cube by setting mask values as 1 in SE for voxels corresponding to interior of sphere of radius 2 voxels. Radius can be only integer values. Relative performance of the approach is shown on circular cylinder and complex cylinder for 6 layers (figure 4.27). In low order $\mu$ layer difference is not appreciable but as layering reaches higher order, improvement is amply clear in a2 and b2. In figure b2, narrow concavity at *A* filled up thus discourages $\mathcal{R}$ to enter such areas while approaching a workspace part. Wider ones

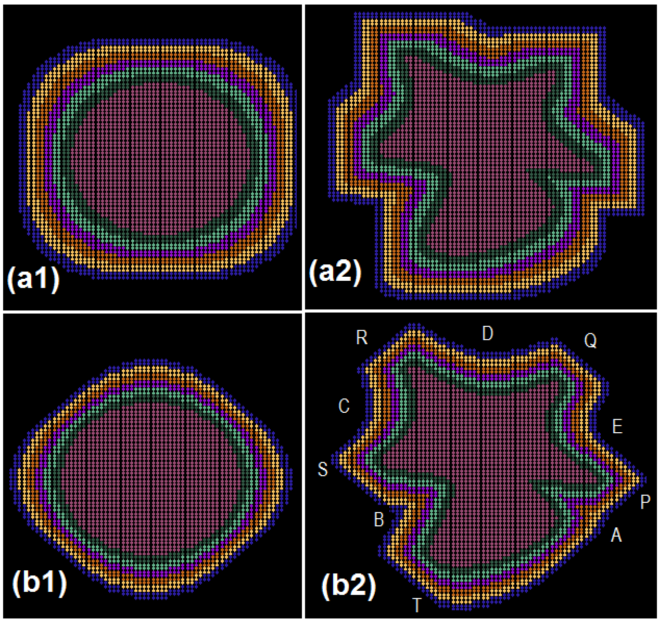



Figure 4.27: relative performance of cube SE (a1,a2) and sphere SE (b1,b2) for *m=2* on a circular & complex cylinder objects respectively.

like $B, C, D, E$ shapes are maintained as these allow free space to remain in-tact for $\mathcal{R}$ to move freely in workspace even with wide μ affected area around object. Using m=3 and m=4 i.e. sphere contained by FAZ $7 \times 7 \times 7$ and $9 \times 9 \times 9$ improved layer formation up to 6 layers are achieved as shown in figure 4.28.

The case shows complex cylinder with concave surfaces, axis parallel to z, bottom planar surface at $\mathrm{Z} = 20$ top at 60. (b) shows voxelisation test results for spherical SE at middle of object height on complex section 3D cylinder for $\mathrm{m} = 3$ in FAZ. (c) shows same object with different neighbor layers using SE formed in FAZ of $\mathrm{m} = 4$ .

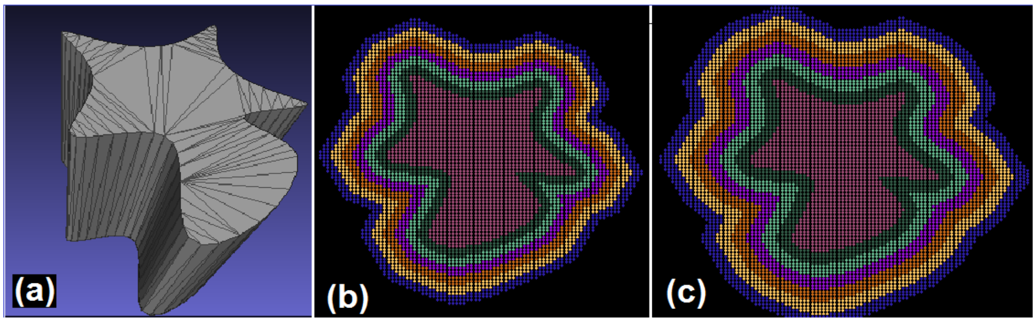



Figure 4.28: a; Complex cylinder with concave surfaces, b; layer formation with m=3 c; layer formation with m=4

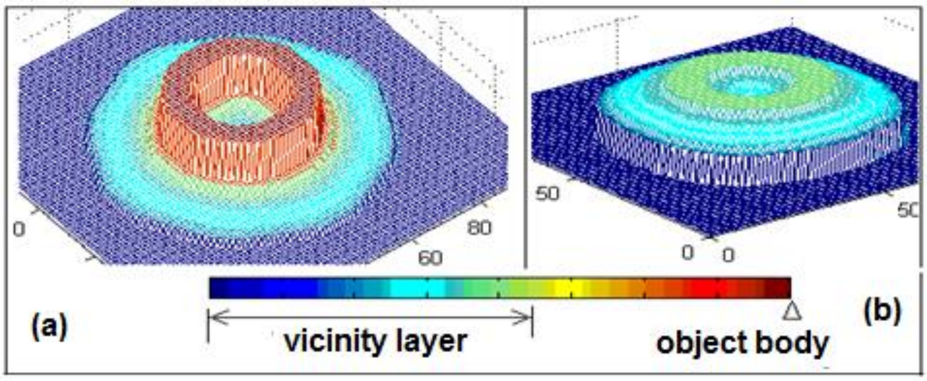



Figure 4.29: a; vicinity coding at middle of hollow cylinder (red) height. b; result at $Z$ above top of cylinder. Voxels here are neighbour voxels from body below.

A hollow cylinder was voxelized and the vicinity space was graded with acceptable fidelity (figure 4.29). Voxelised ranked vicinity formed for hollow cylinder appears in (a) representing vicinity coding at middle of hollow cylinder height. Red is cylinder body. In (b) result is for location at $\boldsymbol{Z}$ above top of cylinder. Voxels here are neighbour voxels from body below. It allows a small object to pass through hollow axial space

### 4.7.4 Comparison of Layer Formation for Different Type of Object

In earlier tests, complex surface object was used. Experiments were carried on planar faceted wedge placed in workspace with '$V$' groove up (figure 4.11a). The spheres were formed in FAZ with $m = 3$ and 5 layers were formed. Ranked neighbouhood was formed for wedge (figure 4.30a) by ranked layering around its voxel version figure 4.13 and the result is shown in figure 4.30,(a),(b) and (c).

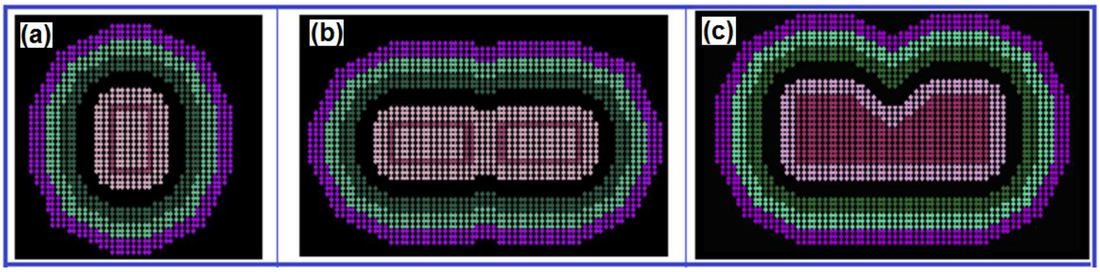



Figure 4.30: Voxelization of wedge object shown in figure 4.13 with 5 layer.

as seen from $x, -z$ and $y$ directions respectively. The plan view (middle) shows section at height above $V$ junction of the block. The test results establish the correctness of layer formation. Note that objective of constant width layer creation has been met successfully by the method developed here.

### 4.7.5 Capability of Developed Technique for high Fidelity Layer Formation

Layer formation by different $FAZ$ size gives different performance. In horizontal slices (figure 4.27), these are visible. For testing on the orthogonal direction, a cone specimen has been used (figure 4.31). For cone voxelization, the test results are shown in vertical sections (a1 to a4) through voxelised layers formed by spherical SE in FAZ size $m = 1, m = 3$ and $m = 4$, Figure 4.31 (b1 to b4) show horizontal sections on H1 ($z = 20$ voxel) to H4($z = 40$ voxel) heights for spherical SE mask with $m = 2$; figure (c1 to c4) show horizontal sections on same height but for spherical SE mask $m = 4$. Fidelity of layer formation is better for higher '$m$'.

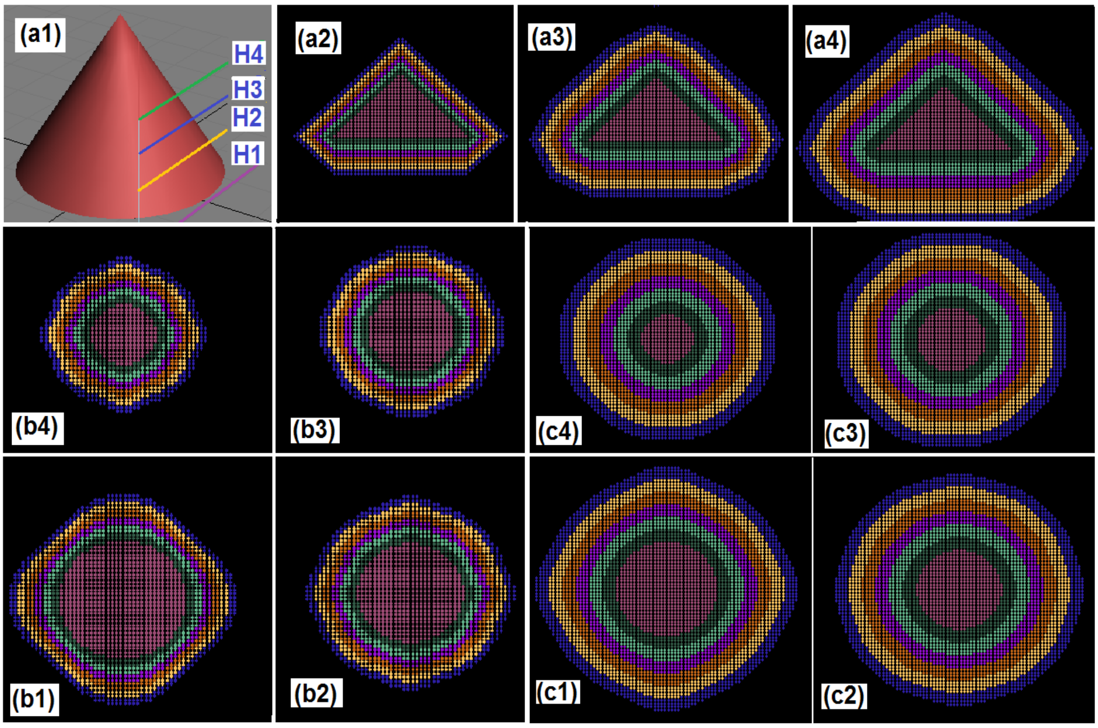



Figure 4.31 (a1) a cone object in workspace and its voxelization tested as sections along vertical plane passing through vertex (a2 to a4) and horizontal orientation (b1 to b4) and (c1 to c4).

It is evident from these results that use of higher '$m$' leads to better layer shapes. This can be attributed to better sphere approximation by higher '$m$' $FAZ$ and corresponding bigger $SE$.

**Table 4.1**: Performance of layer forming techniques.

| No. | FAZ | SE | Size | Observations on layer fidelity |
|---|---|---|---|---|
| 1 | 3 | Cn3 | 27 | Layer size thin, shape distortion in diagonal directions high, redundant computation low |
| 2 | 5 | Cn5 | 125 | Layer size medium, distortion in diagonal directions high, redundant computation high |
| 3 | 7 | Cn7 | 343 | Layer size wide, distortion in diagonal directions high, redundant computation higher |
| 4 | 3 | SR1 | < 20 | Layer size thin, shape distortion moderate, % redundant computation low. |
| 5 | 5 | SR2 | < 108 | Layer size medium, distortion low,% redundant compute lower |
| 6 | 7 | SR3 | < 250 | Layer size wide, shape fidelity and % redundant computation lower than case 5, |

Note: Shaping element C- cube, n- size of cube side in voxel, S approximated spherical shaping element R radius of sphere for spherical shaping in voxel, Size - number of cells in shaping element.

## 4.8 Performance of Layered Voxel Model for Drag Generation

$\mathcal{R}$ moving in the vicinity of workspace part with ranked μ coded space shall develop a drag force as per equation (4.7). The ranked zones can have μ values assigned to them as per a linear or non linear relation with the rank of the zone generally satisfying the concept developed earlier (figure 4.23). In an experiment, a cone (figure 4.30a1) was voxelised and its neighbourhood was formed with 6 ranked $\mu$ zones with $\mu_{(n)}$ as 32, 25, 20, 17, 15, 13 for $n = 1$ to 6 respectively. The ranked $\mu$ zones have been formed by using approximated sphere $SE$ in $FAZ$ with $m = 3$ as shown in figure 4.24. Mark that the $\mu_{(n)}$ coding is non linear function of $\mu$ layer distance from object surface.

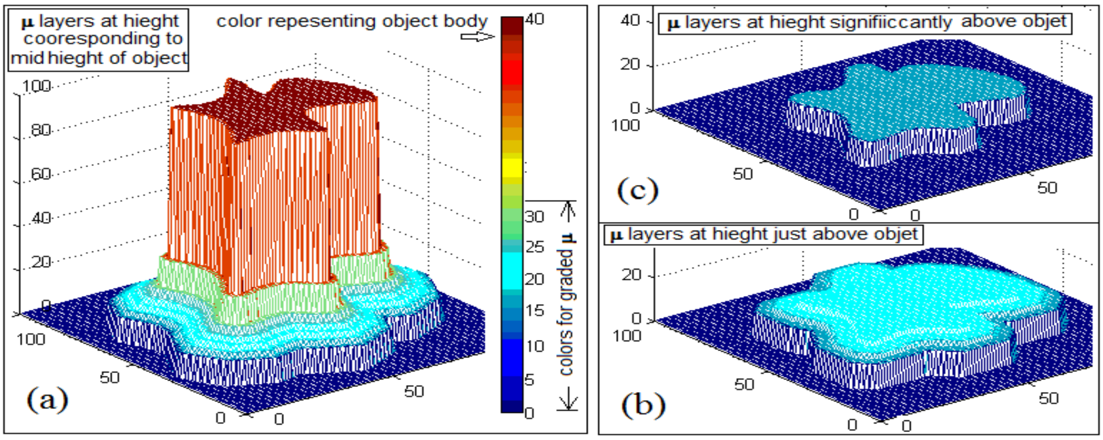



Figure 4.32: $\mu$ assignment to vicinity at different heights near middle and top of the cylinder body.

In figure 4.32, the complex cylinder object has been shown in red. In (a) a single horizontal vicinity layer corresponding to half the height of the cylinder has been coded with $\mu$ and graphically elevated proportional to $\mu$ values to show the work space activation correctness. In (b) a single horizontal vicinity layer corresponding to top of the object has been coded with $\mu$ and similarly shown as above. Mark that layers up to third order are only seen. Being at the top of the object the $\mu$ layer has submerged the object body in display. **c;** A horizontal slice coded with $\mu$ at height higher than case B above object top. Note that being away from top surface; the section is reaching till fifth layer of vicinity of top surface only.

Test probes were moved at linear locus corresponding to different values of $y$ at a $'z'$ level of 40. The corresponding $F_p$ generated is plotted for the moves. The results are as following:-

The $F_p$ values seen by $\mathcal{R}$ (figure 4.33) are stepped in nature. The stepped effect is beneficial in some cases as differential changes in force feedback are recognised without fail by an operator. But the elimination of effect is also possible if desired.

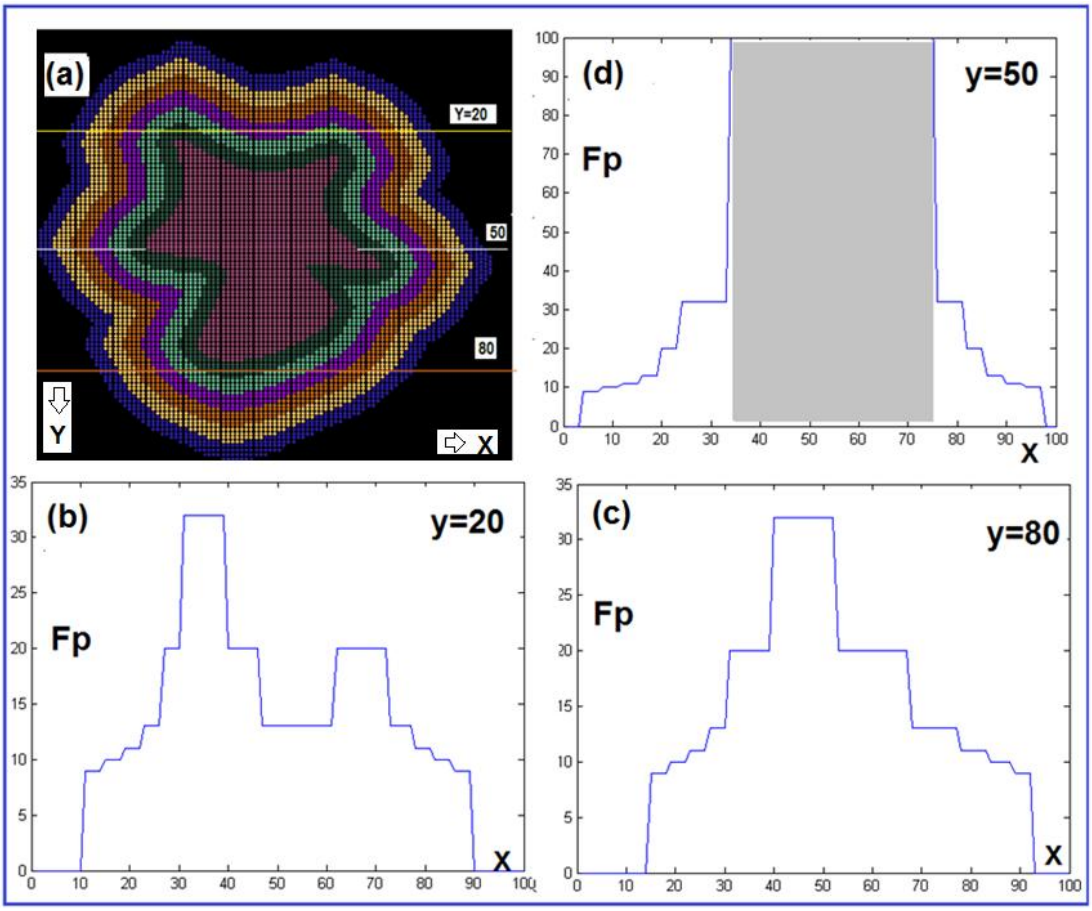



Figure 4.33: $\mu$ values around a complex cylinder object at a level $z = 40$. (b,c,d) show the $F_p$ generated for a point robot on three tracks at $y = 20, y = 80$ and $y = 50$ respectively. Note , in (d) i.e. for $y = 50$ $\mathcal{R}$ passes through object body.

a) Horizontal section ($z = 40$) of complex profile cylinder with $\mu$ layers formed by sphere $SE$ approximated in $9 \times 9 \times 9$ sized $FAZ$. The $\mu$ values in layers are 32, 20 13, 11, 10, 9 starting near surface which has a value 100,

b) $F_p$ variation seen by $\mathcal{R}$ moving very close to object but not touching it ($Y = 20$). Mark the two peaks,

c) drag seen by $\mathcal{R}$ moving at $Y = 80$, d: drag on path at $Y = 50$, note that on contact with body (shown as gray) it assumes high opposition owing to $\mu$ value corresponding to skin.

### 4.8.1 Velocity Effect on Drag '$F_p$'

Velocity effects were tested for a cone object where in the probe graze past the cone without touching at the constant height on linear vector along $x$ direction. figure 4.34 shows the effect of velocity on $F_p$ generation in same neighborhood on same path but at different speed.

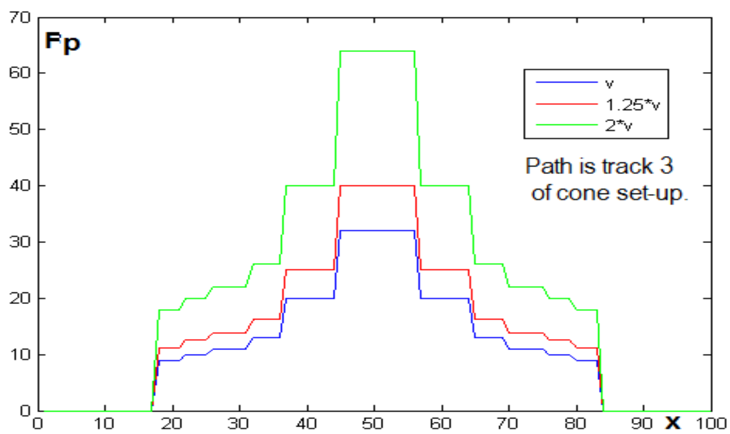



Figure 4.34: Relative '$F_p$ ' faced by $\mathcal{R}$ at voxel position 'x' during move along a straight path parallel to x axis near cone at different speeds .

### 4.8.2 Stepped $\mu$ Change and Its Remedy

The $F_p$ values seen by $\mathcal{R}$ are stepped in nature owing to layers being multiple voxel wide. Single voxel wide $\mu$ layers can give smooth $F_p$ but causes shape fidelity loss. The obvious option of reducing voxel size leads to data explosion conditions. A technique has been developed to produce relatively smoother $F_p$ generation without data size increase. A circular memory buffer is formed between a high limit (HL) and low limit (LL). It holds '$k$' latest encountered $\mu$ (figure 4.35).

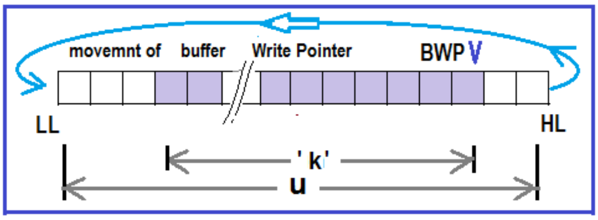



Figure 4.35: Circular buffer for memorising recently encountered μ.

During a move whenever $\mathcal{R}$ encounters a new voxel, it's $\mu$ is recorded using buffer write pointer BWP and BWP is advanced. Only one buffer is used for one $\mathcal{R}$. The maximum buffer depth is $\geq m$, here $m$ value corresponds to biggest $FAZ$ used for any object in the Cartesian workspace. This ensures that in an implementation with varied '$m$' of $FAZ$ for different objects too, the buffer is sufficient for '$\mu_e$' computation using –

$$\mu_e = \frac{1}{k} * \sum_{i=0}^{k} \mu_{(BWP-i)} \qquad (4.9)$$

Mark that only '$k$' number of data is used from current value backward and higher depth of buffer does not affect the correctness of equation (4.9) but increases memory usage. Shaping of the filter can be developed by assigning weights to the past $\mu$ values in following manner

$$\mu_e = \frac{\sum_{i=0}^{k} \alpha_k * \mu_{(BWP-i)}}{\sum_{i=0}^{k} \alpha_k} \tag{4.10}$$

### 4.8.3 Case Based Analysis of Smoothening Processes

Experiments have been carried with $k = 1, k = 10$ and $k = 3$ and constant $\alpha_k$ as well as varying $\alpha_k$. Two sets of experiments for linear as well as nonlinear $\mu$ coding over the ranked neighbour layers are analysed. The 'drag' profiles for the modified algorithm are shown in figures 4.36 and 4.37 as per following test cases-

Cases for linear μ coding.

Case-I.a- No memory ($k = 1$, $\alpha = 1$), linear μ coding.

Case-II.a- Long sustained memory ($k = 10$, $\alpha_1$ to $\alpha_{10} = 1$), linear $\mu$ coding.

Case-III.a- Fading memory ($k = 10$, $\alpha_1$ to $\alpha_{10}$ =1, 0.9, 0.75, 0.55, 0.30, 0.20, 0.1,0.05, 0.02), linear $\mu$ coding.

Case-IV.a- Short sustained memory ($k = 3$, $\alpha_1$ to $\alpha_3$ = 1,1,1) linear μ coding.

Cases for nonlinear $\mu$ coding

Case-I.b- No memory ($k = 1, \alpha = 1$), nonlinear $\mu$ coding.

Case-II.b Long sustained memory ($k = 10$, $\alpha_1$ to $\alpha_{10} = 1$), nonlinear $\mu$ coding.

Case-III.b- Fading memory ($k = 10$, $\alpha_1$ to $\alpha_{10}$ =1, 0.9, 0.75, 0.55, 0.30, 0.2, 0.1, 0.05, 0.02 ), nonlinear $\mu$ coding.

Case-IV.b Short sustained memory ($k = 3$, $\alpha_1$ to $\alpha_3$ = 1,1,1) non linear $\mu$ coding.

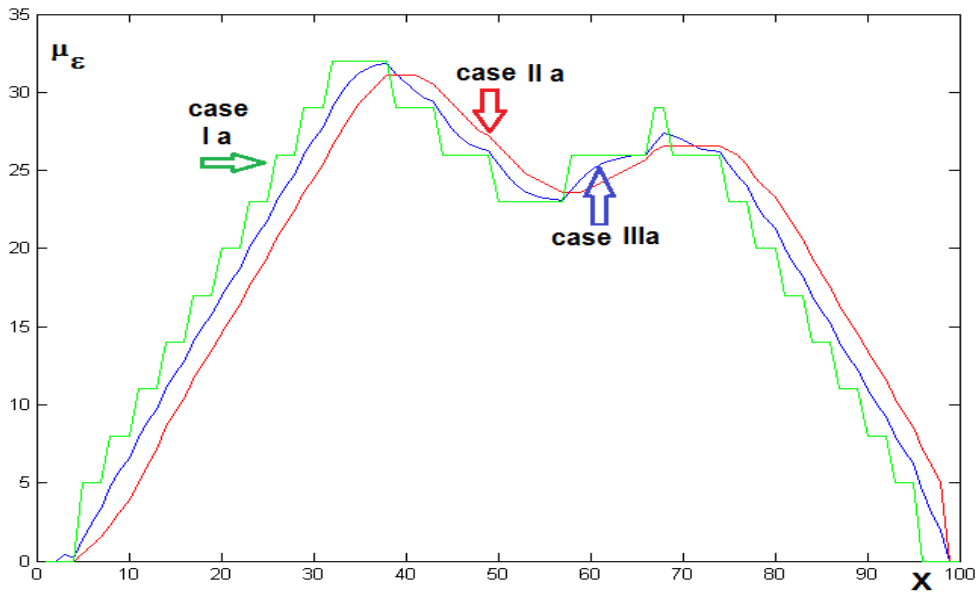



Figure 4.36: results of traverse along position in voxel 'x' in linear $\mu$ coded ranked neighbour(figure 4.33 $y = 20$) and smoothing by cases Ia, IIa, IIIa. Refer Table 4.2, serial no, 1,2,3

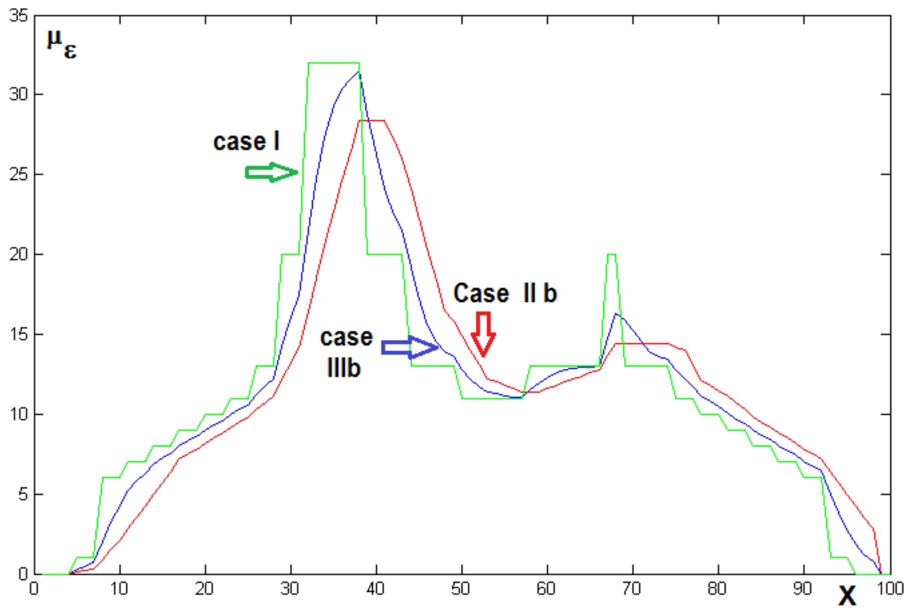



Figure 4.37: results of traverse along position in voxel 'x' in linear $\mu$ coded ranked neighbour (figure 4.33 $y = 20$) and smoothing by Ib, IIb, IIIb. Refer Table 4.2, serial no, 5,6,7

**Table 4.2:** $\mu$ sensing performance with different memory depth

| | **μ** | **case** | **Effective μ behavior** |
|---|---|---|---|
| 1 | I | I.a | Large steps of $\mu_{e,}$ gradual increase closer to object. Operator feel jerky. |
| 2 | I | II.a | Smooth $\mu_e$ gradual increase closer to object, smooth drag generation Delay large, may miss narrow features |
| 3 | I | III.a | Smooth $\mu_e$ gradual increase closer to object, Feel to operator smooth. Response delay moderate |
| 4 | I | IV-a | Fine stepped $\mu_{e.}$ Gradual increase closer to object, feel to operator fine –like notch move, easy to feel, No delay in response (figure 4.36). |
| 5 | II | I.b | Large steps of $\mu_e$ steep increase closer to object. Vicinity change too jerky. |
| 6 | II | II.b | Smooth $\mu_e$ gradual increase closer to object, smooth drag generation Delay large, may miss narrow features. |
| 7 | II | III.b | Smooth $\mu_{e,}$progressive rate increase closer to object, force feel to operator smooth and closeness dependent rate of change. Response delay moderate. |
| 8 | II | IV.b | Fine stepped $\mu_e$ forms comb rub feel. Progressive increase of rise rate closer to object, operator feels fine – taps feel like rubbing on comb-teeth, easy to feel degree of closeness owing to increasing difference magnitude at successive voxels. No delay (figure 4.37) |

**Note**: $\mu$ coding type I linear (32,29,26,23,20,17,14,11, 8,5) Type II nonlinear (32,20,13,11,10,9,8,0,0,0,0) Cases are defined based on memory property

# 4.9 Simplified MPC Function as Haptic Assisted Vicinity Percept

Previous sections establish effectiveness of voxelised and $\mu$ activated workspace model for emulating graded (static) viscous layers and its drag effect on $\mathscr{R}$ for known velocity. Extending the model by placing a kinematics equivalent slave arm in the same environment (figure 4.38) and matching its instantaneous spatial state to that of slave, subjects an attached point on the robot model to viscous drag environment in equivalent form. This point represents $\mathscr{R}$. The emulated viscous drag effect for any point on robot body and its reflection on master can be computed by method outlined in figure 3.7. Torque transducers on robot joints can realise it in physical form. The resulting kinesthetic effect function as Haptic Assisted Vicinity Percept 'HAVP'

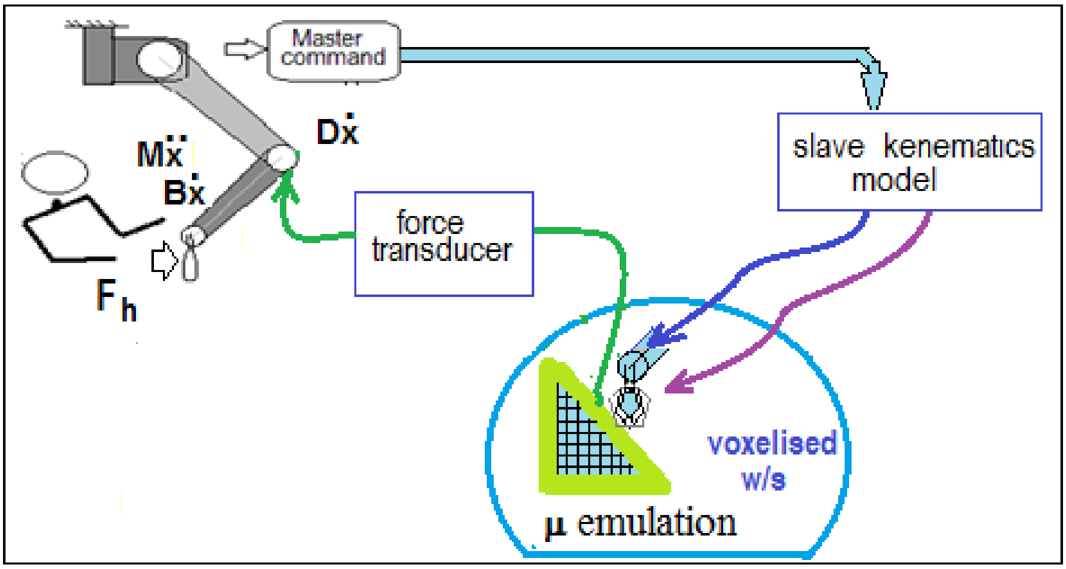



Figure 4.38: The percept formation schema for sensitizing operator to object vicinity in green zone around object by simplified MPC model.

## 4.9.1 Experimental Verification

A Single complex object with '$\mu$' layers has been used and a tool tip of the cartesian robot used in chapter 3 experiments has been used. The robot has 5 degrees of freedom. Tool rotations are fixed at $\theta = -90$ and $\phi = +90$ degrees, $x$ and $Z$ has been set so that the tip passes though the $y =$ 18 cm, $Z$ $=$ 30 cm line parallel to $X$ axis

## 4.9.2 Experimental Set-up

The experimental setup is a software based model of cartesian workspace of the same robot as used in chapter 3. Same AJS is used for developing force feel on its handle.

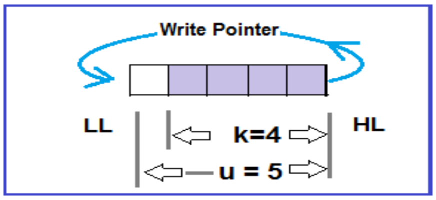



Figure 4.39: Filter buffer design for experiments

In experiments the $k$ value has been chosen to match layer width in terms of voxels to get optimum smoothing performance (figure 4.39). The '$u$' value has been made just larger than '$k$' to avoid initial buffer fill delay in the filter action.

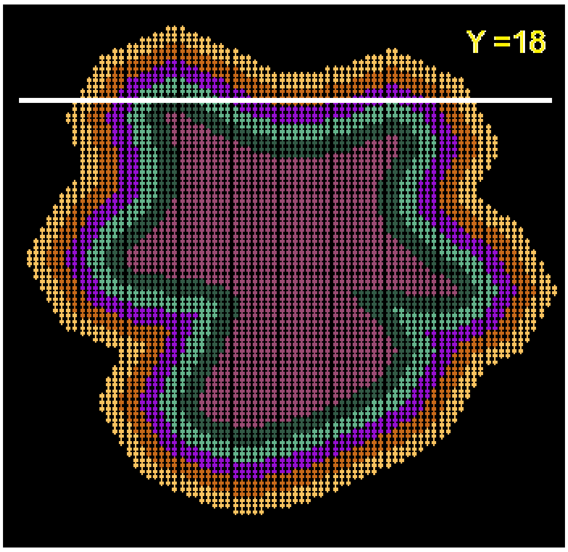



Figure 4.40 Test path of $\mathcal{R}$ for complex cylinder with 6 $\mu$ layers using $m = 4$

The test scheme is same as in section 3.1.3 chapter 3. The online 'drag' formation process used in chapter 3 is implemented in software in the test bench. The Cartesian workspace is 3D linear array of loaded voxel values as per the graded $\mu$ coding. The tool tip movement is simulated for the linear locus (figure 4. 40) in experiments. Test specimen is a complex section cylinder, with graded viscosity layers of thickness 4 voxels. The tool tip locus is at $y = 18$ and at fixed $z$. Move is at constant velocity. The layered workspace is loaded with multiple schemes of $\mu$ coding.

### 4.9.3 Results and observations

Workspace is loaded with multiple schemes of $\mu$ coding in surrounding layers of varying numbers as indicated in result description. In first experiment relative performance for stepped and fine stepped $\mu_\varepsilon$ for cases I.a and IV.a for linear μ coding (32, 29, 26, 23, 20,17,14,11, 8, 5) was simulated. The fine stepped response (gray bars) are very close to native values and produce ***comb rub feel*** however the drag rise rate is constant. The relative performance for stepped and fine stepped $\mu_\varepsilon$ for cases I.b (red) and IV.b (gray bars) are shown in figure 4.41.

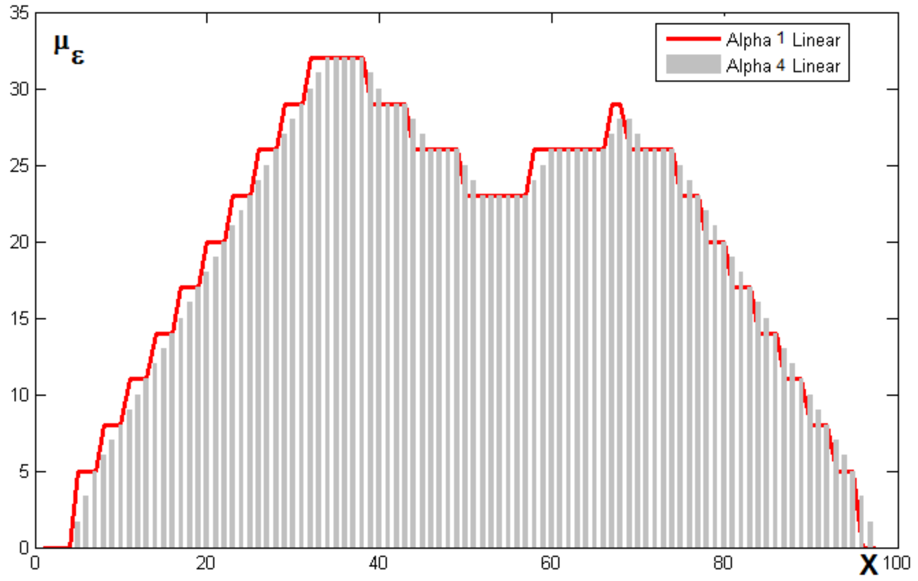



Figure 4.41: Relative performance for stepped and fine stepped $\mu_\varepsilon$ for cases I.a legend 'Alpha 1 linear' and IV.a. legend 'Alpha 4 linear'.

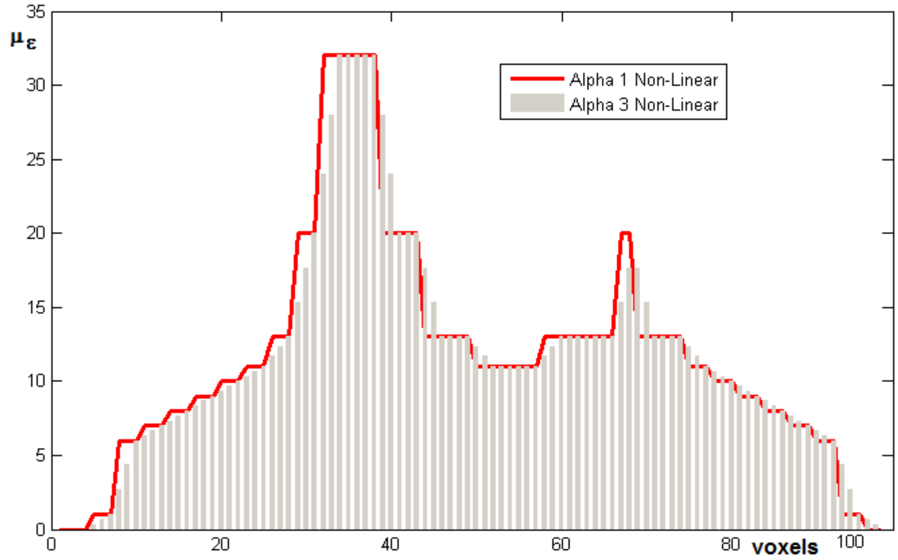



Figure 4.42: Relative performance for stepped and fine stepped $\mu_\varepsilon$ for cases I.a legend 'Alpha 1 non-linear' and IV.a. legend 'Alpha 4 non-linear'.

For nonlinear μ coding ( $\mu_1$ to $\mu_6$ 32, 20, 13, 11,10, 9) the fine stepped response (gray bars) is very close to native values and produce comb rub feel. Drag build up near object is steep.

### 4.9.4 Observations on Linear $\mu$ and Nonlinear $\mu$ Coding

In the experiments the drag performance of the viscosity emulation by linear coding and non linear coding are significantly different. The distance to object is different for the two locations on $y = 20$ that form peaks of drag in figure 4.42 but the separation of peak is enhanced by nonlinear coding of neighbour viscosity (figure 4.43). Mark the distinct rapid rise in case of nonlinear coding. For operating the master device in the vicinity the operator needs to work against the AJS that itself consumes power in producing the opposition. Nonlinear coding results in energy saving during the $\mathcal{R}$ traverse on the path too. The energy saving is proportional to the area between the respective drag plots of the two sets of curves for the linear and nonlinear $\mu$ coding. The nonlinear $\mu$ coding sensitizes the operator at outer periphery or vicinity by moderate rate of $F$ increase. The operator can operate at low velocities. Very near objects, the $F$ increase is very rapid and so a slow operator again faces high rate of $F$ change in opposition signifying object presence in very close vicinity.

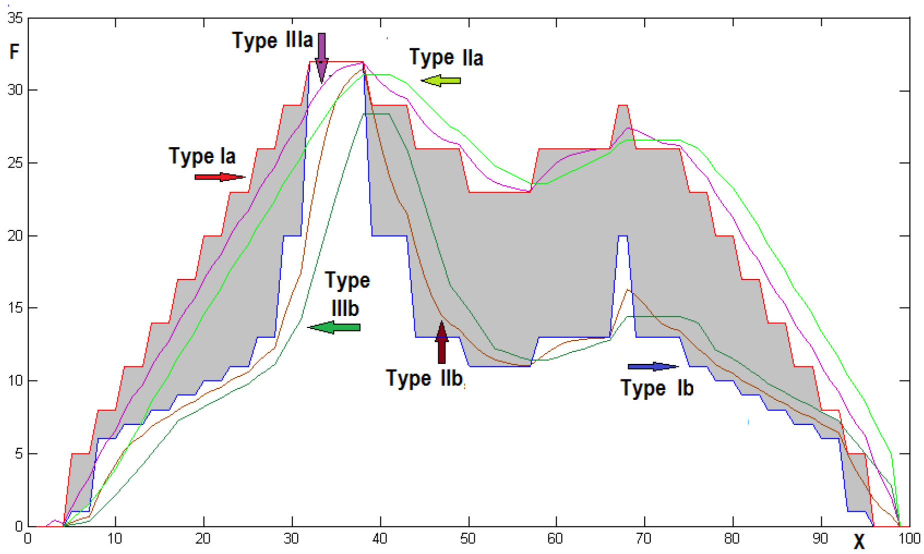



Figure 4.43: Comparative performance of linear (upper cluster) and nonlinear viscosity coding (lower cluster) of the ranked neighbour layers in forming force feedback. '$x$' denotes positions along traverse in voxels.

# 4.10 MPC Model's Effectiveness as Percept

The model performance is oriented to forming monotonically increasing rate of change of drag force as the robot reaches closer to object. The model is very successful in early sensitisation of human operator with low step opposition and forms stronger discouragement closer to object on light balanced joystick like AJS. The response being symmetric i.e. same on approach and distancing away, the feel is 'sticky' while moving away, but the nonlinear coding reduces the effect to a large extent by quick fall in drag effect while moving away. It should also be noted that for moving away, generally a change of direction is required which makes velocity fall and so stickiness is greatly reduced. MPC model being basically position based discrete phenomena, replicated in simplified version too by discrete definitive viscosity zones, also produces frequency modulation wherein a fast moving point receives more frequent steps of drag increase and on slowing the frequency as well as the step magnitude reduces.

# Closure

The viscosity emulation process needs precise spatial modeling of vicinity of complex surfaces of real objects. This has been achieved in this work. Real time sensing of the local viscosity property assigned to a location, where point of consideration on robot body passes, must be accurate and fast. Modeling of precise shape of real workspace object is of utmost importance and necessitates use of accurate CAD models for assigning physics property. Unlike the potential maps [20] that are approximate, the viscosity assignment is precise.

Though models with moderate size voxel (upto a few millimeter in size) are not suited for texture modeling in tactile feel based haptic applications and graphic modeling, it is very effective in the kinesthetic feel development aimed here. The arrayed Cartesian workspace compliant voxel based workspace coding approach with the moderate sized voxels pays by reducing memory requirement while exploiting the efficient array based accessing which give advantage in adjacent space accessing for layer formation. The voxelised viscosity layer forming technique developed here and varying viscosity emulation achieved in this work, are the key to the success of developing a vicinity perception in interactive master driven teleworking. The method is efficient and amenable to automated viscous layer formation from CAD surface models without any shape constraint. The fidelity of ranked layers is superior when spherical forming element of moderate and large cell cluster is used. The consequent accentuation of stepped drag effect is also solved by using filters. The ranked viscosity model shifts most of the computations to off-line pre-processing phase and real time computation burden is minimized. The dependence of the drag phenomena on robot velocity and nearness to object surface makes the technique strong candidate for force absorption based active interface in 'man-in-loop' telecontrol of remote manipulators too. The perception developed by physics emulation is novel in nature and functions successfully in pre-contact phase in position forward modality of teleoperation. It is promising method for incorporation in telecontrolled robotic system for forming a sense of presence in operator's mind through human kinesthetic sensory faculty.

# CHAPTER 5

## A PHANTOM FRAMEWORK

## FOR

## SENSOR BASED PERCEPT SYNTHESIS

## AND

## CO-WORKING WITH BILATERAL TELEMANIPULATOR

For the tele-manipulator system to offer enhanced perception that effectively enables human operator to perceive presence of a nearby object in workspace through kinesthetic means, the haptics based perception developed by MPC model has to be generated for a slave robot arm and rendered in physical form on-line while the robot arm works under the master side control. In chapter 4 the implementation of MPC model on a point robot has been established but to form a generic method a structured approach is required.

# 5.1 Considerations for Percept Synthesis

The point sensor developed in chapter 4 is capable of sensing drag and as demonstrated in chapter 3 the MPC model's use for creating a haptics assisted vicinity feel is feasible through a force transducer. The challenge lies in developing a complete sensing framework where the sensor detects the viscous drag on the robot while it travels through the viscous medium. The drag being property of the workspace represented in Cartesian form, the sensor function has to be fast. On the other hand the haptics effect of the drag has to act on interface to human operator with considerations to compatibility with other force effects prevalent in BTM. Following sections are devoted to this objective.

## 5.1.1 Haptic Feedback to Operator

The drag sensed by a point sensor can be presented to user through haptic rendering on a user interface as a proximal percept that suggests nearness and its rate of change. The interface is a key factor in desired function of the percept.

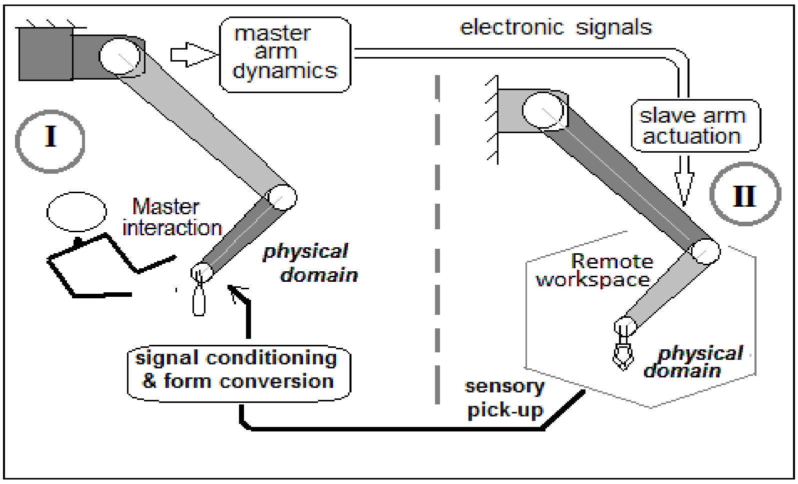



Figure 5.1: General bilateral manipulator: The Master side sends position signals to slave and receive force feedback in one of several ways.

The force feedback interfaces are of two varieties: ground-based and body- based. Body-based (typically gloves and exoskeletons) use a connection point on another part of the wearer's body to provide the force. Body based devices are portable owing to no requirement of attachment to the world (i.e., ground), but this property of portability can be a drawback in many cases as a portable exoskeleton does not strive to stop the wearer from passing through a virtual wall.

User interface like the AJS, introduced in chapter 3, has been shown to create a well perceived haptic feel on user's hand and so works for rendering the proximal percept for a nearing object in workspace. The strength of the percept and its rate of change are related to nearness and rate of change of nearing of a moving sensor to workspace object. This sensor can be used to sense the virtual fluidic drag in the activated voxelised workspace. A corresponding interface can render it as real physical effect to the operator (figure 5.1) and achieve the function of a proximal percept as originally intended to. However the percept must fulfill two requirements

i. It must sense the drag in workspace as sensed by the slave arm

ii. For physical rendering, it should use the master arm joint that receives the force feedback from the slave arm and also sense the operator input to form command to the remote slave (figure 5.1).

### 5.1.2 Working of Conventional Bilateral Tele-manipulator

Our target BTM systems have same kinematics configuration of master and slave. The link lengths too are same. For such configuration, the environment forces acting on a slave joint, through its child link, if reflected to the corresponding joint on master arm with adequate fidelity, forms a very effective force feedback to the master side operator.

The BTM force feedbacks use ground-based approach [81, 82]. The operator side arm is a grounded device as the force closure happens through the frame that hosts the master arm as well as the operator. Seen in conjunction with BTM functionality, the problem of co-working of the MPC based transduction technique shall depend on force feedback modalities.

## 5.2 Development of a ‘Phantom’ Framework

The layered voxelised model of MPC has been demonstrated to produces a sensible percept capable of conveying nearness of an object to a point robot functioning as a haptic probe in chapter 3. A mechanism for fast synthesis of the percept by the workspace activation has been developed in chapter 4. However this phenomena needs to be embedded in a structured framework to address sensing in workspace having multiple objects and achieve emulation of sensing process as would occur on a functional robot body immersed in the activated workspace. The structured framework is referred to as 'Phantom' in following sections.

A generic sensory frame-work development is attempted here for pre-contact perception development. The phantom system is a mediating computer platform forming segment III (figure 5.2). It functions between master side (segment I) and slave side (segment II) of tele-controlled system (figure 5.1). The approach essentially forms a slave robot model using its kinematics equivalent [40] and maintains it in same spatial state in phantom's environment as the slave state in remote work environment. The phantom's environment is a virtual environment that represents all the manipulated objects, fixed objects and the workspace structural features using their CAD models instantiated to their current spatial state in

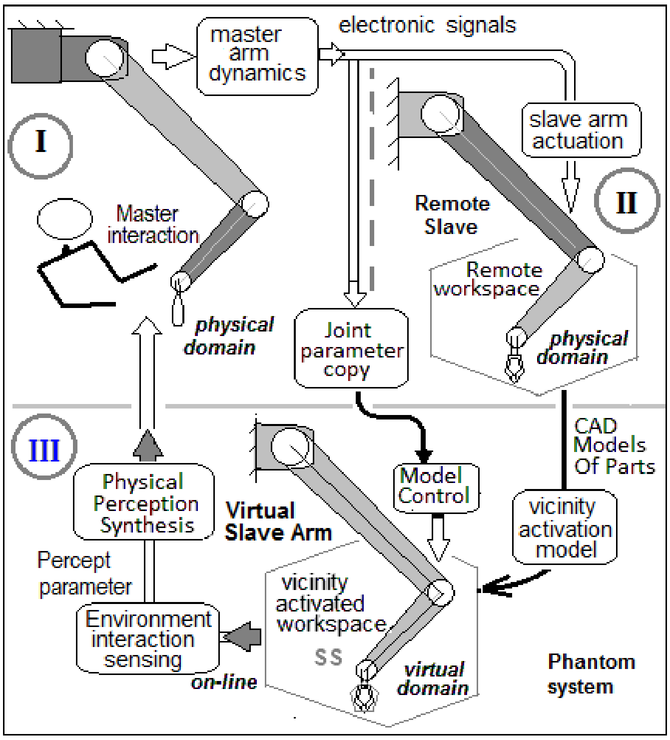



Figure 5.2: 'Phantom-I' environment formation by mediator system.

workspace albeit with added properties as described in chapter 4. The virtual slave hosts sensors, on its body at desired locations of interaction sensing and thus the virtual sensor inherits same Spatial state and dynamics in workspace as the real slave robot and has synchronism with robot's movement.

## 5.2.1 Components of Phantom Framework

The phantom framework has several major components like virtual workspace, a phantom robot that works under master control in this space and the methods to affect incidence of the sensed effects as percept details a follow below.

### 5.2.1a Virtual work space

The robot's Cartesian workspace is digitised in voxel form. For the phantom to work, we form a virtual workspace comprising the voxelised version of the CAD models (STL form) in the robot's voxelised Cartesian workspace. The individual part models, with vicinity layers, are assembled into a 3D array structured workspace model $WVA$ representing the entire workspace using voxels. The OVAs (Chapter 4 Section 4.6.3) being also of same structure as the WVA, the assembly of WVA is fairly well defined sequential process but it needs the drag property preservation by certain pre-decided rule. The WVA, 3D matrix arranged, array data is a record (figure 5.3). Note that the record is accessed by the array index and the index data is not embedded in record. The figure is for easy understanding of scheme.

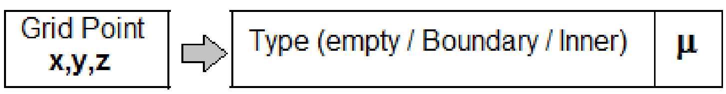



Figure 5.3: Voxel record in WVA.

### 5.2.1b The Phantom Robot

The purpose of the phantom robot is to impart the dynamics of the slave robot to the slave mounted virtual sensor in virtual workspace. Hence it needs a software emulation of the slave robot arm, which hosts the sensors at desired location on itself. This is done by forming kinematics equivalent model.Robot's kinematics [40] model represents it as a set of joints and links and forward kinematics computation provides its spatial state for a given set of joint positions. Generally, for modelling the robot mechanism, the links are represented as sticks of same length as joint to joint distances connected by the link (figure 5.4). The computations are fairly structured by use of DH representations [40]. For a point on the stick model the kinematics model has to consider the last link as the one formed by the point and the parent joint of the link that hosts the point.

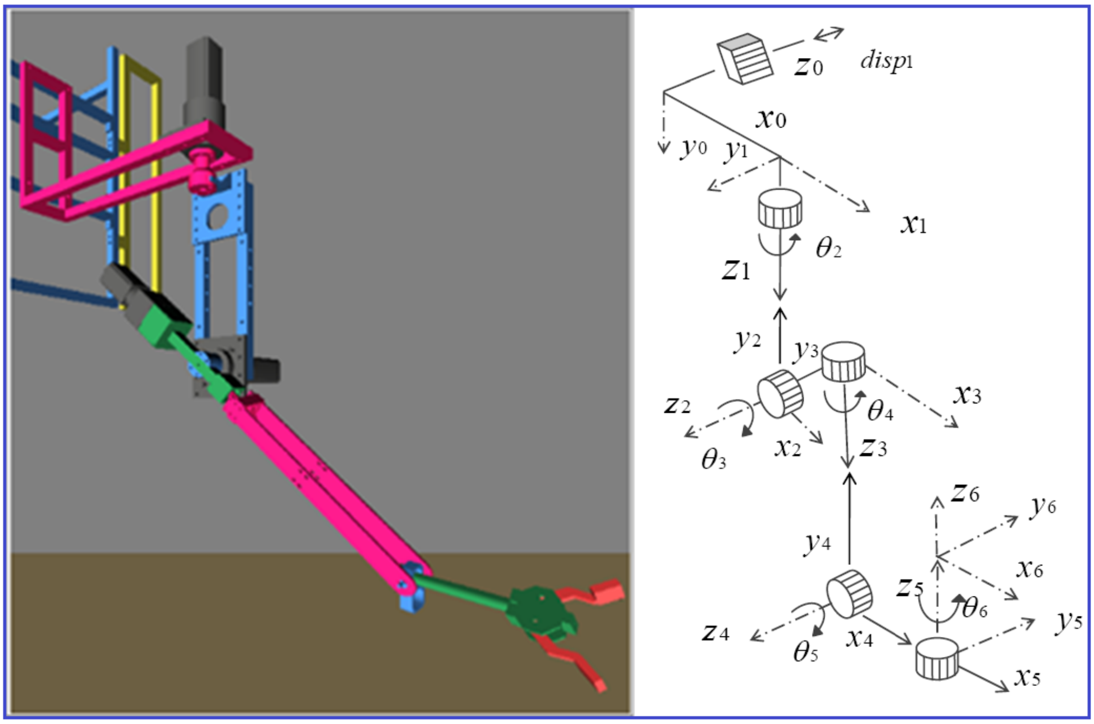



Figure 5.4: Experimental robot modeled in virtual workspace and its DH representation

### 5.2.1c Haptic Rendering of the percept

In principle, for developing kinesthetic feel, force on the operator finger or hand is the haptic clue. The force rendering joystick, described in chapter 2 can be employed for this purpose as it can also generate commands like a master arm in our experimental 1 DOF setup for a slave joint in virtual workspace. In this arrangement (figure 5.5) the handle receive operator force and causes torque on shaft which is given opposite torque by the force transducer motor acting on MPC generated percept parameter.

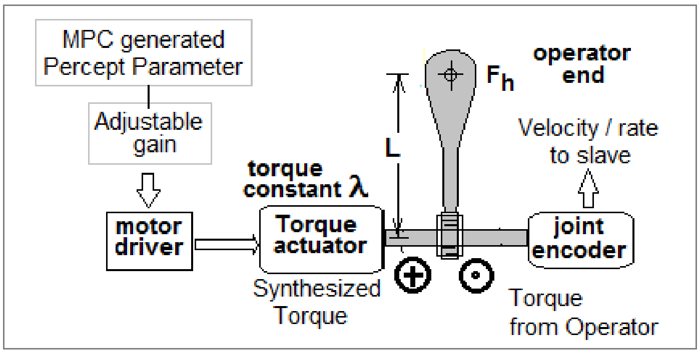



Figure 5.5: A force rendering cum command generation scheme for experiments.

## 5.2.2 Processes in Phantom Formation for Sensor Hosting on Slave Robot

The processes that enable sensor function in phantom include activation of neighboring space around an object and mapping the activated space in discretised virtual workspace (Cartesian) of the slave robot.

### 5.2.2a Workspace Activation

The robot's Cartesian workspace is digitised in voxel form. The virtual workspace WVA is formed by merging the OVA arrays containing the part and neighbour definition voxels (chapter 4). The OVAs are assembled into WVA which is a 3D array structured workspace model in the form of 3D array of voxels for the entire workspace. The OVAs being also of same structure as the WVA the assembly of WVA is fairly well defined sequential mapping of OVAs in WVAs. However the OVA data of earlier mapped part needs to be preserved by a rule. The WVA array data is a record (figure 5.6). Note that the record is accessed by the array index and the index data is not embedded in record. The figure is for easy understanding of scheme.

| Grid Point **x,y,z** | Object Index | Type (empty / Boundary / Inner) | Neighbor Order | **μ** |
|---|---|---|---|---|

Figure 5.6: OVA voxel record. The preservation of both neighbour order and μ value permits accurate merger of layered model in WVA by other criteria .

The Workspace Voxel Array (WVA) now represents the Cartesian Space of the robotic arm (slave). It is built by assembling the OVAs of all components in workspace.

Procedure for Assembly of $WVA(x, y, z)$ from OVA :-

$\boldsymbol{for}\ (i = 0\ to\ OVA\_xrange\ ; j = 0\ to\ OVA\_yrange\ ; k = 0\ to\ OVA\_zrange\ )$

$\boldsymbol{if}\quad \mu.WVA(OVA\_xoffset{+}i, OVA\_yoffset{+}j, OVA\_zoffset{+}k)\ < \mu.OVA(i,j,k)$
***then***
$\mu.WVA(OVA\_xoffset + i, OVA\_yoffset + j, OVA\_zoffset + k) = \mu.OVA(i, j, k)$

Here, $OVA\_xrange$, $OVA\_yrange$ and $OVA\_zrange$ are the offsets from $WVA$ origin for locating $OVA$ origin in $WVA$.

### 5.2.2b Composite Workspace Voxel Layer Formation and Test

Above procedure forms voxelised workspace model that has voxels coded as free ($\mu = 0$) or graded neighbor having varied $\mu$ values. Figure 5.7 show merger result.

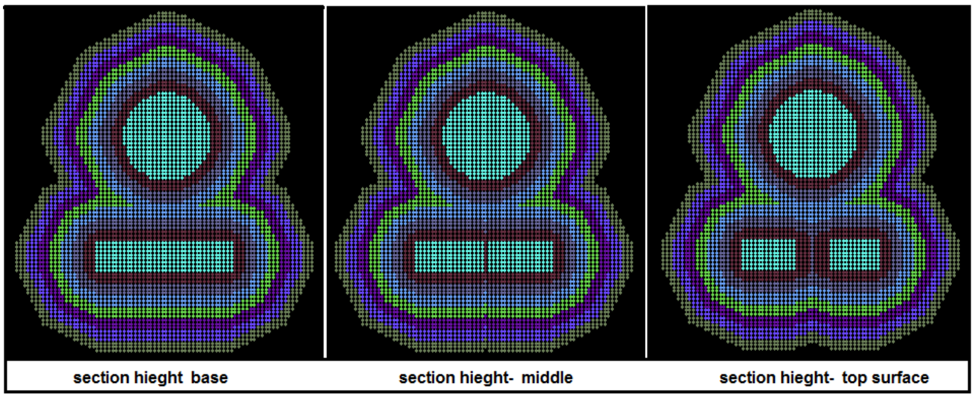



Figure 5.7: WVA formation with two objects. $XY$ plane slices at increasing height $Z$

A typical case of a workspace with two objects, a cylinder and a wedge block (figure 4.11a) formed using the respective objects' OVAs and merger law 'higher $\mu$ prevails' has been used in test and results is shown in figure 5.7. The three slices at bottom middle and near top levels of the parts are plotted here to show the correctness of the model formation. Towards the top, the split between the wedge legs widen and so neighbor voxels are formed within split zone too. The blue voxels are body voxels. The appearance of purple layer voxels in figure 5.7b and its widening with increasing height in the split zone establishes correctness of WVA assembly procedure.

### 5.2.2c Workspace Activation and Tests on Sensor Function in WVA

Two objects with large number of STL surfaces have been voxelized in individual OVAs with 6 $\mu$ layers. The WVA has been formed by 'higher drag-prevails' rule,

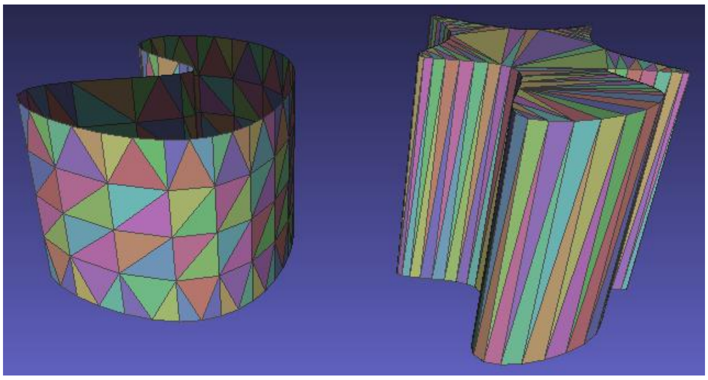


Figure 5.8: Closely placed Specimen parts with large STL surfaces

the results are shown in figure 5.8. Both objects layers are loaded with nonlinear $\mu$ profiles. Two test paths A-A1 and B-B1 were planned (figure 5.9) .

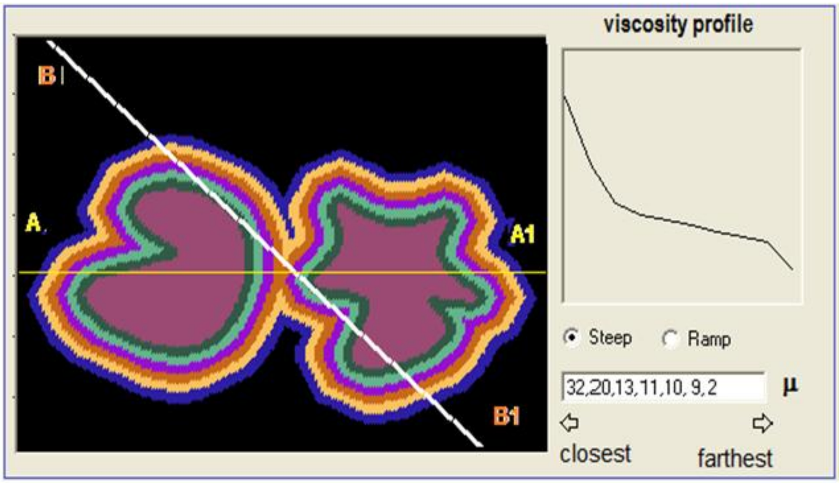
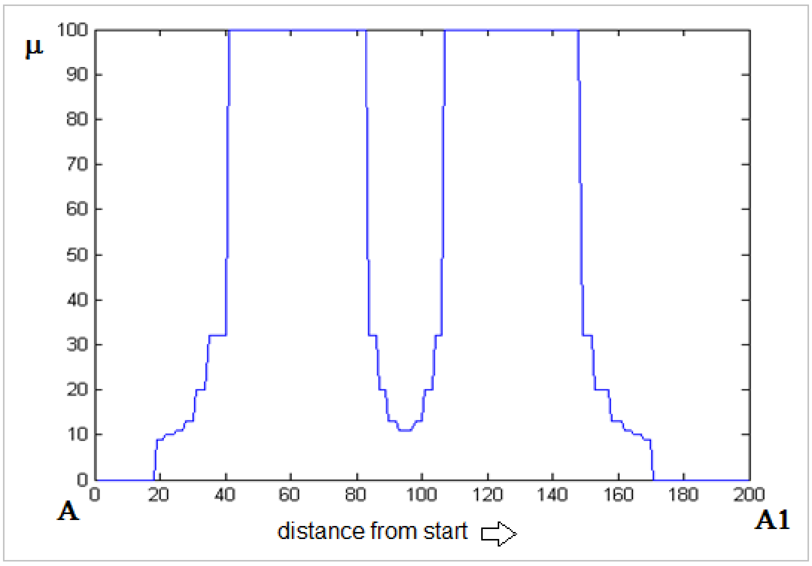
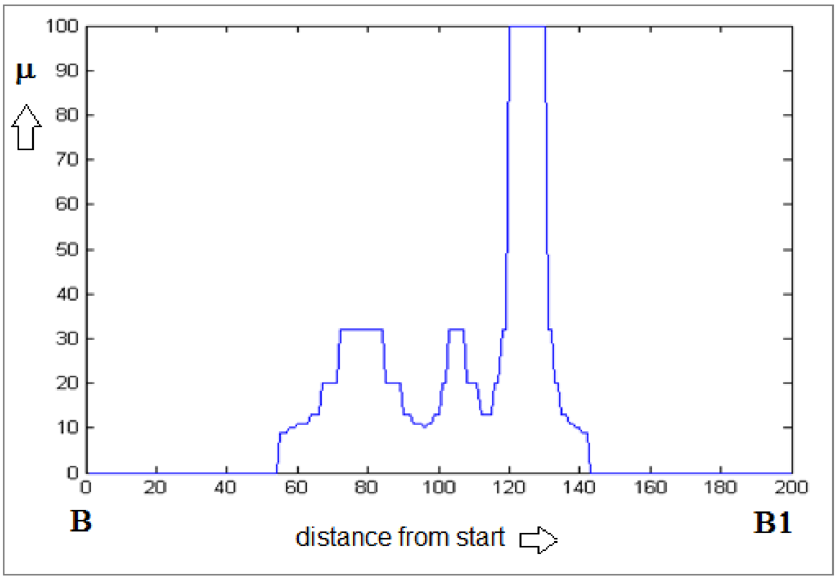



Figure 5.9: OVA merger results and test path vectors.

Moves developed basis drag profiles in figures 5.10. It is used as percept parameter $PP$ in further experiements. Note that one saddle is formed in curve for case with move vector AA1 (figure 5.9) but two saddles are formed in curve for case with move vector BB1. One is from merger and the other from object concavity.




Figures 5.10: Plot of 'percept parameter' for constant velocity on motion path AA1




Figures 5.11: Plot of 'percept parameter' produced by motion on path BB1 for constant velocity

## 5.2.3 A Functional Phantom

A typical phantom formed as described in previous section comprises of a robot arm of well defined kinematics structure [40] and the Cartesian workspace in which the robot body moves. The effect of virtual environment is reconstructed in real form by it. Here we form a typical phantom with an articulated arm and carry out tests on it as a specimen functioning module.

### 5.2.3a Phantom for an Articulated Arm

The phantom system was formed by WVA of 1cm$^3$ voxels and a simplified model of an articulated arm in the same Cartesian space as the WVA.

The articulated arm has 3 joints with $\Psi$, $\theta$, $\phi$ rotations. The link under test is 57 cm long. The bar is 5cmX5cm and 80 cm long and placed horizontal with axis normal to $\Psi$ axis at height such that the rotation of joint $\theta$ to 40 degree hits the bar. The WVA was activated by 6 layers of 4cm thickness around the bar and $\mu$ was nonlinear with

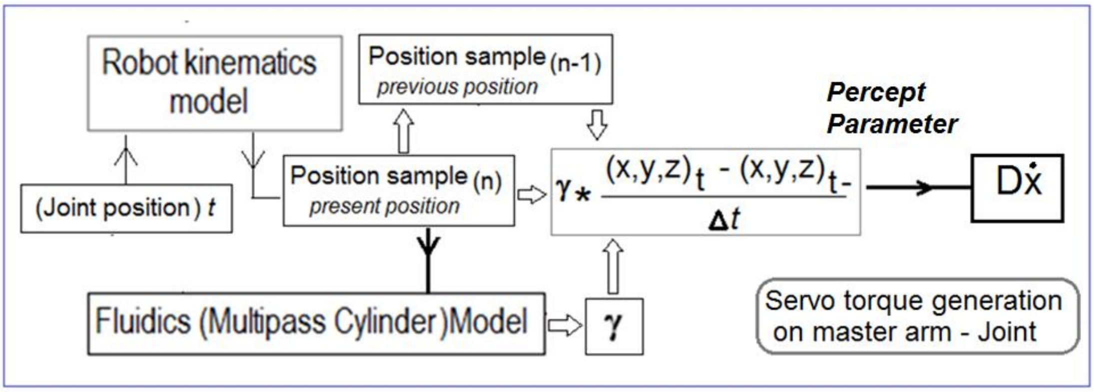



Figure 5.12: The percept parameter ‘PP’ estimation schema.

values 32, 20, 13, 11, 10, 8. The speed of rotation was 5 degrees per second. For generating easily analyzable result only $\theta$ joint was moved in range $0^o$ to $70^o$ for crossing over the bar model. The virtual phantom robot was simulated to follow a master joint and data was collected at 1 degree intervals. The phantom system was formed for standalone operation wherein the robot motion was created by preplanned joint data for causing the arm to go through the object for ‘haptics’ simulation of MPC embedded in phantom.

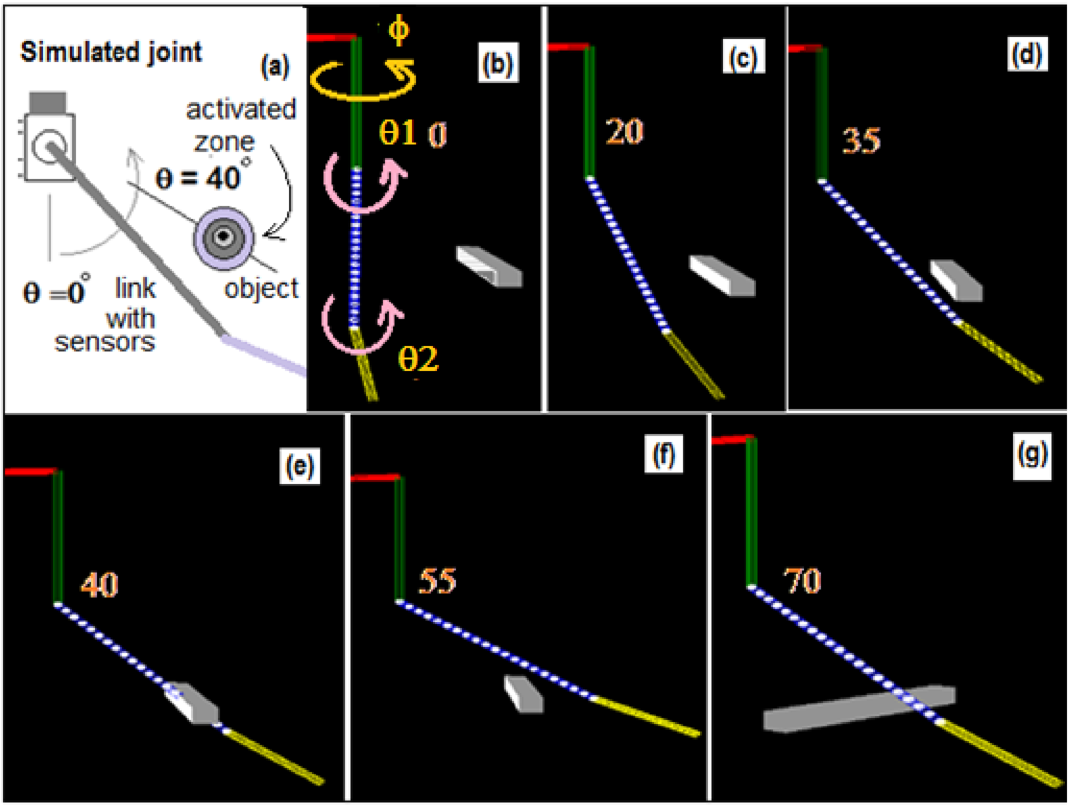



Figure 5.13: Experimental setup for testing phantom's performance on experimental phantom arm.

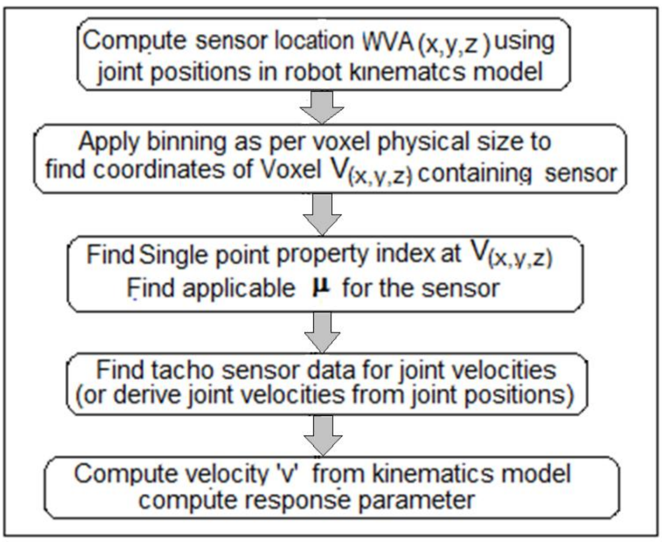



Figure 5.14: process of PP development

### 5.2.3b Functional Test of a Standalone Phantom

A test environment was created by integrating the Phantom processes with features for introducing sensors on the phantom robot body. '$K$' number of sensor locations at equal intervals on link axis are used. Several cases of sensor hosting on a link were formed. Sensor effect integration was implemented as integrated torque by link "$TLJ_n$" on joint $n$ ,

$$TLJ_n = \sum_{k=0}^{K} \mu_\epsilon . S_k * \left(\frac{k}{K} * LL\right) * \theta' * \left(\frac{k}{K} * LL\right) \tag{5.1}$$

Where, $\theta'$ is angular joint velocity,

$S_k$ Components form in Cartesian space are

$$S_k(x) = LJ_n(x) + \frac{k}{K} * \left(LJ_{n+1}(x) - LJ_n(x)\right) \tag{5.2}$$

$$S_k(y) = LJ_n(y) + \frac{k}{K} * \left(LJ_{n+1}(y) - LJ_n(y)\right) \tag{5.3}$$

$$S_k(z) = LJ_n(z) + \frac{k}{K} * \left(LJ_{n+1}(z) - LJ_n(z)\right) \tag{5.4}$$

Integer indices of sensor location in $WVA$ are found as follows.

$$X = \left[\frac{S_k(x)}{voxel\ size(x)}\right]$$

$$Y = \left[\frac{S_k(y)}{voxel\ size(y)}\right]$$

$$Z = \left[\frac{S_k(z)}{voxel\ size(z)}\right] \tag{5.5}$$

Where, $[.]$ is integer operator.

$$\mu_e . S_k = \mu . WVA(X, Y, Z) \tag{5.6}$$

Plugging above in equation (5.1) the total torque on joint $LJ_n$ forms $PP$.

$$PP = TLJ_n \tag{5.7}$$

**Case 1**: The link was populated with 10 sensor. Their sensed drag was converted to force feedback to the $\theta$ joint and total torque sensed was computed and logged. The synthesized torque works as 'Phantom Force Feedback' PFFB in telecontrol, here indicated as percept-parameter PP. 10 numbers of point sensors placed on test link at equal distance intervals from parent joint. Resultant torque is formed by a sum of all torques on joint.

$$T_q = \left(\sum_{k=1}^{10} \mu_\epsilon . S_k * \left(\frac{k}{10} * LL\right) * \theta' * \left(\frac{k}{10} * LL\right)\right)/10 \tag{5.8}$$

**Case 2**: 20 numbers of point sensors were placed on test link at equal distance intervals from parent joint. Resultant torque was determined as a sum of all torques on joint.

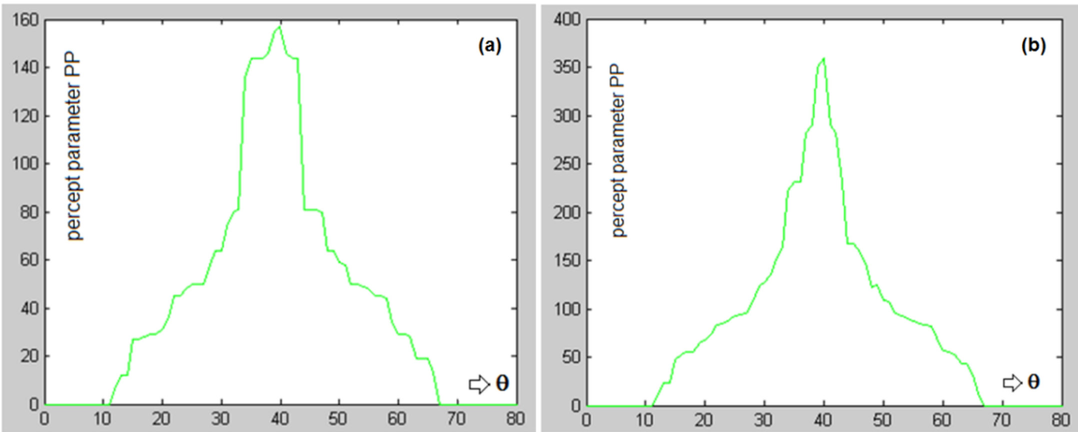



Figure 5.15: Result of Phantom function case 1: a; 10 sensors on arm, b; 20 sensors.

For this case

$$T_q = \left(\sum_{k=1}^{20} \mu_\epsilon . S_k * \left(\frac{k}{20} * LL\right) * \theta' * \left(\frac{k}{20} * LL\right)\right)/20 \tag{5.9}$$

**Case 3:** The joint feed-back torque was formed by summing the torques from the sensors and then normalizing it by the number of sensors inside the active zone. This number changes in different state of the phantom arm. The test runs were done for equally spaced 10, 20, 40, 60 and 90 sensors on the link

$$T_q = \left(\sum_{k=1}^{K} \mu_\epsilon . S_k * \left(\frac{k}{K} * LL\right) * \theta' * \left(\frac{k}{K} * LL\right)\right) / N_c \qquad (5.10)$$

Where $N_c$ is number of contributing sensors i.e. sensors for which $\mu$ is not zero.

$N_c$ is found by following procedure.

$N_c = 0$

$for\ k = 1\ to\ K$

$\quad if\left(\mu_\epsilon . S_k * \frac{k}{K} * LL > 0\right)\ then \qquad N_c = N_c + 1$

*Return* $N_c$.

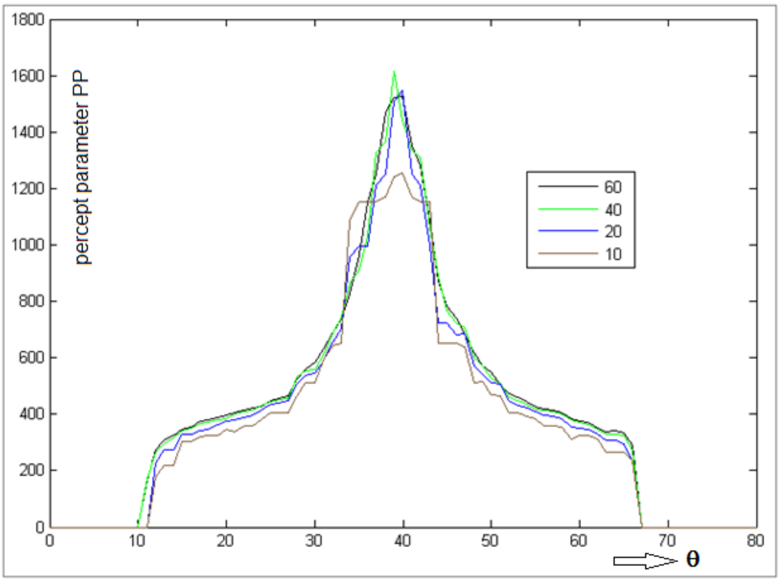



Figure 5.16: Result of Phantom function case 3a with varying sensor points shown in colours as per the legend in the figure.

The results in figure 5.16 show that as the number of sensors increased the synthesized torque by haptic simulation in phantom. Note that for the link dimension used in test sensor numbers up to 40 (green plot) imply sensor spacing larger than voxel size. On the other hand (figure 5.17) increasing the

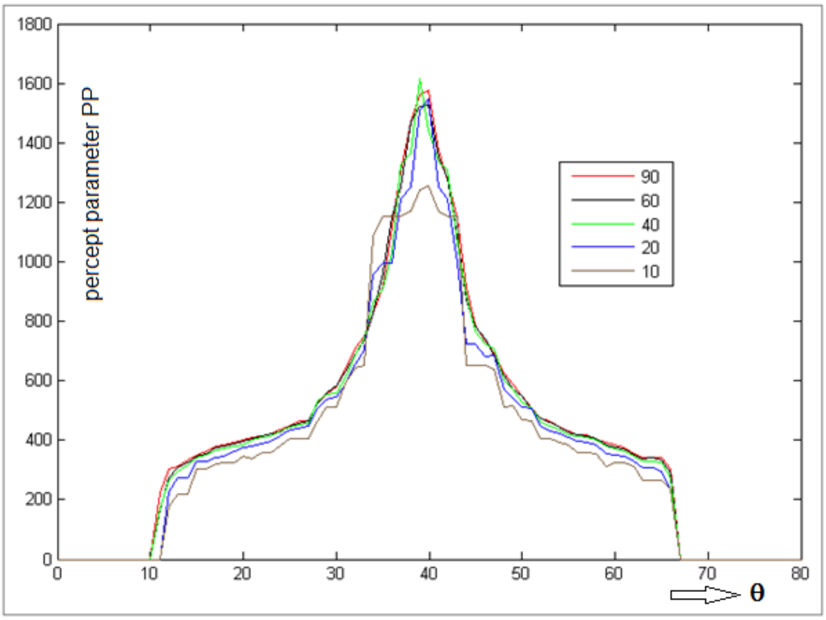



Figure 5.17: Result of Phantom function case 3b.

sensor number to 90 reduces inter sensor gap significantly below voxel size (70%) and as expected the improvement is not appreciable and case of 90 sensor (red plot) gives almost same performance as 60 sensors (black plot). This is significant as while performance does not increase in forming joint torque the real-time performance worsens as extra sensors consume computation time which is crucial in interactive telecontrol implementation.

# 5.3 A Perception Formation Approach Using Simplified MPC Model

The phantom system for the test setup is as per figure 5.13 and is formed by using the activated workspace detailed in section 5.2.2. The experimental articulated robot's kinematics is included and drag force sensing in activated space is integrated. Its function is as per the established behaviour. It functions between master side and slave side of telecontroller system. The approach essentially forms a slave robot model using its kinematics equivalent [40] and maintains it in same spatial state in phantom's environment as the slave state in remote work environment. The virtual slave hosts sensors on its body at desired locations of interaction sensing and thus the virtual sensor inherits same spatial state and dynamics in workspace as the real slave robot and has synchronism with robot's movement. figure 5.18 gives a percept

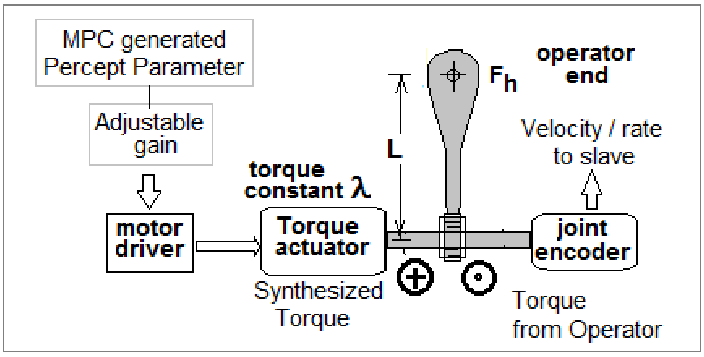



Figure 5.18: The percept rendering scheme

rendering scheme for physical effect on master arm.Transducer for phantom generated physical percept rendering (figure 5.19) has linear force development capability with selective sensitivity as shown in figure 5.20. Figure 5.21 shows robot arm used in experiments. Its kinematics model appears in figure 5.4. Joint $\theta_1$ shown in figure 5.13b was used for experiments.

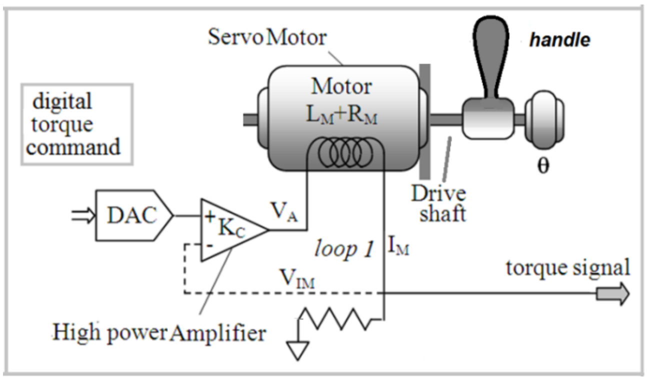



Figure 5.19: Torque transducer for physical rendering of the percept in force form. The torque signal is used by data acquisition system for tracking in experiments.

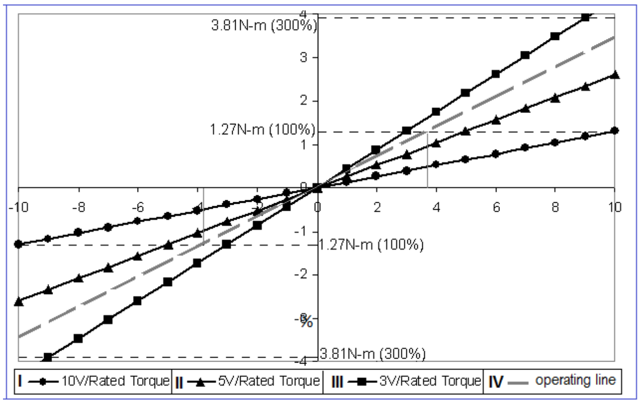



Figure 5.20: The servo motor with linear torque sensitivity. Sensitivity is selectable as depicted by the linear operating curves.

The hand force $F_h$ applied at a distance '$L$' from the shaft has to balance the torque formed, by PP through torque motor with torque sensitivity '$\lambda$'.

$$G.PP.\lambda = F_h * L \tag{5.11}$$

Hence $$F_h \propto PP \tag{5.12}$$

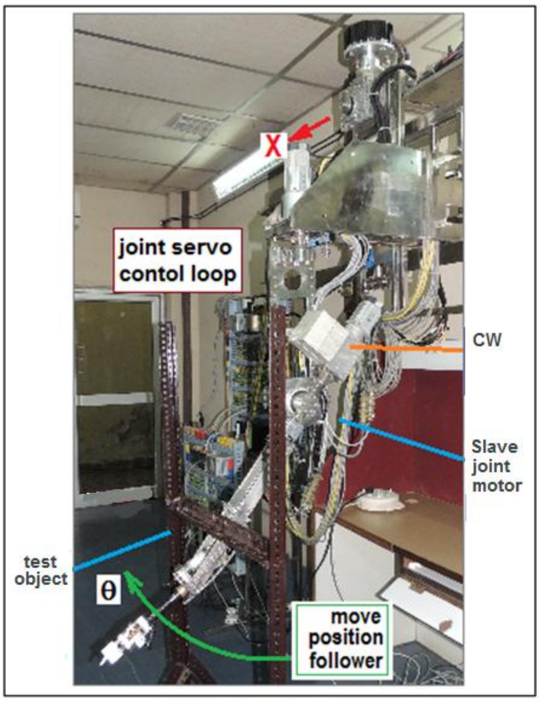



Figure 5.21: Experimental articulated arm. Note that for experiments in this phase test object was present only as model.

The percept parameter is converted to Phantom Generated Force Feedback PFFB that sensitizes operator to a closing-in towards a part

## 5.3.1 Experimental Validation of MPC Driven Vicinity Perception

For validating the MPC model of vicinity percept, experiments were carried out for manually operated master as well as controlled velocity master operation as described in sections below.

**5.3.1a Test Case 'a'**

In initial stage of tests the phantom system sensed the required position of the phantom arm by tracking slave position from encoder signals and maintained the phantom arm in same state as the slave. This was also tested on visual modeler's console (figure 5.13). On the Joystick the force was felt as per the nearness depending on speed. For varying rate of manual movement of joystick the force was felt on AJS handle. However constant speed of master joystick operation is not possible in the manual mode of operation.

**5.3.1b Test case 'b'**

In the previous test case 'a', the operator input is manual and master joint's velocity is uncontrolled. It is individual operator's behavior dependant and when faced with reaction torques, the manual input can fluctuate. Therefore the test setup is built to simulate master motion by a controlled servo loop.

One of our aims is to evaluate the phantom generated percept for a constant speed of approach of robot body to a workspace object. This is a case where the operator in spite of receiving a nearing feel continues his hand motion at same rate without change. The operator attempts to maintain same velocity under opposition from the environment. For this he/she has to apply varying force to the arm. Our target is to observe the varying torque demand in presence of Phantom Force Feedback ‘PFFB’ generated by the phantom. The servo-loop rotating the master joint maintains constant speed and the data tracking system records loop torque generated by control system to maintain a constant velocity against the varying opposition by PFFB. The loop torque is useful indicator of several parameters.

For carrying out constant speed operation the command encoder was turned by constant velocity servo loop. The servo shaft was connected to motor shaft that receives synthesized PFFB representing virtual drag from phantom as in figure 5.22. The master side joint operator has been formed by a test setup. Figure 5.23 shows control and data logging setup. No object is placed in workspace to avoid robot damage. The part exists only in virtual domain.

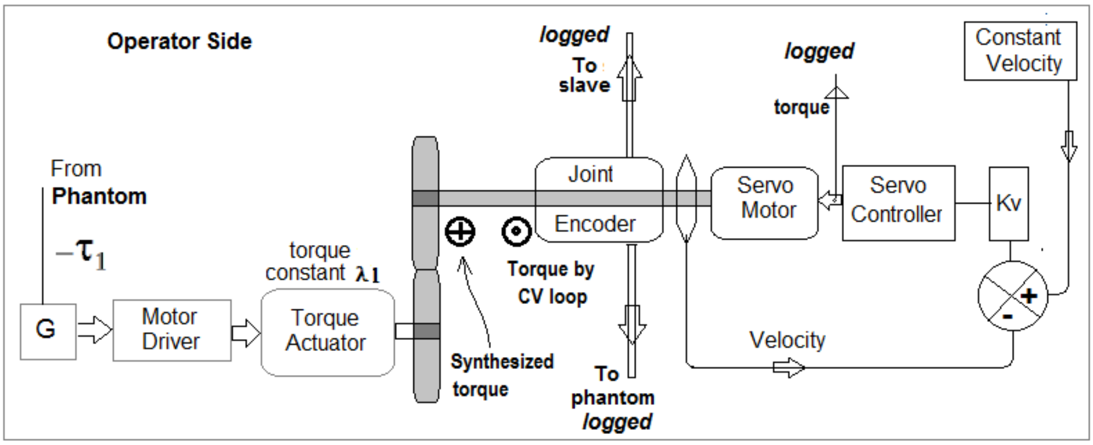



Figure 5.22: Modified test input scheme on master interface with motion emulation by servo controlled handle shaft.

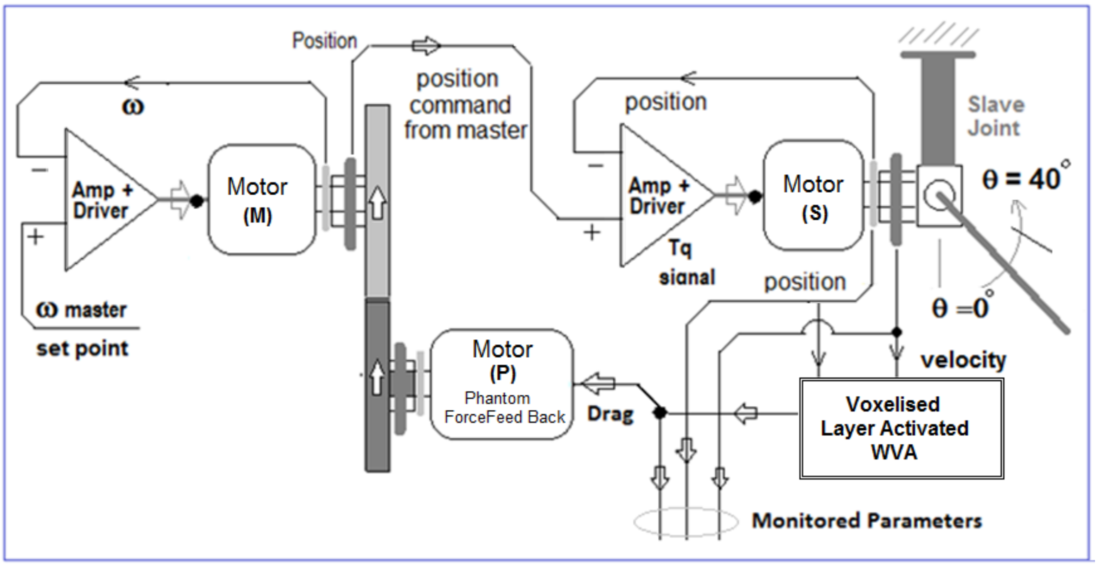



Figure 5.23: Control and data logging scheme for experiments

Following are noted about the test on a single robot joint:

i. The phantom generated virtual drag synthesizes a real force feedback by use of a dedicated torque actuator having linear torque characteristics coupled to a wheel. It has linear torque characteristics.

ii. Test setup is built to emulate master motion by a controlled speed servo loop acting on master joint. (figure 5.22 and 5.23). The well behaved speed command to slave from master generates constant speed and enables systematic tests on drag synthesis and observing its dependence on dynamics.

iii. The percept parameter produced by phantom using sensing of activated workspace is sent to percept renderer (figure 5.19).The torque signal produced by it is acquired as 14 bit data and is calibrated for torque on shaft before plotting as Phantom force feedback (PFFB).

iv. The torque is coupled to the operator handle shaft mechanically to enable its observation as pure phantom rendering of PFFB.

v. The slave position is monitored through encoders

vi. Our aim is to evaluate the phantom generated percept for a constant speed of approach of robot body to a workspace object. This is equivalent to a case where the operator in spite of receiving a nearing feel continues his hand motion without change. The operator attempts to maintain same velocity under opposition from the environment.

vii. Our target is to observe the varying torque generated by PFFB the phantom as PFFB in object vicinity while the object is approached at different values of constant speed. Since the real object in slave workspace may be hit, only its model exists in virtual space.

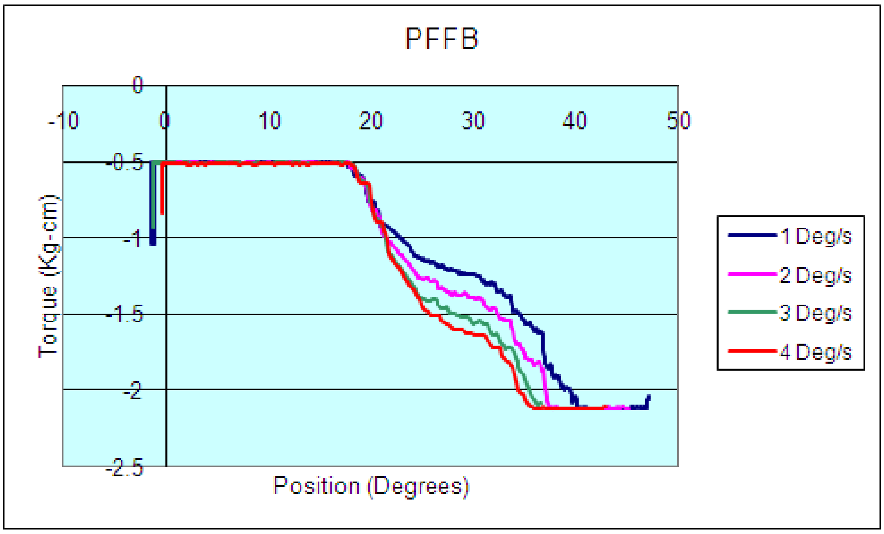



Figure 5.24: speed effect on PFFB rotation of arm is from 0 to 40 degrees, a saturation value of 2.2 kg-cm was imposed in the WVA model.

In our experiments the servo-loop maintains constant speed and the data tracking system records phantom generated force feedback PFFB. These are shown in figure 5.24.

### 5.3.1c Observations from Experimental Results

For clear understanding refer to figure 5.24 in previous section. In the experiments the PFFB generating torque was clamped at 2 Kg-cm for safe operation during experiments.

i. The Phantom worked in synchronism with the slave arm in virtual workspace.
ii. The PFFB generation succeeds in creating opposition to master motion near object and grows as per the $\mu$ profile programmed in the MPC model.
iii. The opposition is speed as well as nearness dependent. For higher speed on same locus of slave robot the phantom produces higher PFFB
iv. The on-set of PFFB depends on relative position from object and does not depend on speed of slave. However in case of higher speeds the rise is faster from on-set itself.

## 5.3.2 Phantom Function for Forming Drag Based Percept

From the experiments following are noted. The PFFB function is like the term ***B*** (Equation 5.13) in force balance equation of master and so can be represented as shown in figure 5.25 b and accounted as $Dx'$ in force balance equation (5.13)

$$Mx'' + Bx' + Dx' = F_h \tag{5.13}$$

This viscous drag based vicinity effect emulation has two outcomes

a. It causes opposition to master arm and so slows down the slave arm in vicinity of object body

b. The opposition to operator arm action forms a kinesthetic sense in the hands of the operator. Therefore we call it Haptic Assisted Vicinity Percept (HAVP).

The effort absorption by physical master arm depends on '*M*' and '*B*' which are very low for light and balanced master arm at moderately steady speed, hence, *Dx*' is dominant (by MPC design) and is reflected in torque produced by PFFB signal. Consequently the viscosity based drag is dominant effect and easily perceived by operator as drag. Figure 5.25a shows force absorbed at different speeds of move

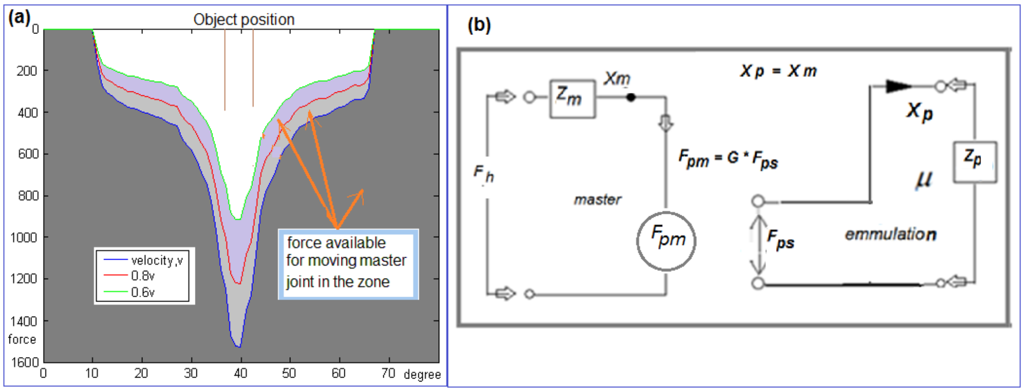



Figure 5.25: a; Force reaching the master joint after part absorption by the viscosity emulation (lighter zones), b; Equivalent function in BTM loop.

along joint axis near the object (figure 5.25a). Near the vertical line pair, the arm is closest to the object and absorption by PFFB is highest. The lower dark part shows part of operator effort available for master arm move after absorption by MPC model. Higher is the approach velocity higher is the absorption. The function is modeled in figure 5.25b.

# 5.4 Integrating MPC Generated Vicinity Perception with Classical BTM

The percept is virtually sensed but physically rendered on master driven by human operator. While in the slave side it is fairly straight forward to implement sensing as already described in previous sections , on master side it has to share the interface with the feedbacks that exist in the BTM.

## 5.4.1 Force Feedback in BTM

For the MPC model to coexist in the tele-control scheme a closer look at the working modalities of the conventional tele-controller is revisited. The traditional bilateral telecontrol uses some form of feedback from the slave robot arm's control loop with the objective of creating environment awareness to the master through force feedback. There are several methods [81,82,83, 3]. A review of bilateral telecontrol with the focus on feedback, reveals major 5 types of configurations (Table 5.1).

**Table 5.1:** Telecontrol Modalities

<table>
<tr><th></th><th>Master Actuator Force</th><th>Slave Actuator Force</th></tr>
<tr><td>(a)</td><td>$F_{am} = 0$</td><td rowspan="5">$F_{as} = K_{ps}(X_m - X_s) + K_{vs}\dot{X}_s$</td></tr>
<tr><td>(b)</td><td>$F_{am} = \frac{1}{A}F_{as}$</td></tr>
<tr><td>(c)</td><td>$F_{am} = \frac{1}{A}F_e$</td></tr>
<tr><td>(d)</td><td>$F_{am} = K_{pm}(X_m - X_s) + K_{vm}\dot{X}_m$</td></tr>
<tr><td>(e)</td><td>$F_{am} = K_{pm}(X_m - X_s) + K_{vm}\dot{X}_m + F_e$</td></tr>
</table>

**Note:** '$X$' represents displacement, $\dot{X}$ velocity, subscript '$m$' stands for master, '$s$' for slave, '$a$' for actuator, 'e' for environment , $K_p$, $K_v$ are position and velocity gains.

Actuator force relationships in Bilateral Telecontrol are summarized in this table. The cases in rows a, b and d are notable for their frugal use of sensor in the form of shaft encoder only. Table 5.1 lists Telemanipulation Controller Architectures in the two controller form where the master side servo loop and slave side servo loops are physically separate and are connected by electrical signals for cross couplings shown by 'm' subscript on right column for slave side whereas 's' and 'e' subscript on left column for master side. Cases are : (a) Position Forward 'PF', (b) Slave Coordinator Torque Feedback 'SCTFB', (c) Position Forward/Force Feedback 'PFFF', (d) Position Exchange 'PE' and (e) Position exchange / force feedback (PEFF).

### 5.4.2 Critical Aspects of Phantom Integration with BTM

For integration of phantom with bilateral telecontrol system that have feedback from slave side (cases b to e in Table 5.1) the Phantom has to coexist with the additional feedback and therefore the physical rendering of force needs to address the homogeneity aspect. A form of force feedback that is a function of the coordinating torque in the position follower slave loop, has been identified as a method in literature [81] The slave coordinating torque based feedback hereafter called Slave Coordination Torque Feedback (SCTFB) has some attraction for us as this mode can work with load torque monitoring of slave servo loop alone without any additional force torque sensors. The Phantom's feedback needs integration with the bilateral control and has to exist with a system having additional channel of feedback reaching master from slave. We generalize these as BTM force feedback and refer to it in general as Slave Force Feedback 'SFFB' (figure 5.26). The SCTFB is a type of SFFB in specific implementations. The phantom effectiveness for BTM modalities indicated in the table 5.1 are treated in following sections.

### 5.4.3 Phantom Application for Position Forward Control

This BTM model appears as case 'a' in row 1 Table 5.1. The configuration does not produce any force feedback from slave side in the classical form. However the Phantom –I (figure 5.2) forms a drag as per equation 5.13. For following the form used in table 5.1, it is written as

$$F_h = F_{am} + X_m Z_m \tag{5.14}$$

This case is same as that treated in section 5.3. Foregoing experiments, in the section show, that the phantom generated 'percept' can work well to enhance feel of presence of a workspace object by drag generation in master interface. The Haptic Assisted Vicinity Percept 'HAVP' is the mainstay in conveying object presence through haptic route in this form of telemanipulation. The feel of object presence can be provided by HAVP. The Closed Circuit Television (CCTV) based tele-operation also becomes more dependable as for the parts of robot body that are occluded or out of 'view cone' the system can generate feedback through this type of phantom. However the master side, which otherwise is passive and tracks only the joint angles resulting from human operator's input on master arm, needs to be active with torque transduction servo hardware.

### 5.4.4 Improved Phantom II for Co-Working with Force Feedback BTM

Phantom application for cases $(b)$ and $(c)$

The previous section has demonstrated that the MPC model is capable of rendering a real force based on a virtual drag. Hence representing this drag by $D\dot{x}$

$$F_h = F_{am} + X_m Z_m + D\dot{X}_m \tag{5.15}$$

$$F_h = F_{am} + M_m \ddot{X}_m + B_m \dot{X}_m + D\dot{X}_m \tag{5.16}$$

Where $M_m$ and $B_m$ are the inertia and damping properties of master arm.

For light master arm moving at normal speed of operation, the contribution of $M$ term may be negligible or may be a source of an otherwise well felt inertial effect under only very high acceleration condition. In well designed master, with low friction at joints, the term $B$ is negligible. For case 'a' in Table 5.1, where no slave feedback is present

$$F_h = D\dot{X}_m \tag{5.17}$$

And for case 'b'

$$F_h = \frac{1}{A} F_{as} + D\dot{X}_m \tag{5.18}$$

For case 'c'

$$F_h = \frac{1}{A} F_e + D\dot{X}_m \tag{5.19}$$

A superior phantom framework Phantom II (figure 5.26) is formed to integrate the BTM force feedback and the phantom synthesized feedback. This framework uses the workspace model WVA with voxel attributes and is equipped use several types of force synthesis for supporting the experiments undertaken in subsequent sections.

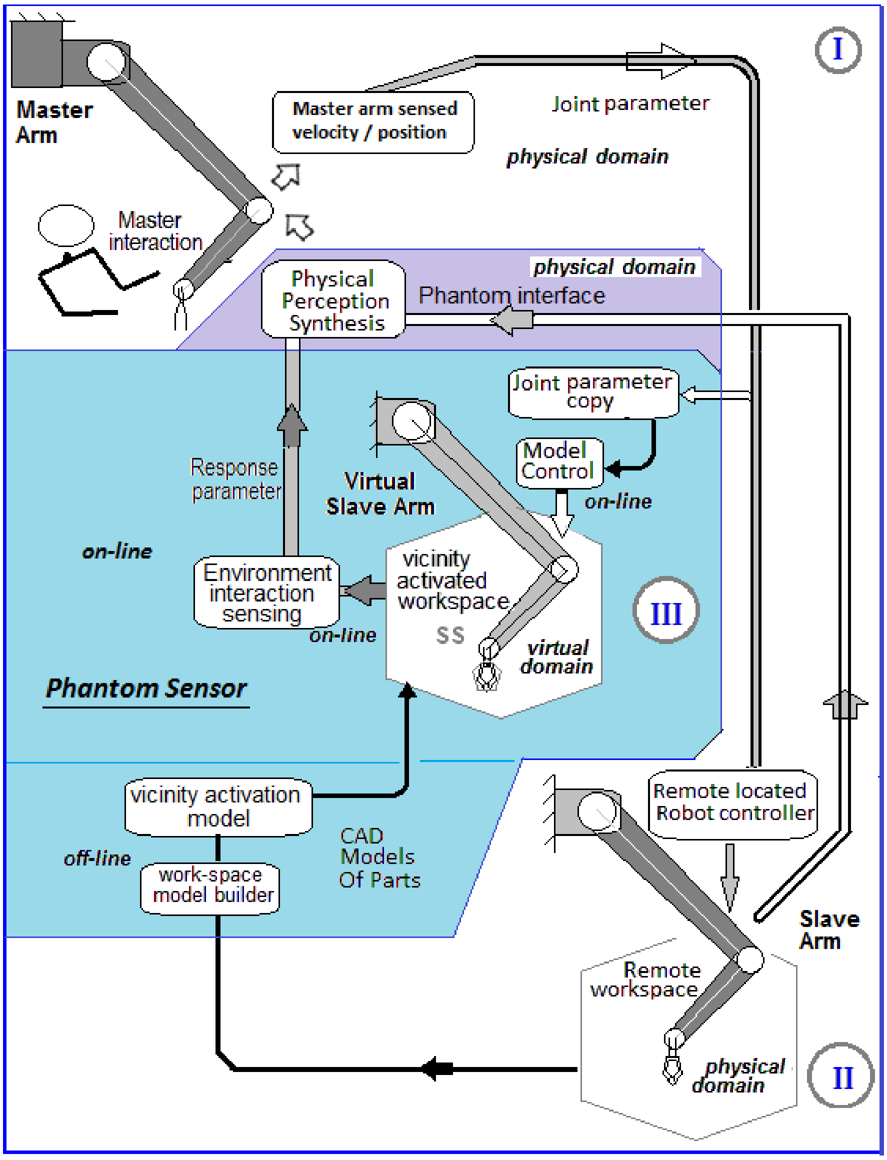



Figure 5.26: Phantom version II implements, functioning with force feedback tele-control systems

### 5.4.4a Percept Integration in Tele-Control with Position Forward and Slave Loop Coordinating Torque Feedback

The master side joint operator has been formed by a test setup where following are achieved for test on a single robot joint:

a. The slaves force feedback (SFFB) is formed by giving input to a servo motor on master joint working in torque mode to deliver opposition force to the operator.

b. The phantom generated virtual drag synthesises and its conversion to real force feedback is done as earlier.
c. The feedback from slave has different modalities in the cases listed in table 1. Therefore the PFFB needs to be maintained in its native form till the largest extent before presentation to human operator.

Note that in experimental setup, the two feedbacks have been merged in mechanical mode. It has the advantage of maintaining the phantom generated proximity effects in exclusive form separate from those developed by the slave feedback, to the largest extent and thus detailed measurement on these can be performed. The experiments are designed for analyzing the simultaneous action of force feedbacks from two sources PFFB like the earlier case and ‘slave force feedback’ SFFB from the slave.

### 5.4.4b Difficulties in Analyzing the Performance of Percept Integration

a. The operator inputs are manual and master joint’s velocity is uncontrolled. For solving this as in earlier experiments (section-5.3), here too a constant velocity servo-loop maintains motion against the varying opposition by PFFB and loop torque is used as the indicator of force effect rendered on human arm.
b. Testing performance of PFFB very near object at high speed is impossible for a non destructive test scenario being built here, as owing to inertia effects, the arm cannot be stopped immediately and hit with specimen object shall cause damage to robot. Therefore the joint under test in our case, is decelerated very near the object to avoid damage to the robot and just after contact the arm is stopped by actuating electromagnetic break.

Hence experiments do not cause full velocity hits to generate data.

### 5.4.5 Experimental Setup for Phantom II Co-Working With BTM Using MPC

An experimental setup has been built to implement and test co-working of the MPC model based phantom with conventional bilateral tele-control of the types indicated in Table 5.1.

The experiments are designed for two major classes of operating conditions

i. Balanced slave arm
ii. Unbalanced slave arm
iii. Both types are subjected to different velocities of working.

In this set up the slave controllers' loop torque signal is monitored and also used for force feedback to the master interface. The master interface is equipped with arrangement for coupling the phantom rendered PFFB (torque) and the SFFB (torque) in mechanical form, in master servo loop.

For a bilateral system with force feedback from slave as per case $b$ and $c$, relations 5.15 and 5.18 are applicable.

This needs a setup (figure 5.27) for its implementation $\tau_1$and is generated by phantom. For convenience we use G as a gain factor amplifying a basis drag which is the Percept parameter $PP$, and $(1/A\,)$ is gain applicable to SFFB depending on '$F_e$'.

Force balance equations on master side is

$$(\tau_1 * \frac{G}{\lambda_1} + \tau_2 * \frac{1}{A*\lambda_2})/L \;-\; M\ddot{x} - B\dot{x} = F_h \qquad (5.20)$$

Here $\tau_2$ is generated by SFFB.

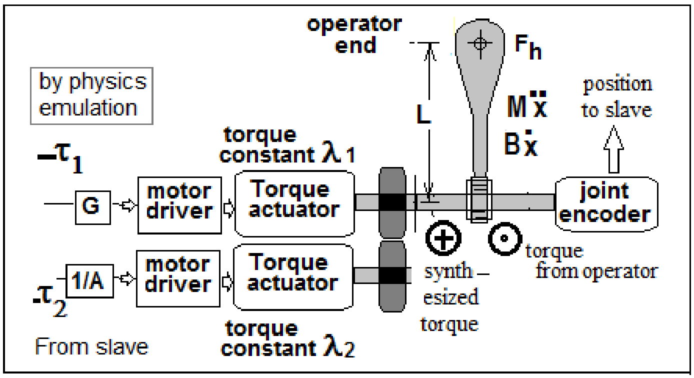



Figure 5.27: Schematic for mechanical coupling of the PFFB and SFFB. The two oppose the operator effort in rotating the master joint shaft coupled to encoder.

All the cases (a) to (e) use ‘position following’ by the slave. The joint control set-up for the test slave arm is shown in figure5.28. It is suitable for encoder based position control. The loop torque is available for logging the feed back in SCTFB modality.

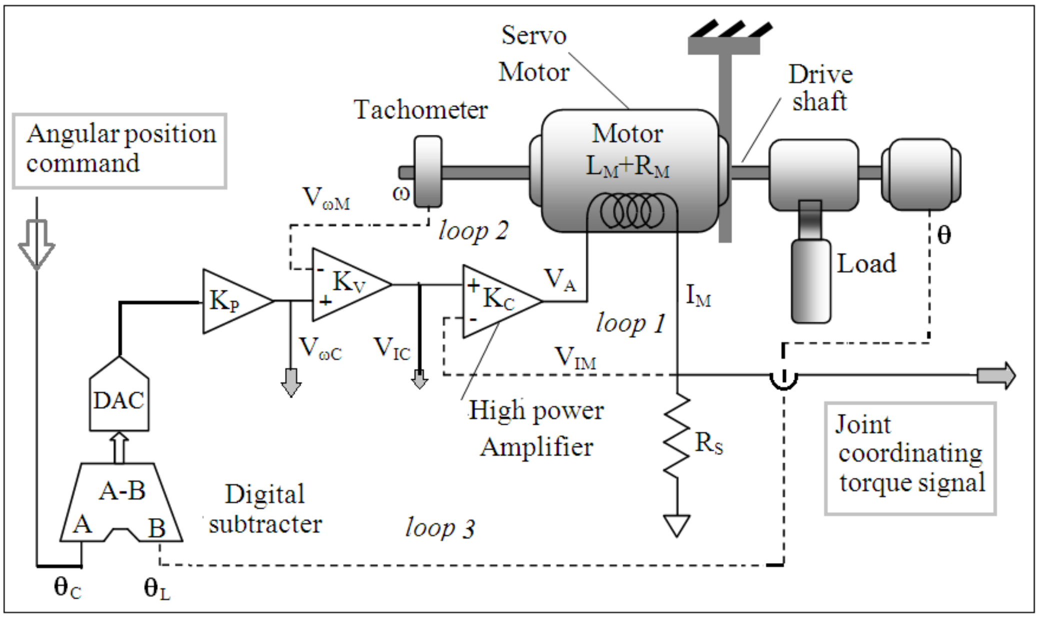



Figure 5.28: A position follower control for slave joint operation. It provides load torque signal for external use which is referred as SCTFB in telecontrol feedback methods.

The slave control is closed loop position follower (figure 5.28) wherein the commands are angular joint position arriving from master. In the test scheme (figure 5.29) wheel 2 and wheel 3 always turn opposite to wheel 1 in torque mode to form FFB. The configuration creates force feel to master by PFFB using phantom FFB motor and SCTFB by slave FFB motor on integrated master interface.

A test setup has been built with force coupling between various elements as detailed in figure 5.29. The obstacle object used to activate the phantom workspace was physically placed in the workspace at a position and pose matching the model in WVA. By quick movement of handle to move slave in the vicinity, towards object an opposition drag was felt by human operator. On the opposite direction the opposition faded. Owing to fast reduction of $\mu$ values the feeling is not sticky. In sticky behavior, the object tends to draw a departing robot back towards it.

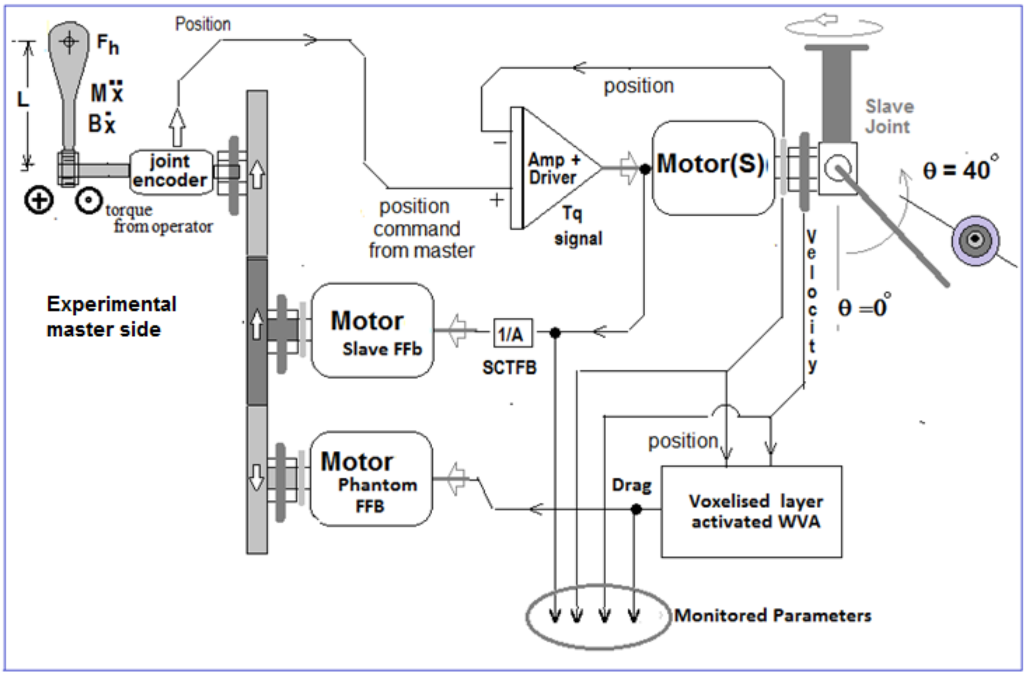



Figure 5.29: Experimental percept rendering and integration with conventional telecontol in position forward and force feedback control scheme

## 5.4.6 Results on Phantom II and BTM Co-working

A slow movement towards the object permitted the slave arm to reach and touch slowly. On contact with object body while the PFFB remained negligible owing to near zero velocity of arm, the SCTFB increased as the hindered arm caused position error to build and the closed loop control increased the joint motor torque.

**Phantom performance for constant velocity master and position follower slave with force feedback type b:**

Like the earlier case a constant velocity movement of the master joint is used in this case also. The master driver – serves for overcoming the two loads from the PFFB and SCTFB. The test setup is as in figure 5.30. By using the control scheme shown in figure 5.31 a constant speed input with torque tracking capability is built.

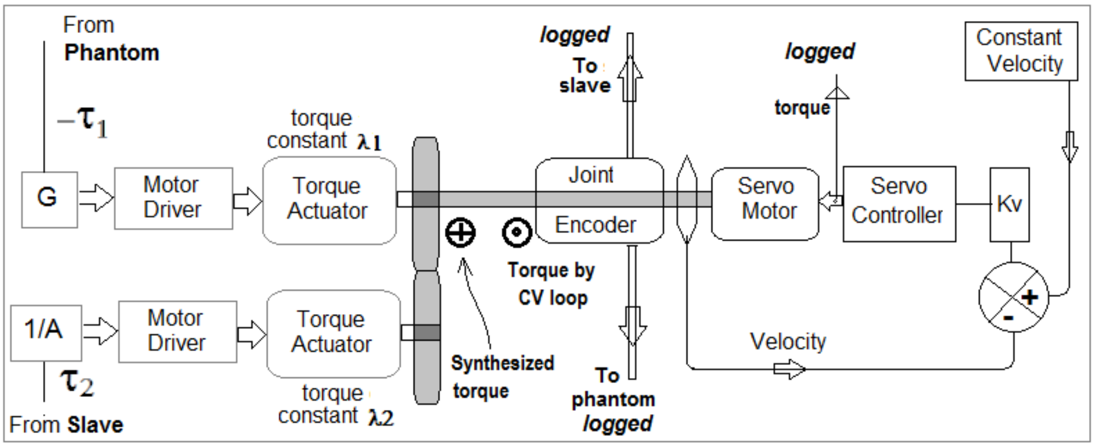



Figure 5.30: Constant velocity master operation against SCTFB and PFFB

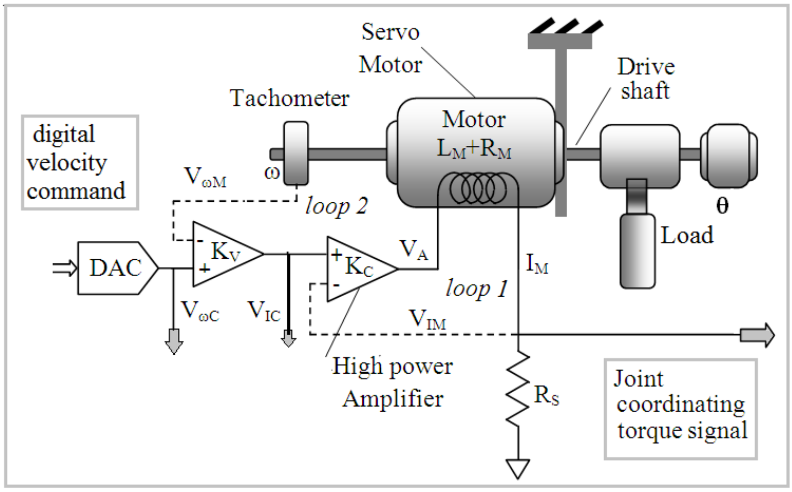



Figure 5.31: Constant speed master joint operation is emulated by velocity servo-loop. The loop torque signal is sent out for load torque estimation by external system.

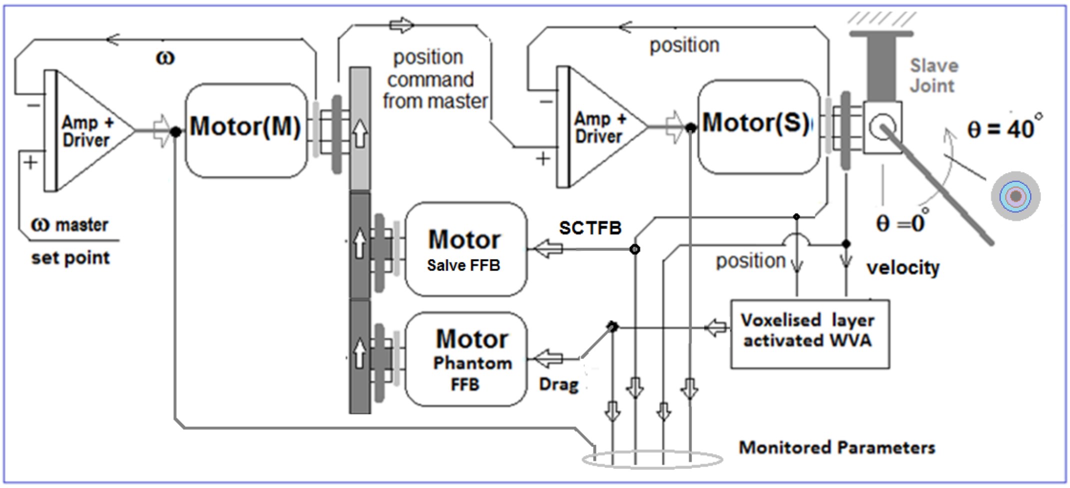



Figure 5.32: Modified Experimental setup with master motor (M) operating in constant velocity mode in a SCTFB type (case ‘b’) tele-control scheme –case ‘b’ and shared percept rendering.

**Master servo loop with constant velocity drive**

In experiments, ‘$F_h$’ application to master arm is emulated by a constant velocity actuator, as human operator cannot precisely control velocity. Also while maintaining position, hand tremors cause velocity change.

Human sensing and control bandwidths decide the rate at which the sampling loop should run. Hogan discovered that the human neuromuscular system exhibits externally simple, spring-like behaviour [84]. Note that human limbs are not passive in all conditions, but the bandwidth at which a subject can perform active motions is very low compared to the frequencies at which stability problems may arise. Some authors [85,86] report that the bandwidth at which humans can perform controlled actions with the hand or fingers is between 5 and 10Hz. On the other hand, sensing bandwidth can be as high as 20 to 30Hz for proprioception, 400Hz for tactile sensing, and 5 to 10 kHz for roughness perception. Experiments in this work have been run at 50 ms update rate.

The experiment was carried out in speed control mode. Master wheel was moved to destination position such that arm hits the obstacle. During this experiment, software generated feedback emulating the MPC model, hereafter referred as Phantom Force Feedback 'PFFB', was applied to Phantom FFB motor and servo-feedback from slave arm, referred hereafter as a Slave Control Loop Torque Feed-back 'SCTFB' was applied to slave FFB motor. Counter weight was applied to make the slave arm balanced. Two sets of experiments were carried out. In first set, slave arm was actuated at a speed of 6.63 mm/sec and in second set at 13.23 mm/sec.

*Naming and legend convention for experimental plots:*

MLT- 'Master Loop Torque' for constant velocity operation by motor (M) (Blue)

PFFB- Phantom created Force Feedback' by torque motor of phantom (green)

SCTFB- Slave control torque feedback(red)

***Experimental results Set** I : Slave arm actuated at a speed of 6.63 mm/s (approx)-*

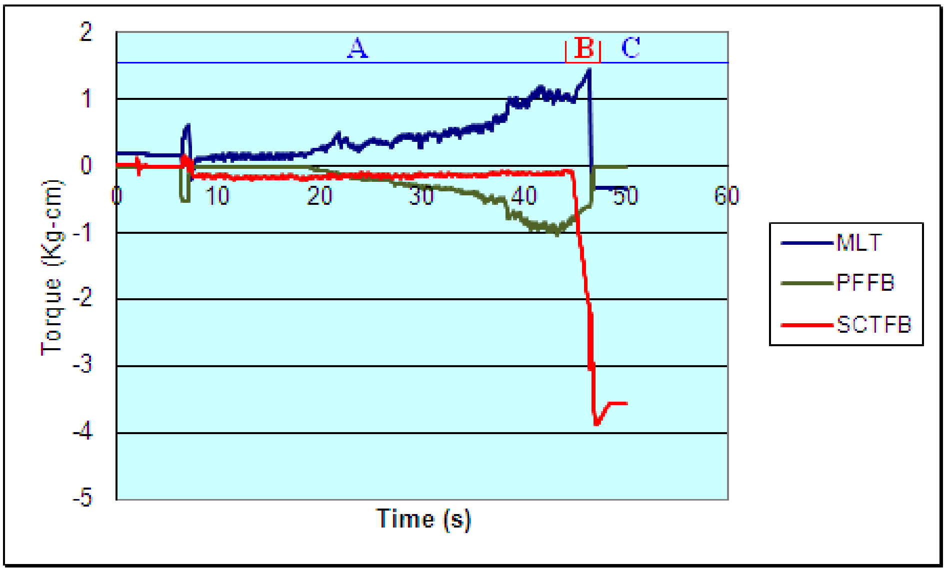



Figure 5.33: Master side MLT torque, PFFB and SCTFB at 6.63 mm/s speed.

The MLT, PFFB and SCTFB plots with respect to time (figure 5.33) for a operating speed 6.63 mm/s (approx). The SCTFB rise in zone B is due to the physical slave arm coming in contact with an object.

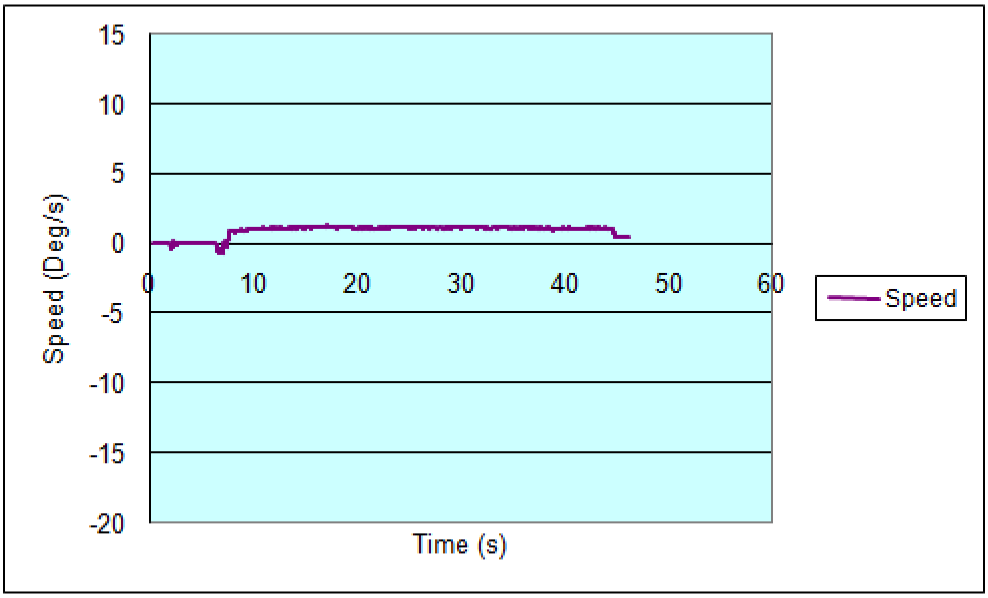



Figure 5.34: Variation of slave speed (Deg/s) with respect to time.

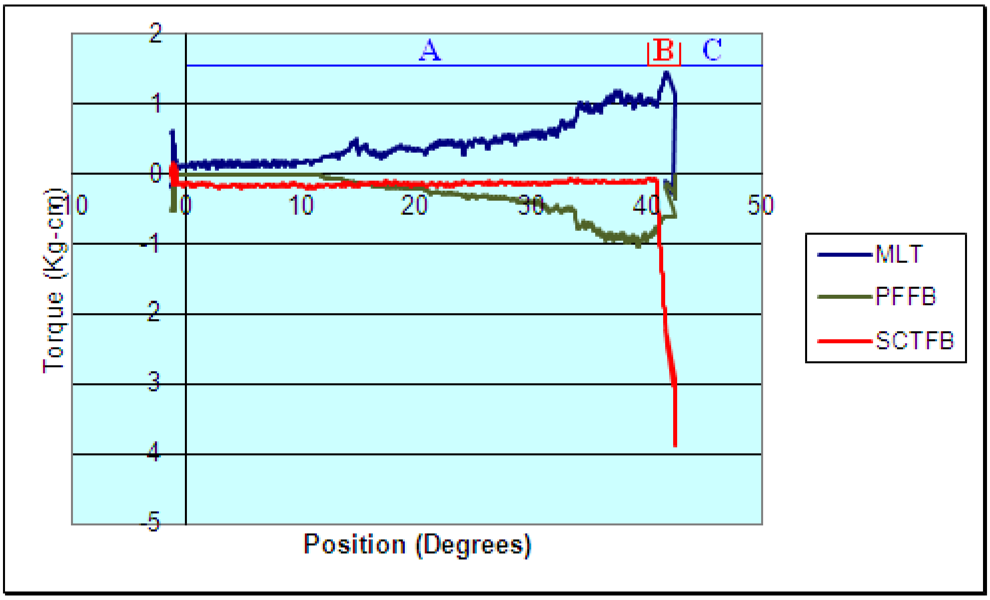



Figure 5.35: Behavior of master side MLT torque, PFFB and SCTFB at 6.63 mm/sec.

The MLT, PFFB and SCTFB plots with respect to arm position for a speed of operation 6.63 mm/s (approx) are as shown in figure 5.35. The SCTFB rise in zone B is due to the physical slave arm coming in contact with an object.

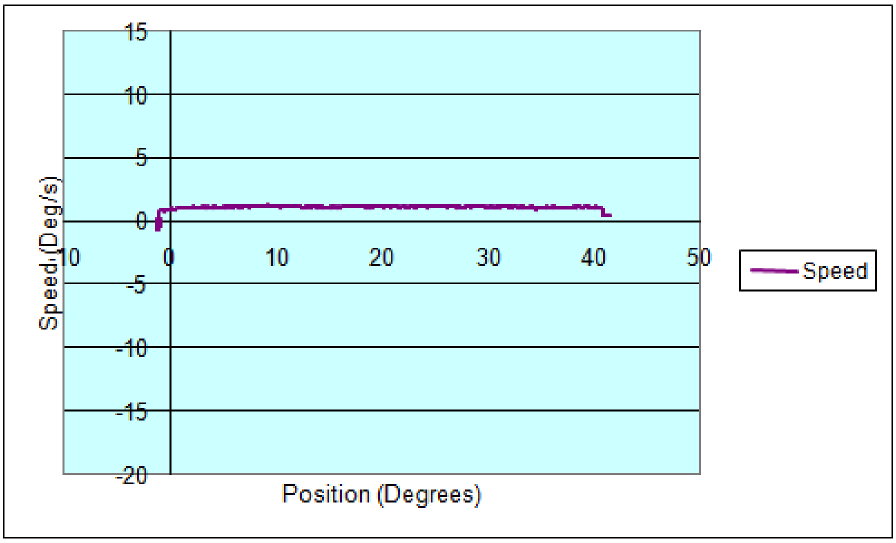



Figure 5.36: Slave speed variation with respect to position for nominal speed.

***Experimental results Set II** : Slave arm actuated at a speed of* 13.26 *mm/s (approx) -*

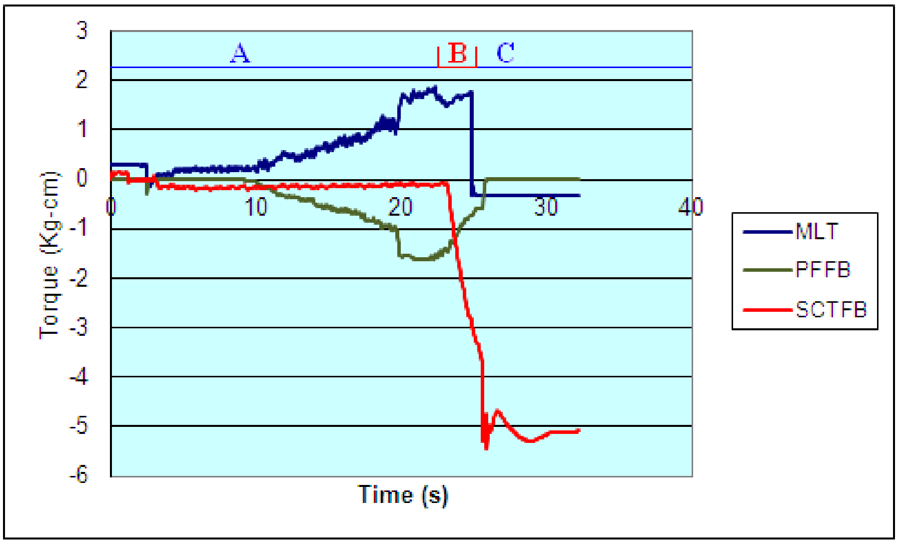



Figure 5.37 Behavior of master side MLT , PFFB and SCTFB at 13.26 mm/s (approx).

The MLT, PFFB and SCTFB behavior with respect to arm position for a speed of operation 13.26 mm/s (approx) is as shown in figure 5.37. The SCTFB rise on right is caused by contact with object.

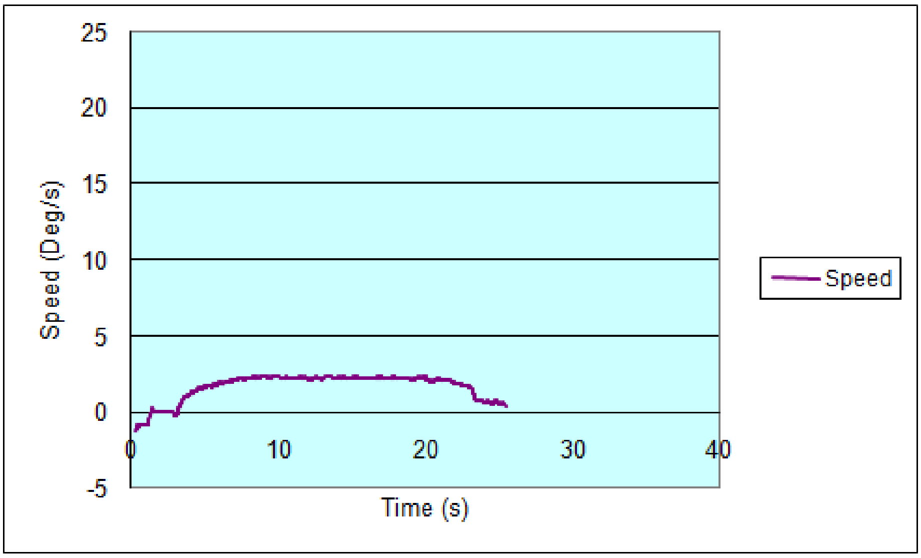



Figure 5.38: Variation of slave speed (Degree/s) with respect to time.

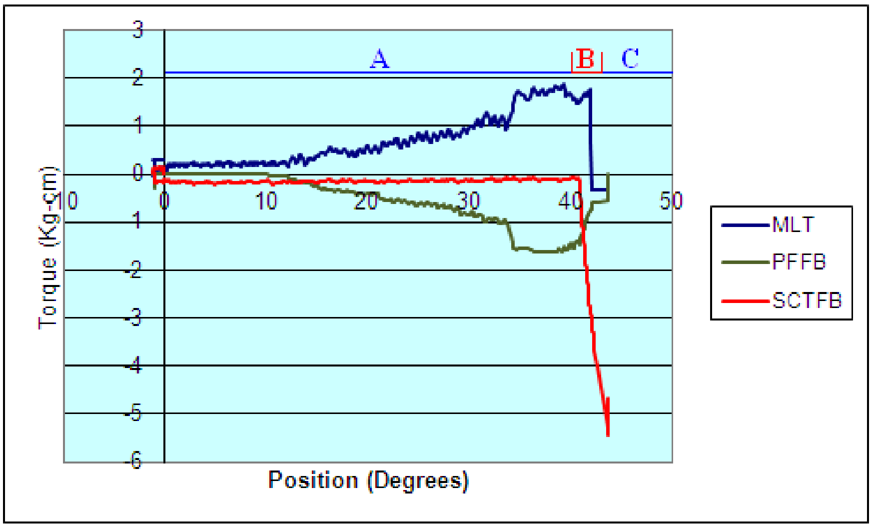



Figure 5.39: Master side MLT, PFFB and SCTFB behavior at 13.26 mm/sec

The MLT, PFFB and SCTFB behavior with respect to arm position for a speed of operation 13.26 mm/s (approx) is as shown in figure 5.39. The SCTFB rise in zone B is due to the physical slave arm coming in contact with an object. Plot to the left of vertical axis corresponds to initial start of servo loop before stabilization. At the start servo loop takes time to stabilize, hence initial few seconds data is not useful.

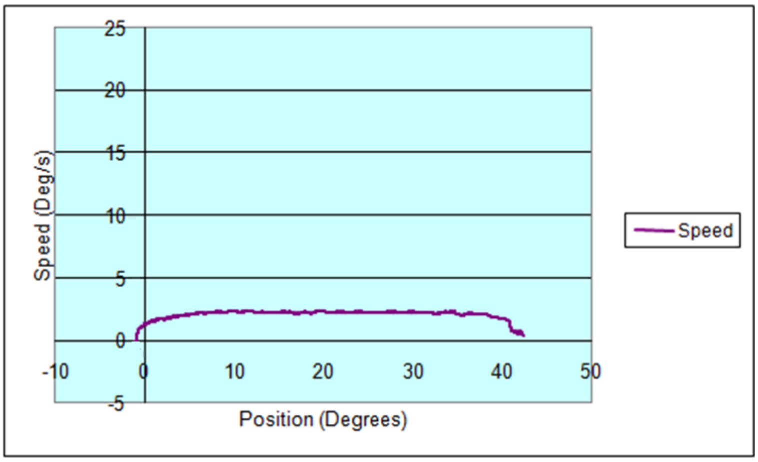



Figure 5.40 Variation of slave speed (Deg/s) with respect to position. Deceleration at 36 degrees set by velocity profile

The arm motion was stopped within a few seconds of contact to protect the robotic arm (slave). The velocity control profile imposed by master had been set to start deceleration about 4 degrees prior to contact position (40 degrees) for safe stopping. Consequently the PFFB starts falling in spite of reaching closer to part. The zone 'A' represents non contact phase wherein MLT varied predominantly owing to PFFB. The zone 'B' is 'just contact' zone where MLT starts initially falling owing to deceleration induced PFFB reduction and then increase owing to SCTFB increase caused by contact. Finally it is followed by fall due to servo shutdown.

***Unbalanced Slave case:***

A set of experiments were carried out on same experimental set up as used above after altering the slave arms counterweight to make it unbalanced. The master arm was again actuated by a constant speed servo loop at a velocity of 6.63 mm/s (approx) and 13.22 mm/sec (approx). Results are recorded fin figures 5.41 and 5.42 .

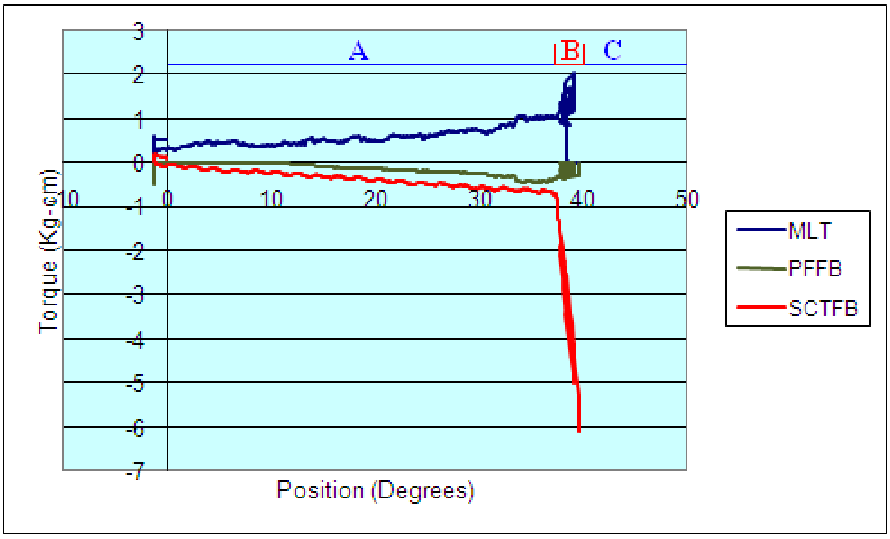



Figure 5.41: Behavior of master side MLT torque, PFFB torque and SCTFB for arm position at speed of operation 6.63 mm/s (approx) for unbalanced arm.

The MLT, PFFB and SCTFB behavior with respect to arm position for a speed of operation 6.63 mm/s (approx) is as shown in figure 5.41.The SCTFB rise in zone B is due to the physical slave arm coming in contact with an object.

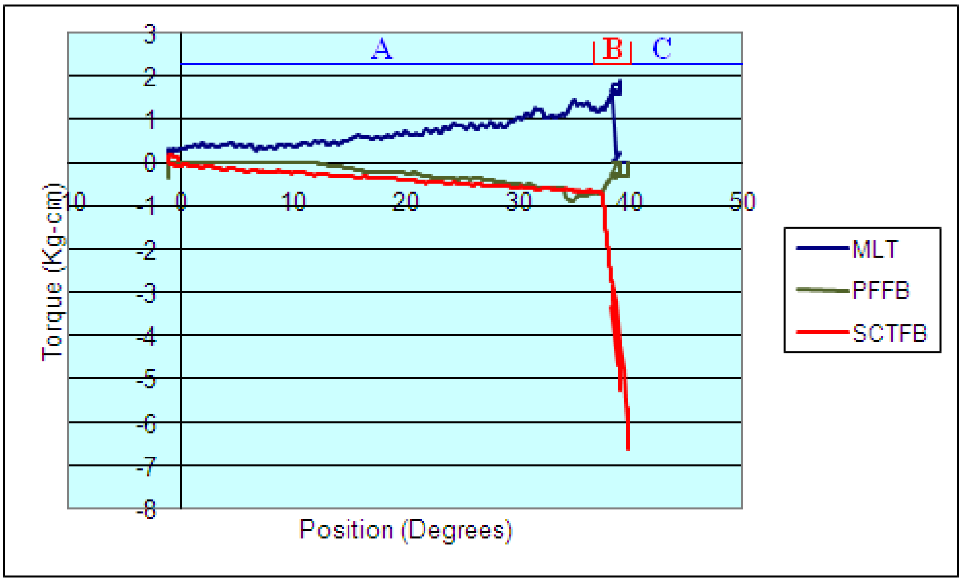



Figure 5.42: Variation of master side MLT, PFFB and SCTFB with respect to arm position for a speed of 13.26 mm/s (approx).

The MLT, PFFB and SCTFB behavior with respect arm position for a speed of operation 13.26 mm/s (approx) is as shown in figure 5.42. The SCTFB rise on right is due to the physical slave arm coming in contact with an object.

## 5.4.7 Observations

In results figure 5.33, 5.35, 5.37 and 5.39, in zone ‘A’ which is non-contact phase, the MLT clearly reflects SFFB i.e. the phantom feedback from MPC is the major feedback. Its effectiveness with BTM is established. Several salient observations are as listed -

1. The experimental master joint has a single light wheel alone as load on the shaft of motor used to cause constant speed input to the slave. Therefore velocity control loop's coordinating torque is consumed by the external loads namely PFFB and SCTFB.
2. For balanced slave arm, operating at constant speed, the SCTFB is very small and gains significant value only on interaction with environment. This happens on contact with external body only. The phenomena ensures that the PFFB and SCTFB are temporally exclusive to each other.
3. The unbalanced arm draws torque for working against gravity. A reverse force gain (rather attenuation) reduces the load effects reflected to the operator in case of heavy duty arms. Higher SCTFB causes continued effort spends by operator during normal unobstructed move in unhindered workspace too. So reducing it is favoured. This also reduces gravity effect. For the PFFB with high gain $G$ but high attenuation $\frac{1}{A}$ for SCTFB, the viscous drag can attain effectiveness in unbalanced arm also with some disadvantage of reduced contact feel from slave.
4. The foregoing discussion also suggest that since the phantom can sense contact with environment it can modify the feedback gain $\left(\frac{1}{\mathrm{A}}\right)$ of SCTFB to higher level to utilise the force generating capability of master joint and effectively stop the operator from moving on after slave makes contact to object and thereby mitigate the above problem.
5. For delicate tele-operations, on the other hand, the pay loads are small and the SCTFB is small. For increasing operator sensitivity the PFFB should be relatively higher. The arm design therefore should be of balanced type.

# 5.5 Generic Phantom for MPC Integration with BTMs

The experimental work established that two slaves, one real and one virtual can coexist (figure 5.43) in a system. The two feedbacks in master loop '$F_{am}$' caused by the physical slave and '$F_{fm}$' caused by the phantom slave are mutually separated on time scale and blend well without complication and can even be generated by the same force transducer used by BTM if electrical multiplexing is employed.

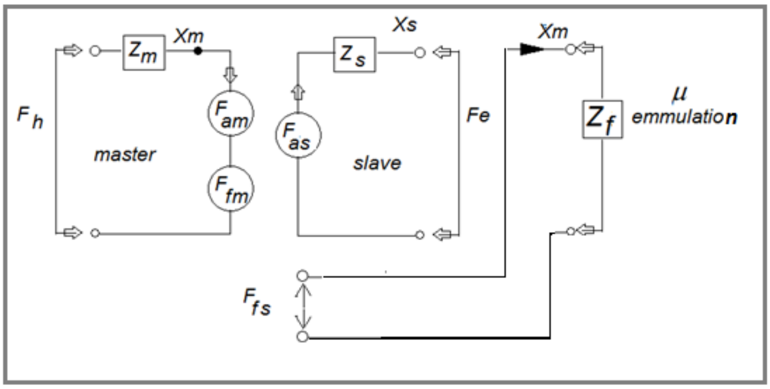



Figure 5.43: An integration approach for co-working of simplified MPC based phantom for percept rendering in integrated tele-control.

## 5.5.1 MPC Integration with Position Forward Telecontroller

The integration scheme is same as shown in figure 5.18. A force transduction by the hardware (figure 5.19) is sufficient for rendering the haptics effect on master interface. The haptics sensitivity can be designed by choosing an operating line in figure 5.20. The integration gives tremendous advantage as MPC generated PFFB is the only kinesthetic feedback available to remote operator who otherwise relies on closed circuit television CCTV feed from remote scenarios. However the haptic assist for the vicinity percept HAVP developed her needs the master side hardware to change in major way as force transduction is required. The HAVP also provides an effective working aid in situations where occlusion or lack of feel of depth in remote viewing persists. Functions of the two effects in master side loop can have variety owing to different types of BTM force reflection modalities.

## 5.5.2 MPC Integration with Position Forward and Force Feedback BTM Configuration

The experiments show that the PFFB and SCTFB work at different time. The former prior to contact and later after the contact. The later has very high magnitude as compared to PFFB which is more of a percept to induce the operator to slow down rather than stop him. The SCTFB is much higher in magnitude after contact and needs higher rating torque transducer. The same suffices for PFFB.

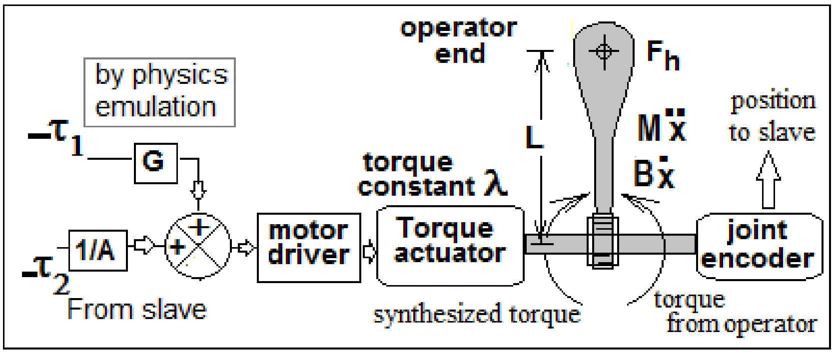



Figure 5.44: Viscosity emulated opposing drag integration for a single DOF.

In working masters the implementation is through one torque transducer on a joint only. The transducer (motor gearbox combination with motor operating on torque mode ) should be rated to supply enough torque to the joint in opposing direction to fully stop a well behaving operator. The feedback from the slave and Phantom are summed by electronic circuit to produce FFB on operator hand. The gain blocks $A$ and $G$ are programmable. Note opposite directions of feedback and operator action.

The PFFB & SFFB can be electrically added to control the master joint motor and the gains can achieved by use of suitable amplifiers with programmable gains. Single torque actuator by electronic control integration.

### 5.5.3 MPC Integration in Position Exchange BTM

This case relates to row d and e of Table 5.1. The master too has a position control loop and receives slave position as position input for a joint that is also actuated by a human. The opposition torque to operator comes from the master servo loop. The position exchange model of BTM gives feel of drag to the master depending on slave lag as compared to the master. In free space when the operator moves fast, owing to the difference of inertia on two sides the slave lags more and causes feel of inertial drag which is similar to that generated by the MPC model based phantom. Therefore additional percept has to be created in synchronism with the robot's nearing to object. For the HAVP type feedback to be perceived as sourced from the phantom.

## 5.6 Comparison of MPC Integration with Different BTM Models

Experiments in preceding section establish that the MPC model works best with balanced and light slave arm. This indicates that the technique can be very effective also for a different application like tele-surgery where slave ends fulfill both of these criteria. Several other salient observations related to MPC co-working with BTM are tabulated in Table 5.2. The two most popular classes of telecontrol implementations, the position forward force feedback type and the position exchange type have been considered. The cases (a) to (c) correspond to position forward type of telemanipulators. MPC function with these has been experimentally established in preceding sections. In table 5.1, cases (d) , (e) relate to PE type BTM , corresponding cases (d), (e) table 5.2 present inertia and drag effects of slave and master on master . Since MPC effect is similar to these, the drag effect does not stand apart.

**Table 5.2:** Salient aspects of MPC integration with widely used BTMs

| Telecontrol Method | Sensor for Phantom | Sensor for SFFB | Sensor on arm | Advantage/ Drawbacks | Remark |
|---|---|---|---|---|---|
| Case a | Position sensor | - | - | Only FB available ↑ | Blending needs a force actuator |
| Case b1 PFFB(I) | Position sensor | Motor Torque | No | No control element added to master ↑ | No additional sensor required |
| Case b2 PFFB(I) | Position sensor | Motor Torque | No | No control element added to master ↑ | Unbalanced arm needs additional proximal rendering |
| Case c PFFB(II) | Position sensor | Force | No | No control element added to master ↑ | Body sensor is a disadvantage |
| Case d | Position Sensor | Position | No | Percept does not stand apart ↓ | Needs additional proximal rendering |
| Case e | Position sensor | Position+ Force | Yes | Percept does not stand apart ↓ | Needs additional proximal rendering |

**Notes:** Body sensors include force torque sensors and not encoders. Any need of such instrumentation is considered as a disadvantage in hot cell ( column 6). The comparison suggests that an additional proximal sensor promise for MPC function enhancement.

# 5.7 Structured Sensing for Generalized Application of Phantom Framework II

Different models of slave robot arm can interact with the voxel occupancy model WVA. Their suitability for serving the MPC model varies as in table. The approach listed as ‘E’ has relatively more advantages.

**Table 5.3:** Performance of different slave models for sensing phenomena

| | **Robot Link Models** | **Interaction with Workspace Voxel** | **Remarks** (↓ disadvantage, ↑ advantage) |
|---|---|---|---|
| A | STL surfaces as sensor | i. Surface wading through fluid<br><br>ii. Voxels crossing the surface cause force development | **i.** Surface orientation based drag. So computationally difficult and erroneous.↓<br>**ii.** STL surfaces being widely varying in sizes, the method development and their validation both are difficult ↓ |
| B | STL nodes | Point with workspace voxel | **i.** Node spacing are shape dependent, gives rise to non-uniform sensor density↓<br>**ii**. Attached external nodes need multiple data sets and their synchronization can be erroneous ↓ |
| C | Line vectors parallel to link axis (figure 5.45 a) | Line with voxel | **i**. Sensors can move with link ↑<br>**ii**. Highly data and computation efficient (along vectors linear interpolation locates sensor). ↑<br>**iii**. It cannot effectively handle varying shapes of links ↓ |
| D | Link shell voxels on STL body shells | Point with voxel<br><br>Voxel model of links undergo rotations. | **i**. After (rotation/ translation) transform holes may get formed in robot surfaces from digitisation ↓<br>**ii**. Voxels of robot at different orientation than workspace voxel ↓ |
| E | Sensor structured by generalized cylinder method (figure 5.45 b) | Structured point with voxel | **i.** Polygon order can be matched with sensor pitch required. ↑<br>**ii**. Polygon pitch along axis also can be controlled as per sensor pitch ↑<br>**iii**. Lowest Error in shape trueness ↑<br>**iv**. It is amenable to optimisations for lean computation (Appendix B) ↑ |

Figure 5.45 below depicts the sensor arrangement . In following treatment sensor structuring framework has been developed based on this approach.

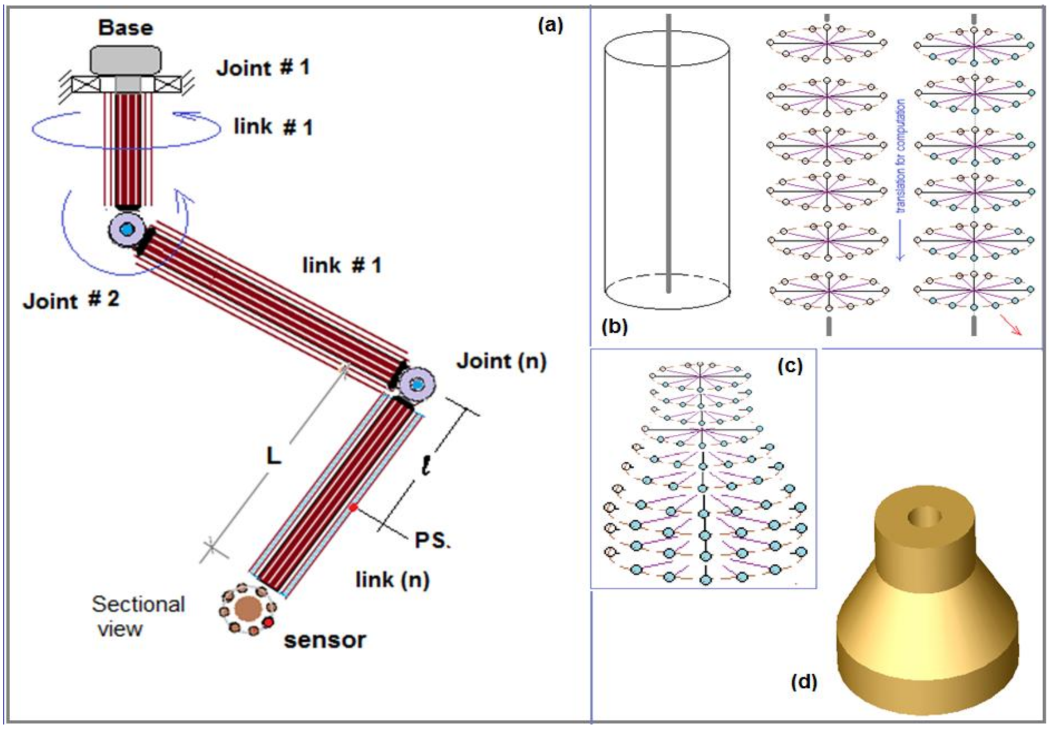



Figure 5.45: a; Suggested sensor along vectors parallel to link axis, b; Sensors on rings for cylinder; c; Similar approach for a varying section. A constant pitch spoke based sensor structure has varying radial pitch for conical links.

## 5.7.1 Integration of Point Sensors

The force sensed by point sensor can be integrated into an aggregate effect if sensors are embedded in robot arm surface. Force feedback interfaces can be viewed as having two basic functions to measure the positions and contact forces of the user's hand, and to display contact forces and positions to the user. The requirement of rendering can be met by structuring the sensors with the joint in mind.

## 5.7.2 Sensor Layout

Our approach is to use generalised cylinder model for links (chapter 3). A link comprises of multiple segments numbering M (integer) along single linear link axis at offsets $LJ_m$ for different segment definition structures (figure 5.46).

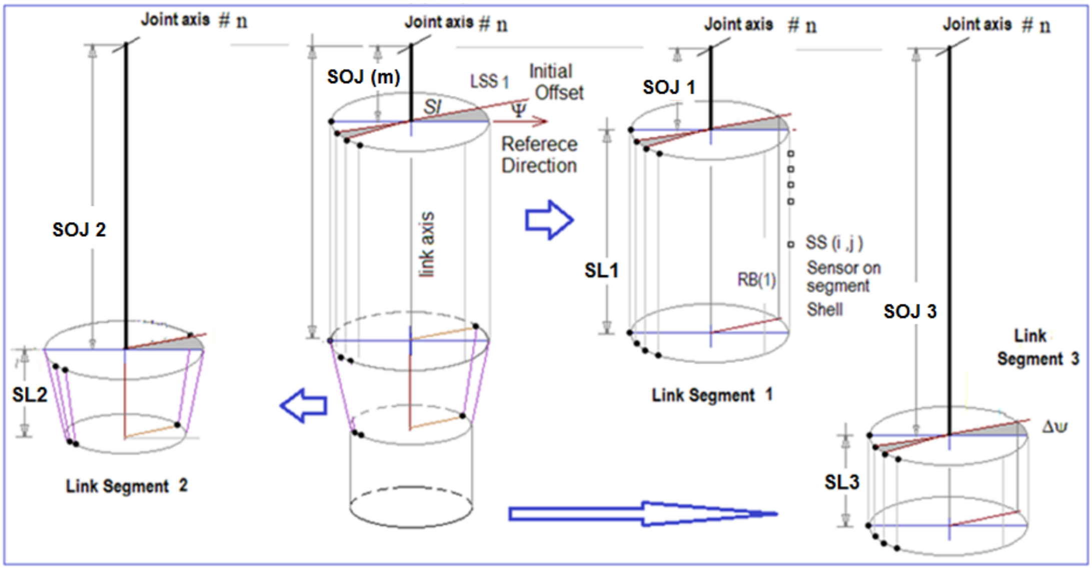



Figure 5.46: Link segmentation for uniform sensor placements.

Segments are of two types, a: constant section defined by single polygon (figure 5.47) along segment length and b: linearly tapering section polygon (figure 5.48) along segment length defined by polygon of order '$p$' at top named Polygon-top or PT with polygon_top-point $PTP_p$ and polygon of same order '$p$' at segment bottom named Polygon-Bottom or polygon_bottom-point $PBP_p$ (figure 5.49). For type 'a' segment, $PTP_{(p)}$ and $PBT_{(p)}$ are same so is computed in the sensor locating methods by linear translation from segment top to bottom by '$SL$' along the link axis. PT and PB data (figure 5.47) are arrays of vector $\boldsymbol{r}_{(\boldsymbol{p})}$, $\boldsymbol{\Psi}_{(\boldsymbol{p})}$ and array size is ( $p\_max\ + 1$ )

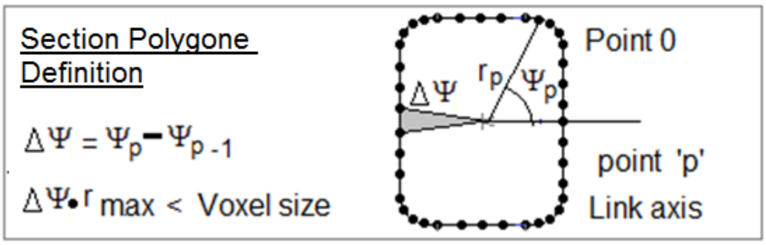



Figure 5.47: Generalised polygon defined link section for variety of cross section.

### 5.7.3 Multiple Segment in Link for Modularised Sensor Layout

The link being modelled as generalised cylinder for sensor placement it may have to be split in several segments. For example on the part with conical shape, number of sensors required for lower part shall be more (figure 5.48). As sensor to sensor distance is governed by voxel size, lower side shall need more than required for the upper side. A closer look shows that the two categorization of segment definition introduced above are very versatile to cover large types of arm.

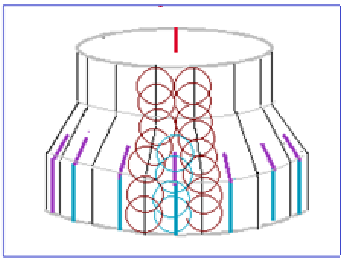


Figure 5.48: For a conical shape larger diameter side needs more sensors.

**Sensor Structuring:**

The point sensors are spatially structured to suit array based data structure for higher efficiency at run time (figure 5.47, 5.49, 5.50).

Range of '$p$' is $0 \leq p \leq P$ .

$SOJ_m$ is $m^{th.}$ segment's $PTP$ offset on link from link's parent joint axis

The PT and PB data are arrays of vector $r_p, \Psi_p$ and array size is$(P+1)$. For PT it is denoted as $r_{pT}, \Psi_{pT}$ , for PB it is denoted as $r_{pB}, \Psi_{pB}$

For segment type 'a', Polygon_Bottom $PB$ is only translated version of $PT$ and so $r_{pB}, \Psi_{pB}$ is same as $r_{pT}, \Psi_{pT}$.

Note that $\Psi_{pT}$ and $\Psi_{pB}$ are same.

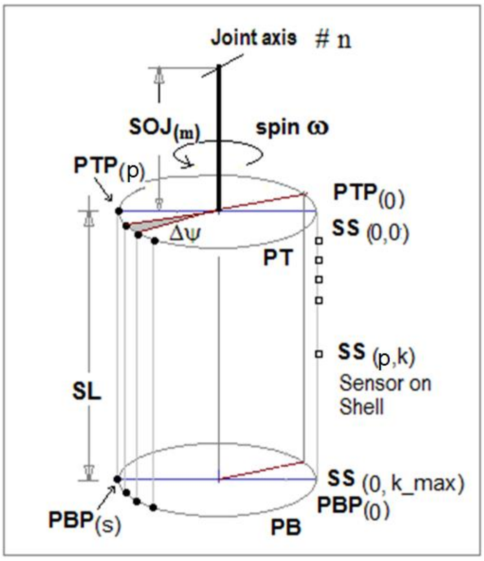



Figure 5.49: Type 1 segment

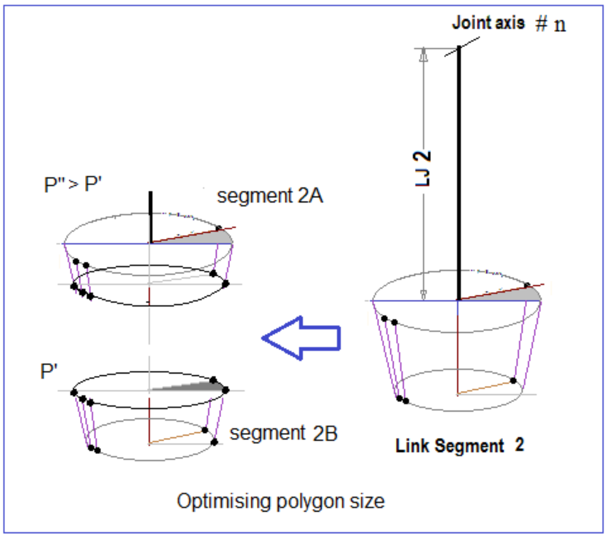



Figure 5.50: Type 2 segment

'$k$' number of sensors are placed along the shell surface Line $SSL_p$ at regular intervals. Here $SSL_p$ vector is defined as

$$SSL_p = \overrightarrow{PTP_p\ PBP_p} \tag{5.21}$$

The polygon points are chosen such that

$$\left|\overrightarrow{r_{pT}\Psi_{pT}, \mathrm{r}_{(p-1)T}\Psi_{(p-1)T}}\right| < voxel\ size$$

and

$$\left|\overrightarrow{r_{pB}\Psi_{pB}, r_{(p-1)B}\Psi_{(p-1)B}}\right| < voxel\ size \tag{5.22}$$

'$K$' is chosen for meeting following

$$\left(\frac{SL}{K}\right) < voxel\ size \tag{5.23}$$

Therefore Sensor on shell '$SS_{p,k}$' in a given segment is defined as

$$\begin{aligned} SS_{p,k}(x) &= \left(\frac{k}{K}\right) * \left(PBP_p(x) - PTP_p(x)\right) \\ SS_{p,k}(y) &= \left(\frac{k}{K}\right) * \left(PBP_p(y) - PTP_p(y)\right) \\ SS_{p,k}(z) &= \left(\frac{k}{K}\right) * \left(PBP_p(z) - PTP_p(z)\right) \end{aligned} \tag{5.24}$$

with respect to the $n^{\text{th.}}$ link's mother join $LJ_n$ coordinates $LJ_n(x, y, z)$

The position of the $SS$ in Cartesian space $CSS_{p,k}$ is then

$$\begin{aligned} CSS_{p,k}(x) &= LJ_n(x) + SOJ_m(x) + \left(\frac{k}{K}\right) * \left(PBP_p(x) - PTP_p(x)\right) \\ CSS_{p,k}(y) &= LJ_n(y) + SOJ_m(y) + \left(\frac{k}{K}\right) * \left(PBP_p(y) - PTP_p(y)\right) \\ CSS_{p,k}(z) &= LJ_n(z) + SOJ_m(z) + \left(\frac{k}{K}\right) * \left(PBP_p(z) - PTP_p(z)\right) \end{aligned} \tag{5.25}$$

$CSS_{p,k(t)}$ is x,y,z defined position of sensor in Cartesian workspace at time '$t$' similarly $CSS_{p,k(t-\delta t)}$is position at time $(t - \delta t)$.

$SSV_{p,k}$ is velocity vector for the sensor $SS_{p,k}$ . It is computed as

$$SSV_{P,k} = \frac{\left(CSS_{p,k(t)} - CSS_{p,k(t-\delta t)}\right)}{\delta t} \tag{5.26}$$

$$Sensor\ Drag = \left|SSV_{p,k}\right| * \mu_{\epsilon}.\mathrm{WVA}\Big(CSS_{p,k}(x), CSS_{p,k}(y), CSS_{p,k}(z)\Big)$$

(5.27)

Where $SS_{p,k}(t)$ is sensor position at time instant '$t$' and $SS_{p,k}(t-\delta t)$ is sensor position at time instant $(t-\delta t)$. $\mu_{\epsilon}$ is either basic or processed version of it.

$$SSV_{p,k}(x) = \left\{\begin{array}{c} \left[LJ_n(x) + SOJ_m(x) + \frac{k}{K} * \left(PBP_p(x) - PTP_p(x)\right)\right]_{(t)} \\ -\left[LJ_n(x) + SOJ_m(x) + \frac{k}{K} * \left(PBP_p(x) - PTP_p(x)\right)\right]_{(t-\delta t)} \end{array}\right\} \Big/ \delta t$$

$$SSV_{p,k}(y) = \left\{\begin{array}{c} \left[LJ_n(y) + SOJ_m(y) + \frac{k}{K} * \left(PBP_p(y) - PTP_p(y)\right)\right]_{(t)} \\ -\left[LJ_n(y) + SOJ_m(y) + \frac{k}{K} * \left(PBP_p(y) - PTP_p(y)\right)\right]_{(t-\delta t)} \end{array}\right\} \Big/ \delta t$$

$$SSV_{p,k}(z) = \left\{\begin{array}{c} \left[LJ_n(z) + SOJ_m(z) + \frac{k}{K} * \left(PBP_p(z) - PTP_p(z)\right)\right]_{(t)} \\ -\left[LJ_n(z) + SOJ_m(z) + \frac{k}{K} * \left(PBP_p(z) - PTP_p(z)\right)\right]_{(t-\delta t)} \end{array}\right\} \Big/ \delta t$$

(5.28)

**Link Definition {**

**Rotation axis** $\big(P_a(x,y,z)P_b(x,y,z)\big)$;

**Link velocity** $\theta'$;

**Link initial position** $\big(LJ_n(x,y,z){:}\,LJ_{n+1}(x,y,z)\big)$ ;

**No. of segments** $M$;

**Segment definition # 1 {**

**Segment sensor gain** $SG$**;** (optional)

**Type** $a$ or $b$;

**Segment top offset from joint** $SOJ_m$;

**Segment length** $SL$;

**Segment initial spin** ω;

$PPT(p)$ array $r_p, \Psi_p, p$ in range *0:P*;

**Number of sensors along shell length** $k$;

**}**

**Segment definition #2**{ **...**

}

.

**Segment definition # M { ...**

**}**

**}**

Note that positions $LJ_n(x, y, z)$ , $LJ_{n+1}(x, y, z)$ rotation axis and link positions are computable using kinematics model and joint parameters using equation (2.2).

## 5.7.4 Identification of Forces Working on a Sensor

The sensors on link sensors face same motion as the link segment body. These can be rotation, translation or combination of these. A robot joint causes an actuation and so causes drag on sensor, since sensor point on body undergoes a motion in Cartesian workspace of robot ' $WVA$' having virtual effect attribute in form of $\mu$. A single joint causes a single motion. This motion is purely rotation or purely translation for the joint. It affects the child link and subsequent lower links of the successive children joints. Force on a sensor on segment undergoing rotation motion can be computed in following manner depending on the following cases -

Case a: rotation axis along the link axis .

Case b: rotation axis is orthogonal to link axis. This can have two variants based on the condition: i- immediate parent joint axis is orthogonal rotation axis: ii – motion inherited from higher order joint axis as non-orthogonal rotation axis

**5.7.4a – Drag estimation for rotation axis along link axis :**

$\mu_\epsilon . WVA\left(CSS_{p,k}(x), CSS_{p,k}(y), CSS_{p,k}(z)\right)$ is written as $\mu_\epsilon . CSS_{p,k}$ in short

From figure 5.47 the $PTP_p$ ring causes a spin torque $S$ on link axis at segment '$m$' of link '$n$'for a rotation speed $\omega$ is given by

$$SLJ_n = \sum_{p=0}^{P} \left[ r_{pT} * \left( r_{pT} * \omega * \mu_\epsilon . CSS_{p,k} \right) \right] \qquad (5.29)$$

the $PBP_p$ ring causes a spin torque with the array $r_p$ $\Psi_p$ (figure 5.47) defined for $PBP$

$$SLJ_n = \sum_{p=0}^{P} \left[ r_{pB} * \left( r_{pB} * \omega * \mu_\epsilon . CSS_{p,k} \right) \right] \tag{5.30}$$

At rotation speed $'\omega'$ the $SS_{pk}$ ring causes a spin torque

$$SLJ_n = \sum_{p=0}^{P} [( r_{pT} + (r_{pB} - r_{pT})k/K)^2 * \left(\omega * \mu_\epsilon . CSS_{p,k}\right)] \tag{5.31}$$

Total Spin Torque from complete segment ‘m’ on joint $LJ_n$ denoted as $S_m LJ_n$ is computed as

$$S_m LJ_n = \sum_{k=0}^{K} \sum_{p=0}^{P} [ ( r_{pT} + (r_{pB} - r_{pT})k/K)^2 * \left(\omega * \mu_\epsilon . CSS_{p,k}\right)] \tag{5.32}$$

*Total Spin Torque on $LJ_n$ from all segments of the link joint is*

$$TTLJ_n^{spin} = \sum_{m=0}^{M} S_m LJ_n \tag{5.33}$$

Its direction is opposite to the spin rotation

### 5.7.4b - Drag estimation for rotation axis not along link axis

In these cases the sensor distance from the rotation axis may be computed in two ways depending on the conditions below. It may be noted that treating the cases differently reduces computation load. Otherwise the later case below in ‘b’ can be generalised for all cases.

### 5.7.4b(i) - Drag estimation for links with Immediate Parent Joint axis as Orthogonal Rotation Axis :

The sensor rings at progressing distance from the link top are considered at distance defined by $k/K$ ratio of link length and for rotation speed $\theta'$ it is computed as

$$TTLJ_n^{ORD} = \sum_{m=0}^{M} \sum_{p=0}^{P} \sum_{k=0}^{K} \mu_e . CSS_{pk} * \left(\frac{k}{K} * SL_m + SOJ_m\right) * \theta' * \left(\frac{k}{K} * SL_m + SOJ_m\right) \quad (5.34)$$

Where ORD stands for 'Orthogonal Rotation Drag', $SOJ_m$ is $PTP$ offset of segmentation on link from joint axis, $SL_m$ is length of $m^{th}$ segment.

**5.7.4.b(ii) Drag estimation for links with higher order Joint axis as Non-orthogonal Rotation Axis :**

In this case the Sensor rotation occurs around a generalised axis at speed $\theta'$ in space which is the rotation axis of some joint of the robot arm and hence is well defined. Considering $DSS_{p,k}$ as distance of sensor $SS_{p,k}$ at sensor location $CSS_{p,k}$ from the axis of rotation 'ζ' that causes motion of the sensor $SS_{p,k}$ around it the magnitude of drag torque is computed as following . The direction is opposite to rotation direction of axis motor causing motion.

$$TTLJ_n^{NRD} = \sum_{p=0}^{P} \sum_{k=0}^{K} \mu_\epsilon . CSS_{pk} * DSS_{pk} * \theta' * DSS_{pk} \quad (5.35)$$

Where NRD signifies non-orthogonal rotation drag.

Computation of $(DSS_{pk} * DSS_{pk})$ is done as below-

$$\left(DSS_{p,k}\right)^2 = \left(CSS_{p,k}(x) - I_{p,k}(x)\right)^2 + \left(CSS_{p,k}(y) - I_{p,k}(y)\right)^2 + \left(CSS_{p,k}(z) - I_{p,k}(z)\right)^2 \quad (5.36)$$

Where $I_{p,k}$ is intersection of normal from $CSS_{p,k}$ point on the rotation vector $\zeta$. The vector $\zeta$ is rotation axis of an upper joint relative to the host segment of present sensor $SS_{p,k}$ and is computed by using the robot kinematic model and robot joint parameters.

#### 5.7.4c Drag estimation on sensor for translation motion-

Opposition force caused by translation acts simply opposite to translation direction and is dependant on translation velocity at sensor on link body and $\mu_e$ at the location where sensor exists in $WVA$.

$$Sensor\ Drag = \mu_e.\left|SSV_{p,k}\right| \text{ at } \left(CSS_{p,k}(x), CSS_{p,k}(y), CSS_{p,k}(z)\right) \tag{5.37}$$

For detailed treatment on drag estimation see appendix 'c'.

### 5.7.5 Notable Feature of Sensor Structure

The foregoing sensor structuring allows minimisation of real time computation by determining sensor location in workspace by computation of segment top sensor polygon and sensor bottom sensor polygon, while for large number of sensors in-between, efficient linear interpolation is used. For long arms the gain is considerable and computation per sensor reduces with length to cross-section ratio.

## 5.8 Salient Observations on Phantom Function

The approach here has successfully formed a two tier method for solving the remote perception synthesis. The neighbour forming and its sensitisation from the CAD STL object model is automation friendly for all shapes. The other part is the sensor. A phantom framework streamlines the entire remote perception implementation scheme by structuring process. A typical framework comprises a virtual robot, a virtual workspace and a robot control system that mimics the slave robot motion in synchronism with the slave robot operation in real-time.

An efficient sensing scheme with moderate complexity has evolved. Our approach attempts to modify the free space impedance in near vicinity artificially and so transparency in true sense is compromised. Since the effect is velocity dependent, at lower velocity the deviation from the otherwise natural feel caused by the coordinator torque in free space is preserved In non vicinity area, the transparency is unaffected. The objective here is to create a perception of this vicinity space by a drag feel rendered to operator moving at high speed. At low speeds the drag sensation occurs at very close vicinity leaving the naturally free feeling workspace intact to a larger extent.

The percept HAVP is a sensitization tool and not intended to stop the operator. The historic equivalent of Wall model [87] and its limitation in producing high stiffness is not an impediment as the impedance transition starts earlier than contact and the operator is sensitized by crisp interaction. The phantom framework achieves fusion of traditional BTM's force feedback and virtual drag to augment tele-perception by giving feel of presence.

# Closure

The chapter has established MPC model's success in co-working with BTM for the version used in hot cells and demonstrated HAVP function. Section 5.8 has focussed on the notable aspects of the phantom system and forming percept if further enhancement is desirable. The model should aim at creating a percept to the operator of tele-manipulator for inducing some behaviour that can help or support a procedure of the normal work intended and pursued by the operator without taking away control from him/her. In that sense it is a procedure oriented support to operator

However the method of communicating the spatial condition viz. a vie workspace part by haptic rendering can disturb a manipulation action underway as the method works by producing jerks. Fine tuning of percept with consideration to 'just noticeable difference' [88] may be effective.The percept must be basis for other methods of creating a more complete rendering in multimodal fashion to reduce dependence on the haptic route alone. Also in recent times research has indicated that haptic and cross-modal methods have advantage. Aspects of visual and haptic cross-modal integration have been treated from two perspectives: attention intended to be cast in a given situation and dominance. Haptic rendering is often presented along with visual display to address this duality and our percept must support this functionality. The attention drawing ability is related to the phenomena of 'standing out' in given context and dominance is related to inflow of distal perception. A mix of two must be aimed at.

# CHAPTER 6

## ENHANCEMENT OF VIRTUAL SENSING BY PHANTOM

## AND

## EVOLUTION OF MULTIMODAL PERCEPT

In forming feel of presence, while sensory fidelity is important from a technical point of view for immersion, the user's sense of "being there" is qualitative in nature. The degree of presence is related to the user's subjective psychological response to a Virtual reality scenario presented to him. The scenario presentation is designed to reach human brain for forming a perception and elicit a response. The user response is individual and context dependent and its success lies in appropriateness of invoking distal percept.

# 6.1 Multimode perception from MPC model

Use of multimode perception has been reported using auditory and visual methods. These have sound backing of well formed concepts.

## 6.1.1Concept of Auditory Percept

It is known that 3D sound simulation can improve the capabilities of virtual reality systems dramatically [89, 90, 91, 92]. But, realistic 3D sound simulations demand a tremendous amount of computational power to calculate reverberation, occlusion and obstruction effects and so are costly. Doerr et al [93] have attempted to use 3D sound in a training system as a way to direct and guide trainees on aircraft cockpit simulators to observe specific events in 3D space. An auditory percept can fall upon operator independent of the kinaesthetic interface and so offers more flexibility in implementation. A more evolved audio-percept can be formed to communicate a relative position of contact with environment using the same framework as the position of contact in workspace can be coded by intensity mapping on stereo or surround sound audio speaker systems.

### 6.1.2 Concept of Visual Percept

In developing perception of real object based on virtual haptic interaction, the percept is often presented along with visual display. The issues involved in cross-modal interaction have been addressed in past research. Klatzky and Lederman [60] discuss aspects of visual and haptic cross-modal integration from the perspectives of attention and dominance.

Spence et al. [94] have studied how visual and tactile cues can influence a subject's attention and observed that visual and tactile cues are treated together in a single attentional mechanism and wrong attention cues can affect perception negatively. Preference of sensory dependence has been studied by analyzing perceptual discrepancies in application scenarios where cross modal integration yields a unitary perceptual response. One example is the detection of object collision. During object manipulation, humans assess whether two objects are in contact based on a combination of visual and haptic cues. Though early studies of sensory dominance pointed to a strong dominance of visual cues over haptic cues [95], but later research in late 1990s and early 2000 have lead the psychologists to agree that sensory inputs are weighted based on their relative appropriateness or functional reliability, measured in terms of accuracy, precision, and cue availability [96, 97, 33]. The design of contact determination algorithms can also benefit from existing studies on the visual perception of collisions in computer animations. O'Sullivan and her colleagues [98, 99, 100] have investigated different factors affecting visual collision perception. Their work demonstrated the feasibility of using these factors for scheduling interruptible collision detection among large numbers of objects.

# 6.2 Phantom Version III

## A Framework for Multi-Modal Percept Formation

The revised framework (figure 6.1) hosts a sensor for different percept that is more suited to distal percept formation using synthesized view assist and supporting telecontrolled slave manipulator's working in vicinity of parts. This distal percept is called View Assisted Distal Percept (VADP) hereafter in this work. Its working evolves in following section. The framework now includes a graphic rendering system in addition to force transducer. The graphics subsystem maintains a virtual workspace using CAD models of workspace parts in workspace as per their real world instance which can be visualised through the graphics renderer pipe line provided a 'look from' and 'look at' points in the workspace are provided to it. The virtual domain supporting the sensing has voxelised workspace version with layered shells. The layering process is same as detailed in chapter 4 and so is not repeated here. The synthesised view based percept can work as proximal as well as distal percept with a good variety of context based rendering mainly oriented to support the operator. However for its success, the framework needs improved sensing capabilities. The enhanced sensors also improve the function of auditory percept parameter, as in the more evolved phantom framework. The percepts can be in different domains like auditory and visual domains. The MPC generated response parameter is treated as Percept-Parameter 'PP' similar to 'RP' but is produced by more evolved sensing processes.

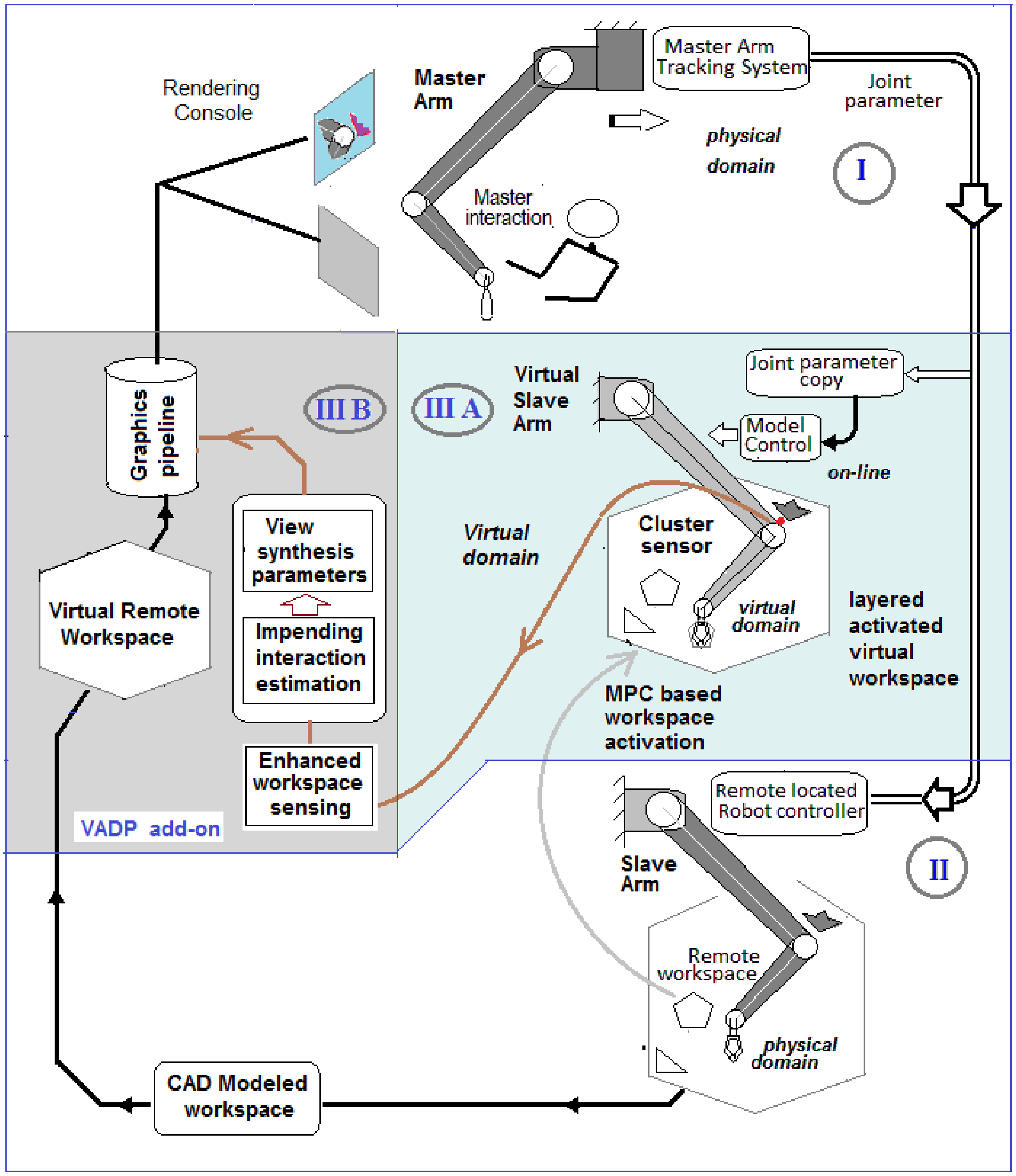



Figure 6.1: Phantom III ; an enhanced sensing and percept developer framework.

## 6.2.1 Sensor Enhancement for Improving Haptic Assisted Vicinity Percept:

The MPC based sensor developed in chapter 5 may be evolved further by use of array of sensors and developing cluster sensing techniques around it. A design framework is evolved here to facilitate array sensing in voxelised, activated, virtual work-space and build Enhanced Sensors 'ES' of several types. In figure 6.1 View Assisted Distal Percept

VADP is formed by adding III B segment to III A i.e. the phantom devised in chapter 5. Enhanced workspace sensor is the major component of III-B.

## 6.2.2 Enhanced Sensor Design Framework

The sensing so far is at a point and uses relation (4.7). Enhanced sensor 'ES' having improved performance and versatile functionality can be formulated by array sensing.

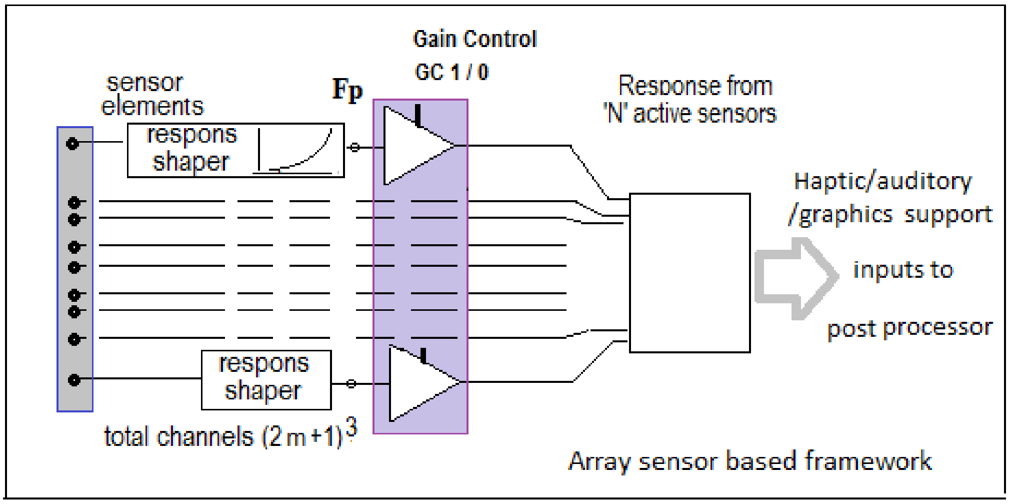



Figure 6.2: Structure of enhanced sensor in phantom sensor framework.

A design friendly form of 'ES' (figure 6.2) has cluster of point sensors that coincide with neighbour voxels if central point sensor is located at $V_{(x,y,z)}$ in virtual workspace . Thus $V_{(x,y,z)}$ is in WVA and is accessed by indices $X, Y, Z$. For flexible encoding of neighbour effect, response shaper is used. It assigns a scaling factor to sensed $\mu$ values as per a designed linear or nonlinear transfer function. Sensed zones in WVA which are primarily cube shaped, are changed to other shape by gain control 'GC' in *shaping element* 'SE'. For example cells at cube corners are out of a fully contained sphere in cube, so GC is '0' at these. Resultant response $F_r$ is formed by some form of aggregation (figure. 6.2).

### 6.2.2a Implementation of Enhanced Sensor-Type 1

A Sensor Local Array (SLA) is formed as eight connected senor grid set around a center voxel $SZ_{(i,j,k)}$and is of type 'integer' (figure 6.3). It has $(2m + 1)$ layers of $(2m + 1)$ long and $(2m + 1)$ wide sensor grids. Its indices $i, j$ and $k$ align with $x, y$ and $z$ directions respectively and range from $-m$ to $+m$ for ensuring that each sensor is located in separate voxel if center location $SZ_{(0,0,0)}$ is located at a sensing

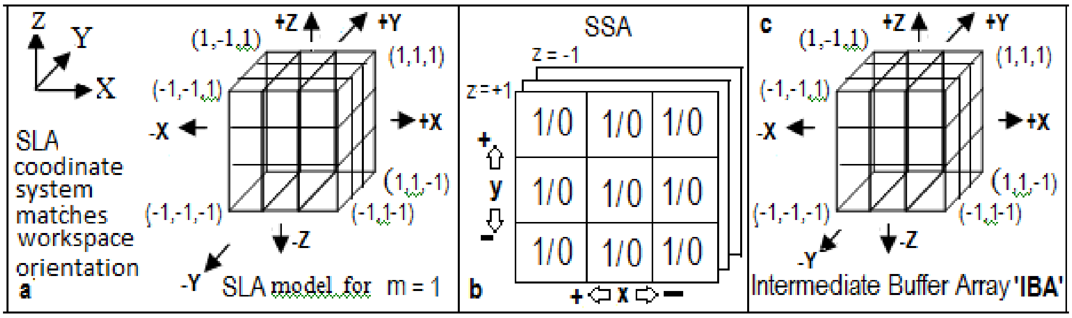



Figure 6.3 a; Sensor local array, b; Sensor shaping array for forming other solid shapes within SLA, c; intermediate buffer array.

location. SLA is filled with sensed voxel's $\gamma$ from its voxel record.. A Sensor Shaping Array 'SSA' is also similarly formed but it holds binary value for use as control mask. A 'zero' value switches sensor 'off' at corresponding location. Intermediate bufer $IBA_{(i,j,k)}$ ,formed similarly, has integer data in cells.

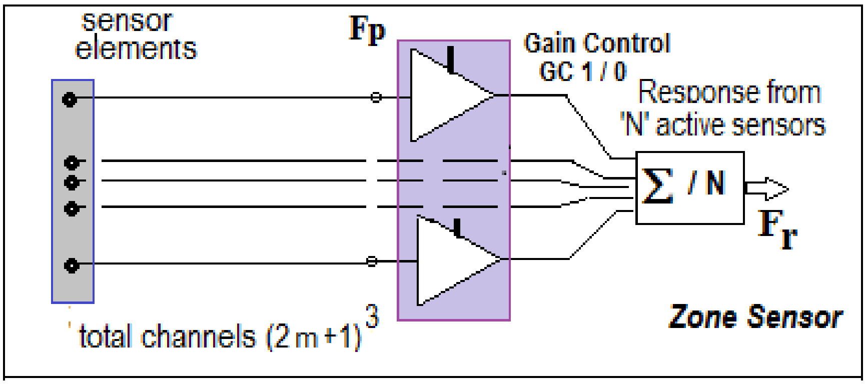



Figure 6.4 Enhanced sensor type-1 averaging sensor .

*Averaging sensor implementation :*

For $(-m \leq i \leq m, -m \leq j \leq m, -m \leq k \leq m)$

$SSA_{i.j.k} = 1$, if $(i^2 + j^2 + k^2) \leq m$ else $SSA_{i,j,k} = 0$;

For $(-m \leq i \leq m, -m \leq j \leq m, -m \leq k \leq m)$

$IBA_{i,j,k} = SLA_{i,j,k} * SSA_{i,j,k}; W = W + SSA_{i,j,k}$

$$F_r = V * \sum_{i,j,k=-m,-m,-m}^{i,j,k=m,m,m} \frac{IBA_{i,j,k}}{W} \qquad (6.1)$$

### 6.2.2b Enhanced Sensor Type 2

The type 1 enhanced sensors can only shape a zone as per discrete value and perform averaging over the zone at very low cost of computation by using Boolean operations mostly. The type 2 enhanced sensors are similar in structure to type 1 with the only difference that the SLA array and the IBA arrays are fraction data in range 0 to 1. All mathematical operations are floating point. The SLA array is loaded with precision float data to control output for response shaping of each SLA zone contributor Voxel, as per the SLA gain. The gains are decided by some criteria. For example for the voxels on the periphery of sphere in SLA weight will be assigned as fractional value indicating how much of the voxel belongs to sphere (figure 6.5). Owing to variable weight, the computation time increases as floating point computations increase but extra constraint on sphere size constraint are applicable. The sphere shaping distortion which is proportional to the fraction $\left(\frac{no\ of\ periperal\ voxels}{no\ Of\ body\ and\ periperal\ voxels}\right)$ and occurs in binary cell SLA, is eliminated. Cluster sensing for it appears in figure 6.6 .

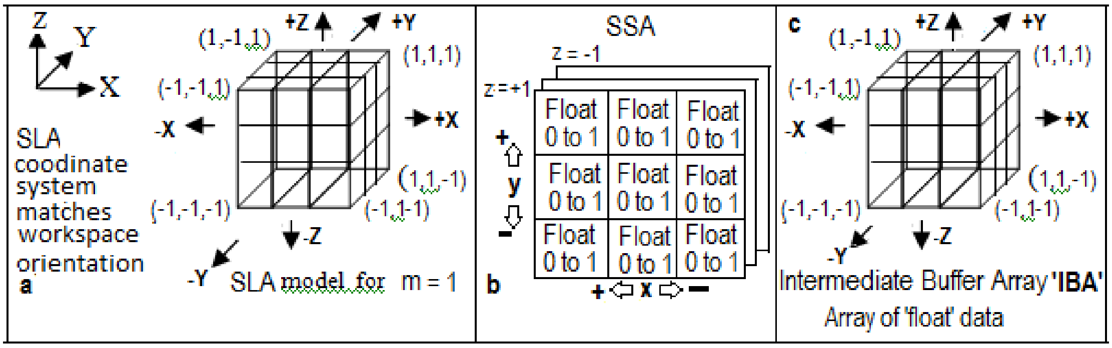



Figure 6.5 a; Sensor local array, b; Sensor shaping array of float data for forming weighted volumes, c; intermediate buffer array of float type.

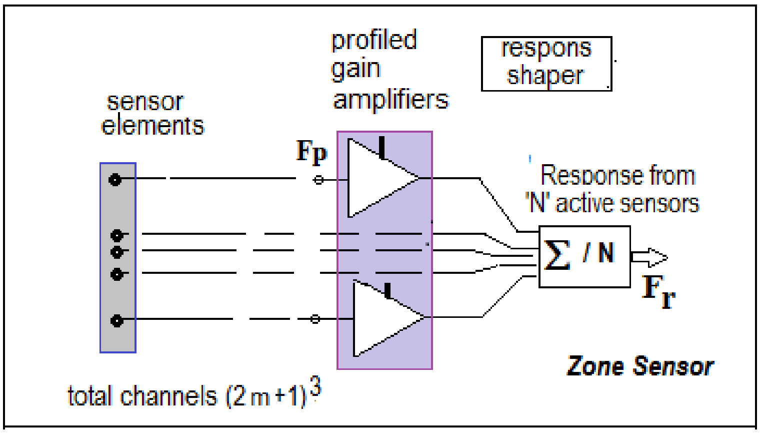



Figure 6.6: Enhanced sensor type-2 cluster formation of accurate shape by elemental weight control on sensor element followed by averaging

# 6.3 Salient Applications of Enhanced Sensor

The enhanced senor has many applications, the salient among these are as below-

## 6.3.1 Smooth Response Development by Haptic Probe

Spatial low pass filters are effective in smoothening point sensor response. A smooth sensor generates changing response through multi voxel layer as the proximity varies within thickness of layer too. It has higher spatial resolution matching voxel size rather than layer thickness which has lower bound dictated by layer shape fidelity. Iso-symmetry is formed by assigning '1' to SSA cells within radius '$m$' and total weight $W$ of SSA is found by using equation (6.1).For comparison the buffer based results from chapter 4 are reproduced here.

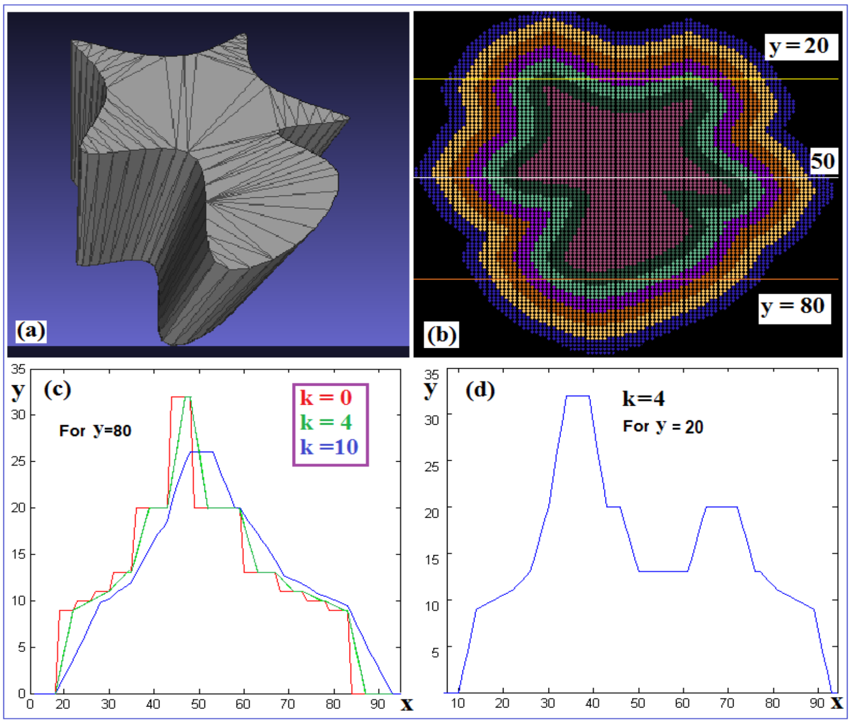



Figure 6.7 a; Complex cylinder as STL model, b; its voxel neighbourhood c; $F_p$ smoothing on y=80 track and d; on track y=20.

In figure 6.7 (a) shows the complex cylinder as STL model, (b) shows its voxel neighbourhood with 6 layers of μ using spherical FE, (c) shows the result of $F_p$ smoothing using $\alpha_\kappa$=1 for all $k$ and $k$ = 0, 4 and 10 on a track at lower side of object ($y$ = 80). It shows that high k value yields smoother $F_p$ but causes delayed sensing. ( d) shows $F_p$ for $k$ = 4, $\alpha_\kappa$ =1 for sensor moving on upper side (y=20) .

For the same test case as in figure 6.7, enhanced sensor using shaping element (SE) of spherical type and different radius yield superior smoothing using relation (6.1) without delay in sensor response (figure 6.8 ) unlike that prevalent in buffer approach (figure 6.7). Sphere SE uses nearly 48% less computation than a cube shaped SE, has better symmetry of sensing around its location and forms smoother '$F_p$' that is now used as percept parameter PP also Performance of point, cube and sphere SE of $m = 1$ and $m = 2$ confirms this observation (figure 6.8). Comparison of performance show that best results are achieved for the larger sphere in SLA 5x5x5 for 4 voxel thick layers of neighbor coding.

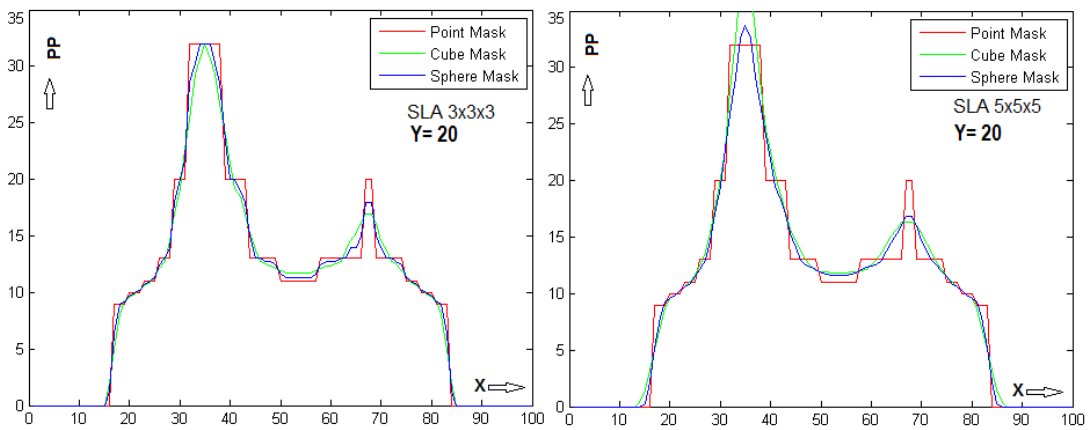



Figure 6.8 (left): relative performance of $F_p$(red) $F_r$ cube(green) and $F_r$ sphere (blue) SE of sizes 3x3x3, (right): SE of size 5x5x5 near complex cylinder .

### 6.3.1a Advantages of enhanced sensor:

The enhanced senor using cluster sensing offers several advantages. The haptic probe's smoothing is superior to the history- buffer based approach as there is no delay effect and owing to elimination of history the persistent data size reduces leading to huge memory saving when large number of sensors is considered in structured sensor approach developed in Chapter 5.

## 6.4 New Sensing Modalities of the Haptic Sensor

Several new sensor configurations can be derived from applying the cluster of sensors. These are described below

### 6.4.1 Netting Sensor

The point sensors structured in chapter 5 work in virtual fluid environment as mesh of robot link body shape. Their replacement by enhanced sensor leads to their function as netting-like model wherein the fluid passing through the net causes the drag effect. The parameter '$m$' (section 6.2.2) when set equal or larger than the sensor pitch on link body gives the effect of netting as all voxels of the workspace passing through SLA then contribute in the $\mu$ computation of the local zone formed by the surrounding body sensor on link body.

### 6.4.2 Sensor for Slender Link Bodies

The enhanced sensor can be seen as spherical sensors of diameter '$2m + 1$' placed along the axis of link body to cover the length of robot link. This reduces the computations at run time as less number of computations has to be done for computing $SS_{p,k}$. An experiment on the same setup as used in Chapter 5 shows following results..With 20 sensor locations along link (cluster radius 3 voxels) the percept variation with distance to object is smooth and monotonic with approach to object. It matches with 60 sensors on as many locations on link. This has

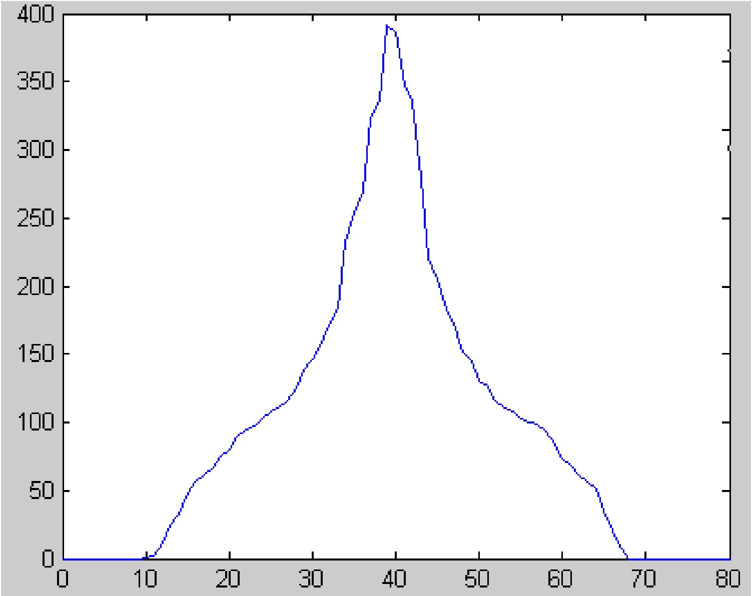



Figure 6.9: Drag percept from 20 enhanced sensors with m=*3* for a unit velocity.

tremendous advantage in real time application where fewer sensors on robot link shell means lower computation load for torque calculation.

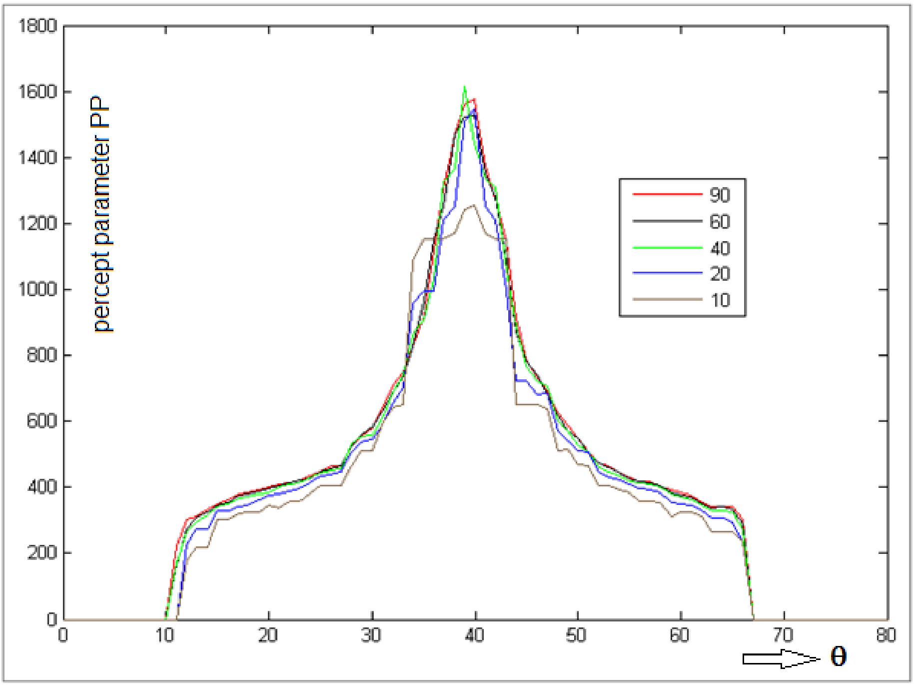



Figure 6.10: Drag percept from 20 enhanced sensors with *m=3* plotted along with that for 40, 60 and 90 point sensors (type-simple sensor).

However the cluster sensing, though fairly straight forward has some processing load for array accessing, addition and averaging using convolution based method which are fast

### 6.4.3 Finger Sensors

Fingers of robot are thin and can be made more amenable to sensorisation by using enhanced sensors. This can be attributed to following major factors.

Most significant advantage accrues from the fact that MPC reaction direction being only motion direction dependent, the contact direction in workspace need not be sensed by the sensor and for a symmetric sensor configuration, which sensor generates the signal need not be identified. Additional factors in its favor are

a: their shells are small

b: they being at the lowest end of kinematic chain of robot arm need more computation to determine sensor position in workspace for estimating the corresponding $\mu$.

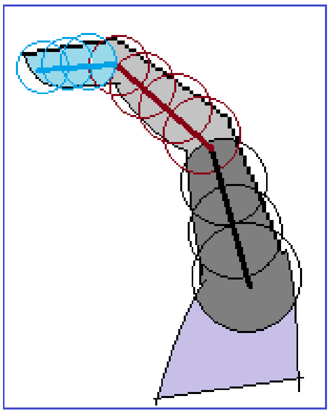


Figure 6.11: Finger's haptic sensing scheme formed by enhanced sensor with progressively reducing '$m$' along link elements' axis(finger segment) .

c: enhanced sensors placed along the finger segment axis at pitch equaling '$m$' can cover the finger body well. Note that differently colored sets undergo different kinematics computations only for the cluster center sensor that takes advantage of linear interpolations as in chapter 5. Generally '$m$' reduces towards finger tip.

d: The enhanced sensors being spherical need only positioning of their centers under dynamic condition of the hosting robot and no rotation has to be computed [101].

# 6.5 Auditory Percept Formation from MPC

The novelty of the audio percept lies in the tight synchronization of the auditory mode and the haptic mode. It is possible owing to relatively less bandwidth required for audio rendering as compared to that for synthesized graphics. User interaction with the activated environment generates feedback forces. These forces are rendered to the hand by a haptic force feedback device, and to the ear as amplitude and tone controlled sounds. This is more than synchronizing two separate events. Rather than triggering a pre-recorded audio sample or tone, the audio and the haptics change together when the user applies different velocities to the object. For the purpose of driving different interfaces, the response from the MPC model to a dynamic robot is generically labelled as 'percept parameter' PP, in following sections.

## 6.5.1 Enhancement of MPC Function for Force Feedback BTM

The MPC model maintains passivity in operator controlled motion loop as stopping motion causes $F = 0$ condition and operator can hold the robot at place in vicinity of a part. However the operator should continue to perceive vicinity in some way, else sudden motion can be initiated by him/her in close vicinity zones in absence of vicinity feel and in systems having limited bandwidth for feedback, contact with considerable mechanical impact can occur. Solution lies in using $\mu$ value in the layered vicinity in an ingenious way to form multimode interface at master. Multimode interface is formed by feeding tones to a audio speaker based on $\mu$ (figure 6.12). While the PP is used for developing opposing torque, $\mu$ dependent variable frequency audio tone conveys the

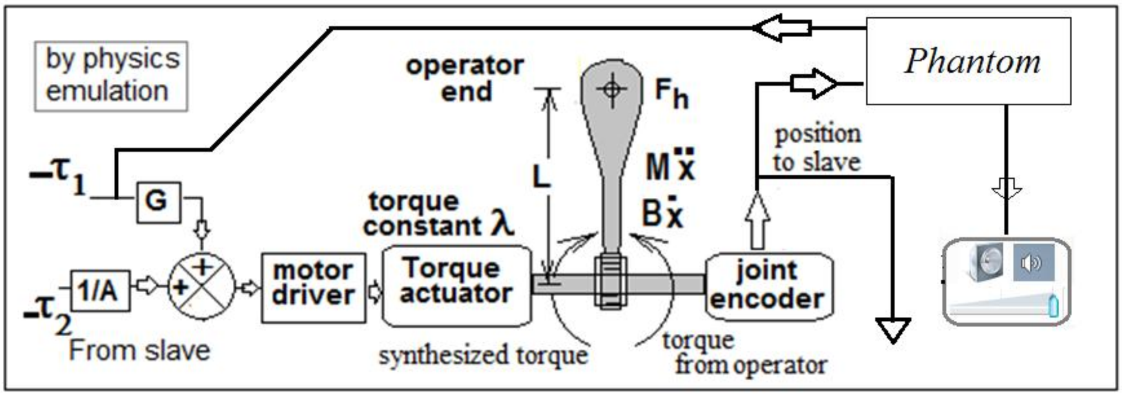



Figure 6.12: The audio percept is rendered simultaneously with haptic percept for force feedback type manipulators.

relative closeness to the object even when robot is held standstill by operator. The amplitude is related to speed of slave with a non zero minimum assigned to zero speed . The audio percept support to the haptics percept is necessary in two cases-

a: The robot movement is slow or zero near object .

b: The slave arm is unbalanced and so causes high background force on the haptic interface as common mode force signal to operator. The tone conveys the nearness.

## 6.5.2 Auditory Percept Formation

A combination of both modulations is applied by forming tone with base amplitude and then applying gain to it based on velocity by some non linear or linear law. Note that minimum gain value is one. So a basic tone signal is audible even at zero speed.

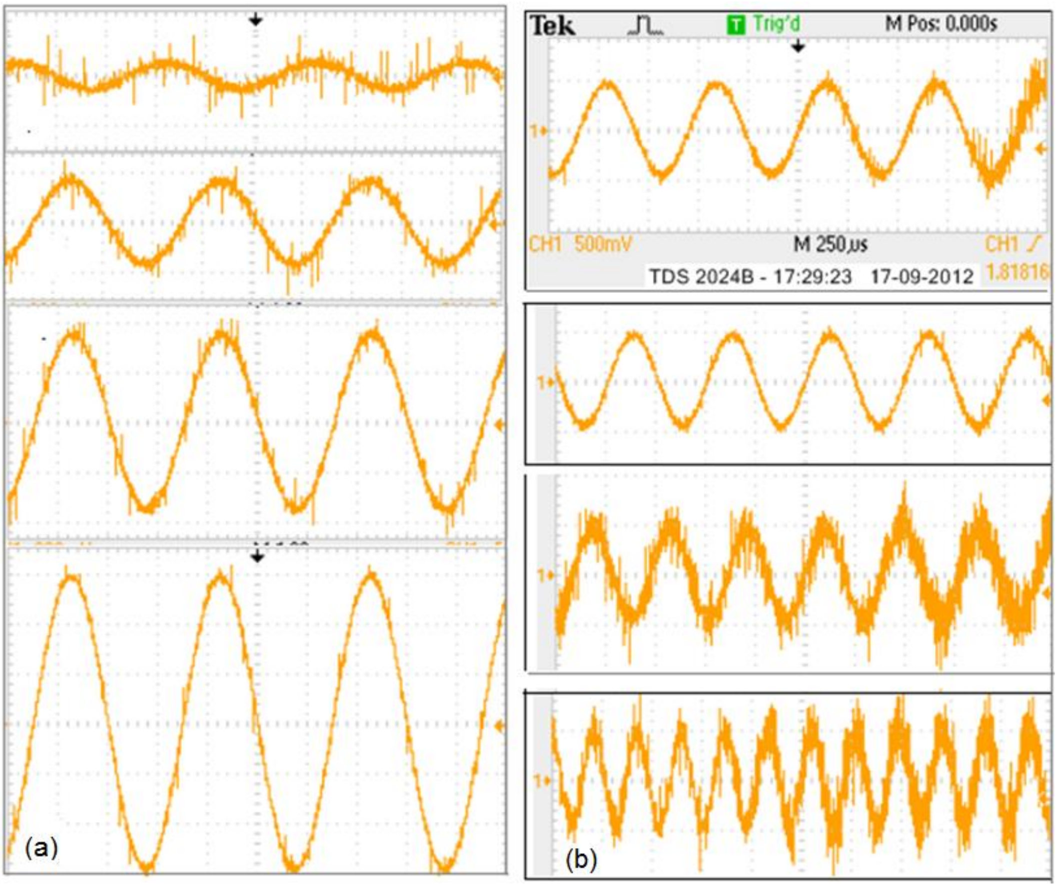



Figure 6.13 a; The amplitude modulated audio driver for communicating speed of approach, b; the tone modulated audio driver for communicating vicinity to operator.

## 6.5.3 Application of 'Phantom' for Wider Types of Tele-Control

The drag percept developed in chapter 5 is not effective for tele-control systems employing position exchange based BTM since the percept does not stand apart in the master position loop. By using the Percept parameter (PP) to simultaneously

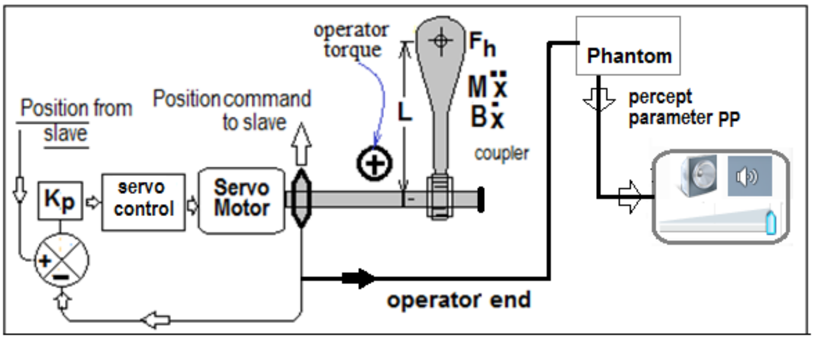



Figure 6.14: Use of auditory percept rendering for integration of phantom system with PE type of tele-control.

convey vicinity as in previous section allows the user to identify the drag by phantom as different from the position exchange based FFB in tele-control system. The scheme of 'Auditory Percept formation' is shown in figure 6.14

### 6.5.4 Role of Auditory Percept

The aforementioned applications of auditory percept strengthen the 'haptic percept developed by the MPC model using the sensory framework as detailed in Chapter 5.Its major advantage is its being non interfering in any way in the local master control loop and economical realization. It has further promise if implemented in elaborate form to develop spatial location perception on stereo and surround sound form. Doerr et al [93] have attempted to use 3D sound in a PC-based training system as a way to direct and guide trainees on aircraft cockpit simulators to observe specific events in 3D space. Auditory effects can positively affect the perception of sound events and their localization [92].

## 6.6 Visual Percept Development for Aiding MPC Function

**Forming a Distal Percept**

In performing a perceptual analysis of virtual sensory percepts, it is convenient to define the different stages of sensory perception [78]. A '*proximal stimulus'* is defined as the sensory information falling upon a receptor. A '*distal stimulus'* on the other hand is the distant source of such sensory information. Although we, as human beings, interact with an environment of distal stimuli, we only have direct access to proximal stimuli. Thus the act of perception is often described as the transduction of a proximal stimulus coupled with the judgment of what distal stimulus most likely caused that sensation. This act of inference is often called the '*perceptual hypothesis'* and results in the generation of an internal representation of the outside world known as a '*percept'*. As long as the perceptual system is presented with enough salient sensory information in a proximal stimulus, a correct perceptual hypothesis can be made and an appropriate internal model of the actual distal stimulus will be created. For example, a proximal stimulus might be an image of a cube falling upon one's retina. One's perceptual system extracts the salient information such as edges and angles from the proximal stimulus and, as a result, one infers that the distal stimulus is a cube located across the room. This proximal stimulus might be very different from the last time one viewed that cube—lighting conditions may have changed, viewing location may have changed, or it may even be a cube that had never been seen before. Nonetheless one identifies the object as a cube and builds an internal percept. Our ability to draw the appropriate perceptual hypothesis despite changes in viewing conditions is called *perceptual constancy* and is important in allowing us to generate a robust internal model of the outside world. Clearly, certain sensory information was critical for the identification and generation of the percept, while other information dependent upon viewing conditions may have been ignored. Because sensory perception is a complex process of inference based on certain features and not others, the key to designing a virtual percept is to ascertain which features are vital and which can be ignored.

Whether our visual system is presented with a photograph of a cube or a rough sketch of a cube, the image is likely to be identified as a cube and an appropriate internal perceptual model will be generated. In a sense, the photographic representation is analogous to physical modelling of the distal stimulus while the sketch representation is analogous to perceptual modelling of the proximal stimulus. Although a sketch contains much less sensory information than a photograph, a sketch artist is skilled at providing only the appropriate sensory features that assure the desired perceptual analysis of the image. A good sketch can often be a more effective representation of sensory information than a poor photograph. We can extend the analogy of the sketch and the photograph to more exotic perceptual representations such as the virtual haptic sensations produced by force reflecting systems. Rather than producing a physically accurate “photographic” representation of a haptic sensation, a ‘*perceptual designer’* could “sketch” haptic sensations by combining only those appropriate perceptual features which make up the desired percept. Such an approach may be more effective than “photographic” dynamic modelling of a haptic sensation, particularly in cases where force reflecting equipment lacks the fidelity to generate a completely realistic “photograph” of the stimulus. The MPC model has been motivated by this approach. In this work inspired by above observation very simple 3D graphic projection of the workspace model presentation for aiding distal perception has been developed by using untextured object model projection for the robot body part that do not employ texture mapping to achieve better speed and interactive context based presentation to convey the surrounding workspace near a virtual body sensor susceptible to hit.

## 6.6.1 Dedicated Sensing Techniques for Visual Percept Development

The development of visual percept needs more evolved sensing by use of cluster of sensors rather than a single sensor.

### 6.6.1a Sensing for Principal Gradient Detection

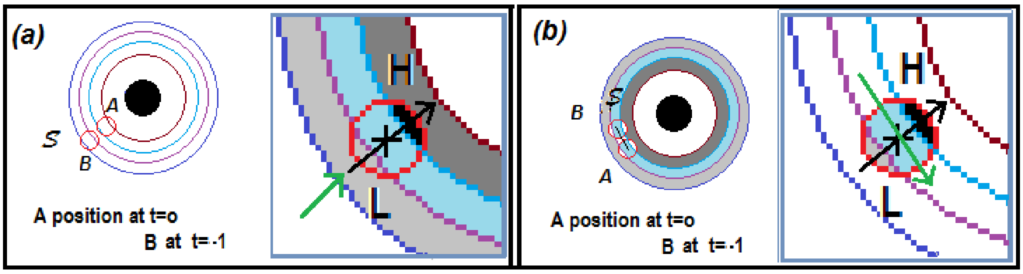



Figure 6.15: sensing maximum gradient, a; gradient direction along move, b; direction across the motion

A sensor moving towards an object sees increasing drag (figure 6.15a) but while moving along object wall and not closing in on any further (figure 6.15b) may see constant drag.

A clustered set of sensor around the point sensor succeeds in forming sensitivity with other attributes. We define two 3D arrays; max filter $MF_{(i,j,k)}$ and minimum filter $LF_{(i,j,k)}$of same size as $SSA_{(i,j,k)}$ for holding binary values 1/0. First shaped sensor zone data is buffered in $IBA$ by following method

**6.6.1b Formation of Principal Gradient Vector**

$$For\ (-m \leq\ i \leq m, -m \leq\ j \leq m, -m \leq\ k \leq m),$$

$$IBA_{(i,j,k)} = \mathrm{SLA}_{(\mathrm{i,j,k})} * SSA_{(i,j,k)}$$

Then, within the shaped sensor zone the maximum value of $\gamma$ say $H_\gamma$ is found by scanning $IBA$. Next, all voxels in IBA with values $H_\gamma$ are marked –

$$For\ (-m \leq\ i \leq m, -m \leq\ j \leq m, -m \leq\ k \leq m),$$

$$if\quad IBA_{(i,j,k)} = H_\gamma\ \ then\ MA_{(i,j,k)}\ = 1;\quad else\quad MA_{(i,j,k)}\ = 0$$

To find location $P_{H_\gamma}$ of $H_\gamma$ zone, method is –

$$For\ \ (-m \leq\ i \leq m, -m \leq\ j \leq m, -m \leq\ k \leq m),$$

$$if\ MA_{(i,j,k)} == 1, then\ \Sigma i\ = +i, \Sigma j\ = +j, \Sigma \mathrm{k} = +k,\ ct\ =\ +1$$

'$ct$' is count of $H_\gamma$ occurrence. Position of $P_{H_\gamma}$ is –

$$X.P_{H_\gamma}\ = \frac{\Sigma i}{ct}, Y.P_{H_\gamma}\ = \frac{\Sigma \mathrm{j}}{ct}, Z.P_{H_\gamma}\ = \frac{\Sigma \mathrm{k}}{ct}$$

Similarly we find minimum $\gamma$ value '$L_\gamma$' and form $LA_{(i,j,k)}$.By using $L_\gamma$ and $LA_{(i,j,k)}$, instead of $H_\gamma$ and $MA_{(i,j,k)}$ respectively, $X.P_{L_\gamma}$, $Y.P_{L_\gamma}$, $Z.P_{L_\gamma}$ are found. Vector $P_{L_\gamma}$ to $P_{H_\gamma}$, is 'principal gradient vector' **PGV.** Its direction is principal gradient direction '$PGD$' and principal gradient strength '$PGS$' is $|H_\gamma - L_\gamma|$.

## 6.6.2 Experimental Test on Principal Gradient Detection

The PGD sensor (figure 6.16) was moved in the vicinity of a cube object located between $Z = 40$ and $Z = 60$ in workspace with 6 layer neighbor coding using non linear $\gamma$ map. In top and bottom right figure, pointing of red and green vector toward body center appears as downward and upward tilt in them respectively. The blue vectors remain horizontal as their height corresponds to center of cube and hence show no tilt.It was moved on $z$ values 40, 60 in XZ plain located at $Y = 35$ (close and parallel to edges of cube) and also at $z = 50$ (close to wall). PGD behavior in 3D space can be visualized by forming Principal Gradient Vectors PGV. The sensor moving at $Z = 60$ (near top edge) generates PGV that tilt downward near object and

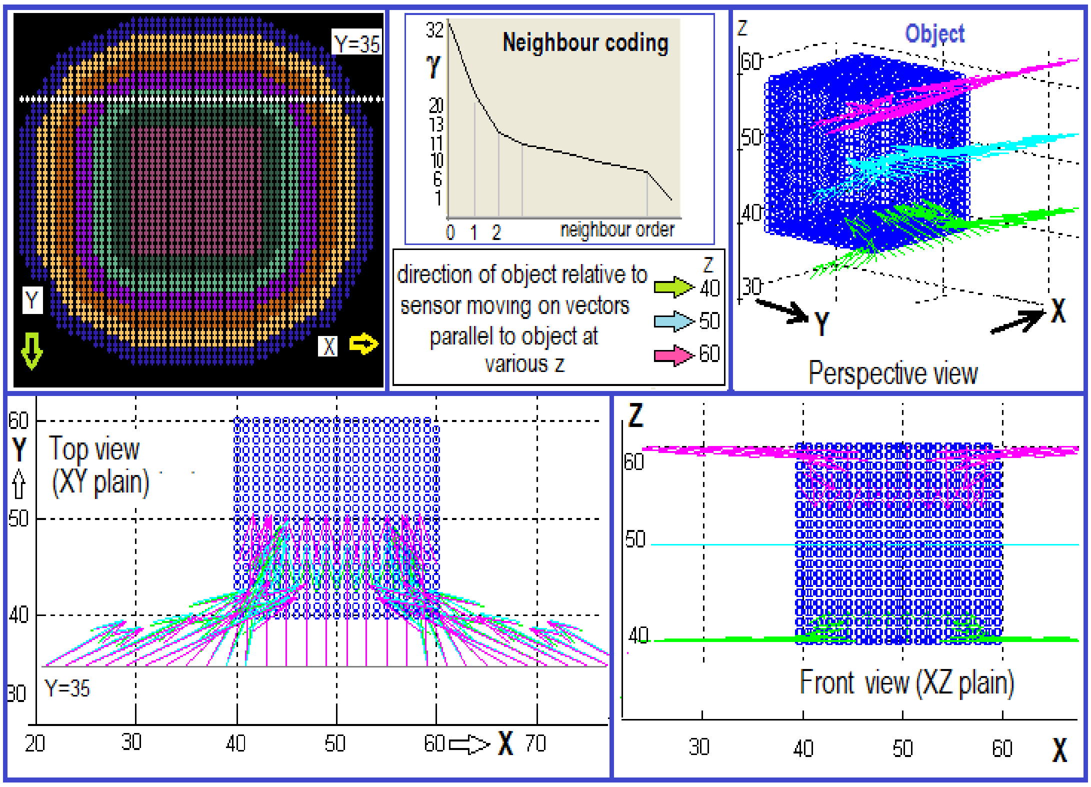



Figure 6.16: Principal gradient direction and strength sensing by a sensor passing through vicinity of a cube object. X,Y and Z are Cartesian coordinates in workspace.

it tilts upward when sensor moves on track at $Z = 20$ (near the lower edge) of object. PGVs indicate relative object direction and higher PGS (longer vector) shows closeness to senor.

## 6.6.3 PGD Sensor Tool for Innovative HMI on Operator Console

### 6.6.3a Effectiveness of PGD

The effectiveness of PGD has been tested in vicinity of a complex cylinder (figure 6.17) for visual appreciation by graphical display of **PGV** sensed by PGD detector.

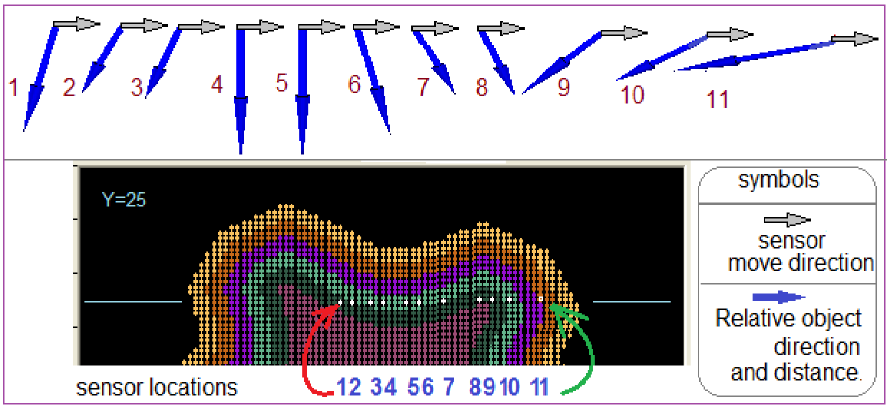



Figure 6.17: A sensor in neighborhood of complex cylinder produces PGD based imminent interaction coding in perceivable form for a motion initiation from standstill.

Grey vector shows sensor movement and blue vector shows **PGV** with length inversely related to PGS (blue vectors). At each location of sensor, the 3D displayed vector coveys object direction and nearness very effectively. PGD based sensing is established by response captured at sensor locations, 1 to 11 on $y = 35$ track (figure 6.17 top row).

### 6.6.3b Unsafe Motion Detection:

The motion vector and PGV vectors can be displayed in 3D perspective view as related entity for perceiving relative object position on graphic 'HMI' at high refresh-rate owing to computationally economical rendering.

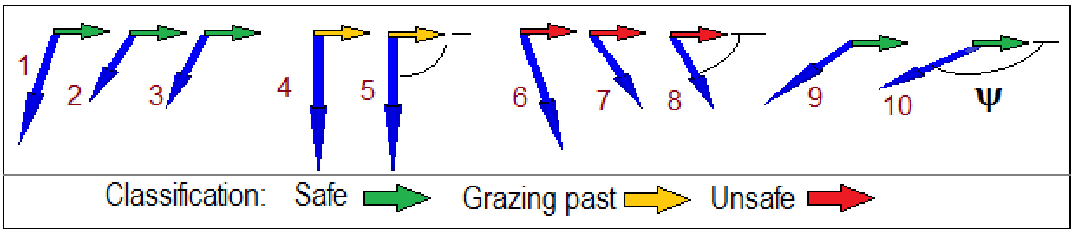



Figure 6.18 Classification of move initiation as safe, neutral and dangerous type by the unsafe motion detector UMD.

### 6.6.3c Relative Inclination of PGD to Motion Direction

The Inclination of PGD to motion direction is an effective basis for classifying just initiated motions after standstill in object vicinity as safe or unsafe. Angle Ψ, in plain containing motion vector and PGV, is used for imminent contact warning (figure.6.18). This is unsafe motion detection UMD function. It is used for the Impending Interaction Sensor 'IIS' in the following Section 6.9. The monotonically increasing gradient encoding of 'μ' has significant advantage; (i) it ensures quick rise of smoothened $F_r$ (ii) The magnitude of 'PGV' is higher nearer to object surface and serves for closeness detection at low speed or standstill more effectively

# 6.7 A Method for Perceiving the Contact Prone Scenario by Synthetic View

While approaching an object or navigating past one, perceiving the local scenario is essential in many cases. In tight workspace around concave bodies such analysis gain importance.

A 3D graphics renderer named Virtual Viewer System 'VVS' has been built earlier using *Microsoft Direct –X ver.10.* [102] has been used. It renders all the objects of workspace by treating them as '*in-situ'* CAD models [50]. It treats robot link objects through robot kinematics-model to form the suspended robot (figure 5.4) model. Its properties include 'camera position $X_c, Y_c, Z_c$' and 'look at point' $X_L, Y_L, Z_L$ in virtual space.

## 6.7.1 Contact Zone Assessment Using Enhanced MPC with PGD

Motivation to develop this tool comes from non viability of instrumentation for viewing instrumentation in cells as these cannot be taken close to radioactive parts [103]. In conditions reported by MPC as too close to object, often a need arises to view the impending contact for either manoeuvring the robot or tool-tip out of the zone or to plan an approach ahead to grasp or to engage with the object. Visualisation of the local scenario near contact generally can be done best along a plane passing through the current position and normal to the vector **PGV.** In experiments, a synthesised view (figure 6.19) with a cone directed along **PGV** and pointing to object is found useful .

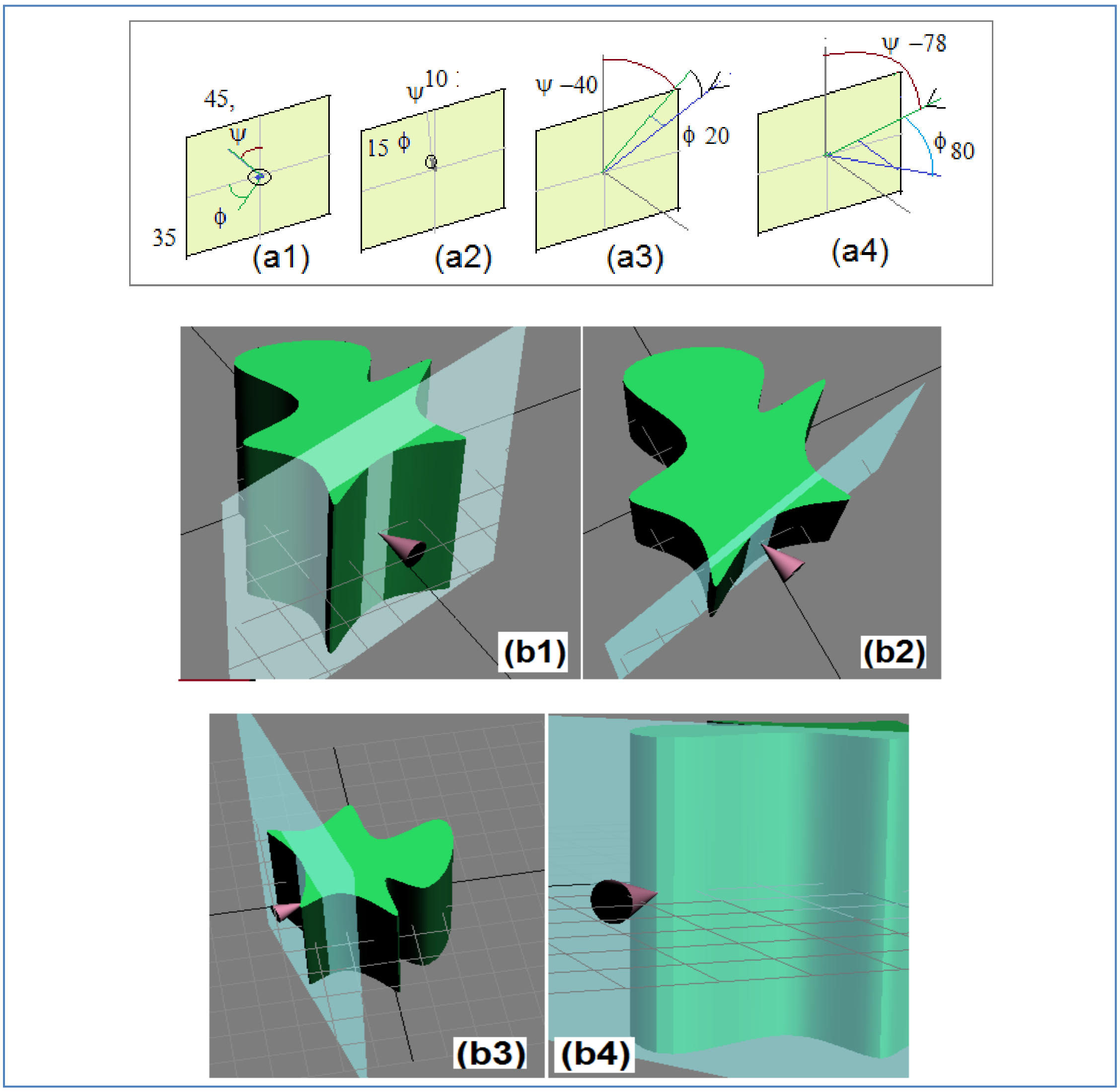



Figure 6.19: Visuals based perception enhancement: b1 to b3 by rendering visual artefact as aid while viewing them form positions (a1) to (a4) relative to the artefact .

This cone is a virtual object . A semitransparent plane for which the PGV is normal vector and located at cone tip is very effective aid for accentuating the presence of concavity more effectively. The viewing from positions close to this plane is very effective as it gives lateral view of the contact. The semitransparent plane is also very effective near convex surface as the plane is mostly tangential to the convex surface.

In figure 6.19, visuals based perception enhancement is shown in b1 to b3, by rendering visual artefact which is a semitransparent plane normal to PGD formed from MPC output while viewing them form positions (a1) to (a4) relative to the artefact . An approach to concave zone is easily perceived owing to the aid.

## 6.8 Compensating Lack of Peripheral Vision

Since the function of human peripheral vision is to detect movements and impending collisions based on the immediate scene itself we base our method on developing a distal percept using synthesized view that is presented to operator on when a '*susceptibility to hit*' is detected. Our detection method relies on the MPC percept generating parameters as elaborated in chapter 4.

The sensor objective is not to merely support a force transducer that responds according to the model as discussed in chapter 5 but to achieve maximization of pre-hit time to the impending hit for ensuring enough time for the operator to respond to such event. Since the visual percept in the form of 3D graphics rendered on operator console can be overwhelming if wide views are presented in fast changing manner, the suggested synthesized view aid must be oriented to view limited workspace zone of pertinence. For example if a tool nearing an object or a body part of robot nearing an object, while generating a drag based percept, also produces a distal percept by visual rout, it can be very effective in conveying the remote workspace scenario to operator. The near contact conditions can occur at different locations at different time, their detection is necessary for generating only pertinent view to the human operator of the telecontrol system.

A major limiting factor in constrained view telecontrol as indicated in Chapter 1 is the lack of peripheral vision that an operator of an interactive system is otherwise used to . For large robotic slaves in work-cells this shortcoming is a major concern. The aforesaid hit susceptibility sensing, when implemented on sensors at periphery of the slave arm , can mitigate the difficulties arising from lack of peripheral vision, by view synthesis to a good extent.

The sensor should also aid the function of graphic pipeline with parameters like 'observer location' or 'look from' and 'look at point in the pertinent zone of workspace'. In addition the sensor must support a graphics system in forming best view for appreciating near contact condition. The percept parameter PP formed by the phantom system in Chapter 5 has the property of increased sensitivity based on nearness as well as speed of move. Susceptibility to hit has similar dependence on nearness and speed with a difference that speed needs to be treated as vector rather than scalar quantity in some conditions .We form following rules.

## 6.8.1 A Rule Set for Contact Assessment

We form rules as following -

Rule 1. For sensor location very close to a body surface the hit susceptibility should be high irrespective of robot motion as slight change in position at low speed too can cause hit.

Rule 2. For sensor location not too close to a body surface, the hit susceptibility should be high for any movement above a specified threshold irrespective of robot movement direction as change of direction can cause hit in concave zones around object body .

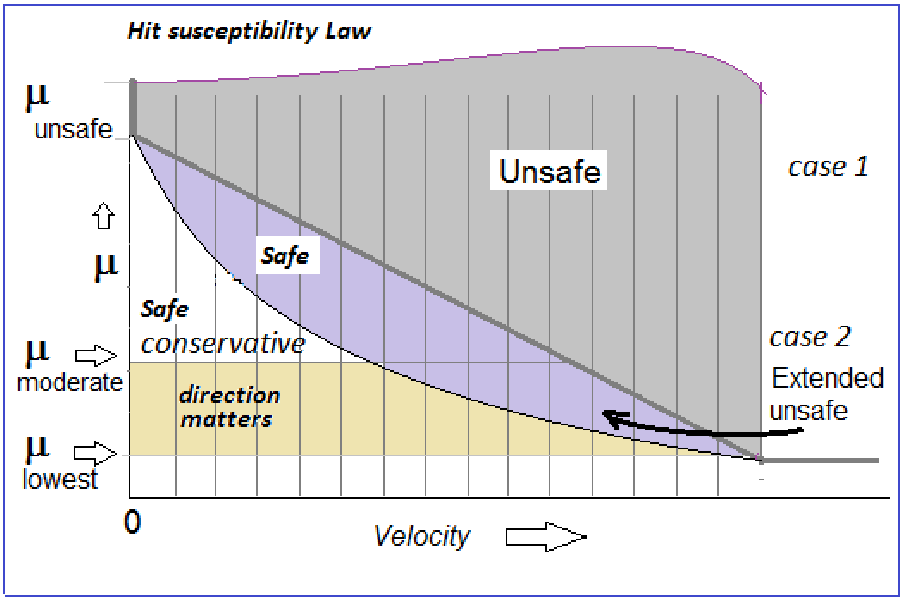



Figure 6.20: The hit susceptibility rule set is formed on the basis of parameters shown here. Mark the safe and unsafe zones of normal and conservative types.

Rule 3. For sensor location moderately close to a body surface, the hit susceptibility depends on robot movement direction. Only move towards object surface should cause trigger. (figure 6.20)

Therefore

a; The direction of the object from the sensor needs to be determined

b; The sensor motion vector should be determined

c; Criteria of ' towards object surface' should be established.

The rule set is implemented as below. The safe zone is lower side of a linear boundary or a nonlinear (conservatively safe) boundary. The law is coded either as equation or a look-up table with velocities recorded for each layer number from which interpolated $\mu$ values are accessed and compared with the levels of '$\mu$' online. *LUT-zone* can be safe or unsafe. At any time instant 't' location of the sensor,

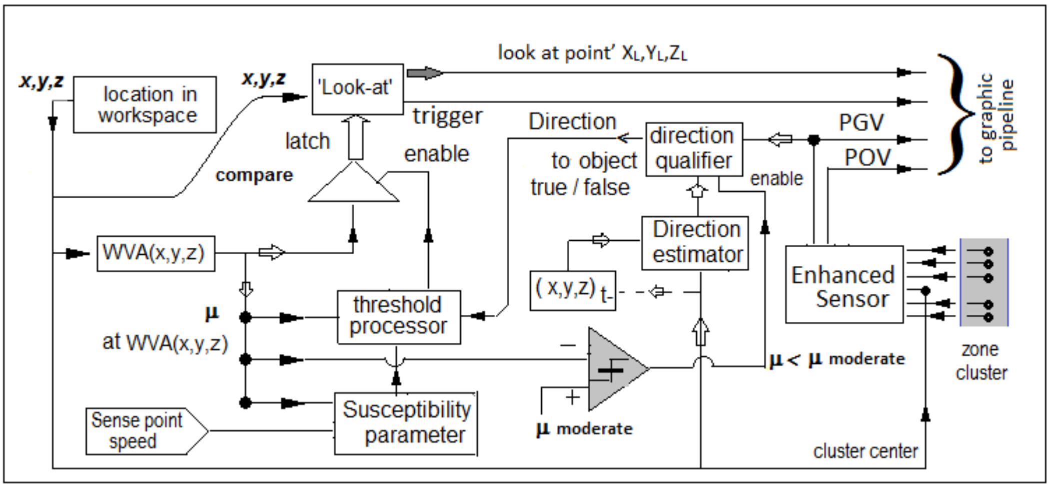



Figure 6.21: Impending Interaction Sensing Function and its interface with cluster sensors and graphic pipeline for Visual Assisted Distal Percept formation.

determined from joint parameters, when binned as per the workspace world coordinate *vs*. WVA scale, gives indices to the WVA (figure 6.21). The accessed $\mu$ value, velocity and relative direction to PGV i.e. object approach direction quantified by cosine of the relative direction angle and referred as 'Value' are used to draw decision as outlined below. Threshold processor implements this rule. The trigger registers latch and present sensor location as 'look at point' when conditions of 'hit-susceptibility' is satisfied as per logic outlined below.

Method: Hit Susceptibility Assessment

$If \quad \mu.WVA(x,y,z) > \mu\_unsafe \quad SET\ enable\ TRUE$

$If \quad \mu\_unsafe < \mu.WVA(x,y,z) < \mu\_moderate$

$If \quad LUT\ zone\ unsafe \quad SET\ enable\ TRUE$

$If \quad \mu.WVA(x,y,z) > \mu\_moderate \quad AND \quad LUT\ zone\ unsafe$

$$AND\ \frac{PGV * MOV}{||PGV|| * ||MOV||} > Value$$

$SET\ enable\ TRUE$

## 6.8.2 Performance of the Developed Technique

The 3D graphics module referred earlier has been used here too. A test set up- with 6 DOF articulated robot hanging from wall mount. For the experiments 'the look from' point is formed as shown in figure 6.22 and two objects the blue vertical plate and pink horizontal plate at back of arm is modelled in phantom (figure 6.23). Voxelized workspace WVA is formed using 6 layers and μ 32, 25, 20, 17, 15, 13 for $n = 1\ to\ 6$ respectively. Sensor is attached at the opposite side of arm on the joint 3 motor body.

The "look from" vector generation module has been formed to compute camera position for viewing the contact location in WVA point '$O$'. For this the 'look at point' registered by the hit susceptibility assessor is used as the target point . To form view vector , the algorithm computes nodes on face centres of a cube which has its centre at the look at point '$O$' here and axis parallel to WVA coordinate system . Vectors from any of the nodes E,W,S N, T, B , 1,2,3,4,5,6,7 and 8 can be formed as per selection ' rule select ' in figure 6.22 and camera is placed along chosen vector at distance seeing the point '$O$' .

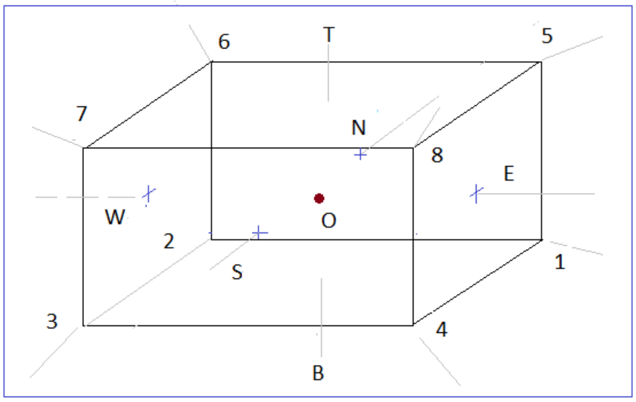



Figure 6.22: Forming look from locations for given a 'look at' point generated by the phantom as susceptible hit zone centre.

The above described function resides in block IIIB in figure 6.1. The subsystem IIIB, responsible for graphic rendering, maintains graphic renderable models of workspace with the constituent objects. It may be noted that this is the source model from which voxel models are formed and used in block IIIA of figure 6.1. The results of the experiments are shown in figure 6.23. The VDAP sensor was placed on the dark gray motor end of the arm.

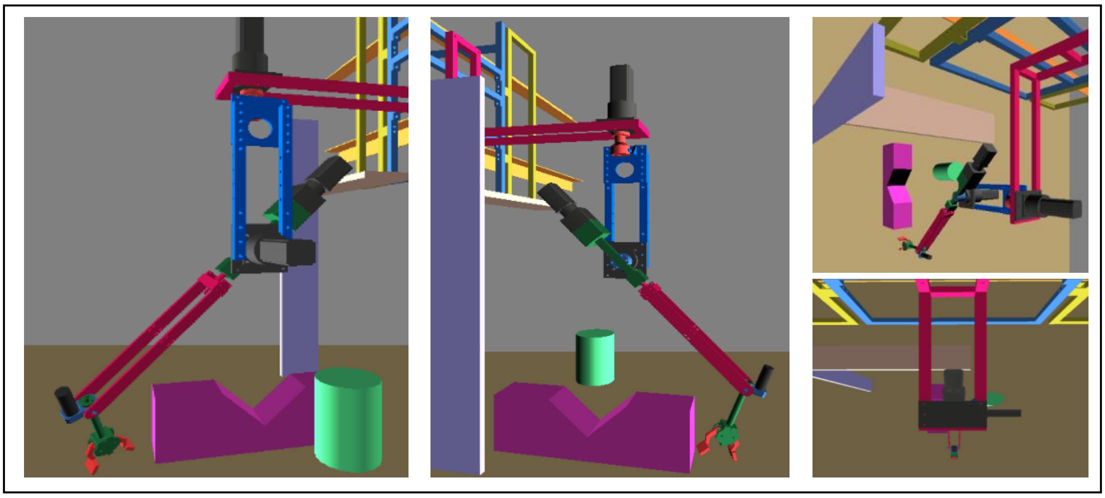


Figure 6.23: Experimental result : view of near hit situation from direction 4, 3 , 7 and T in left , middle , right top and right bottom respectively

## 6.9 Function of Visuals Assisted Distal Percept

The 'Visuals Assisted Distal Percept' VADP rendering is achieved by the block III B in newly developed phantom using IIS, graphic pipeline and the view parameter formation rules. It aids the operator in perceiving remote environment more effectively by supporting MPC function.

Notable Features of the enhanced Sensor Framework in Phantom 3 developed here.

**1)** *Versatility of PGD as Sensor:*

*a) Effectiveness of PGD:* has been also tested in vicinity of a complex cylinder (figure 6.17) for visual appreciation by graphical display of **PGV** sensed by PGD detector.

*b) The motion vector and PGV vectors:* can be displayed in 3D perspective view as related entity for perceiving relative object position on graphic 'HMI' at high refresh-rate owing to economical rendering.

*c) Inclination of PGD to motion direction:* is an effective basis for classifying just initiated motions after standstill in object vicinity as safe or unsafe.

**2)** *Efficacy of Smoothening Methods:* The smoothening using cluster sensor approach produces smoother Fp without delay that is prevalent in time serial buffer based method.

**3)** *Gains From Nonlinear γ Coding:* The monotonically increasing gradient encoding of '$\gamma$ ' has significant advantage;

(i) It ensures quick rise of smoothened $F_p$ (figure 6.8),

(ii) The 'PGS' is higher nearer to object surface and serves for closeness detection at low speed or standstill more effectively.

**4)** The 'impending hit' criteria formation based on the basic HVP percept itself proves its effectiveness for peripheral view synthesis for robot body parts away from the end-effector which generally is under view in limited vision conditions.

# 6.10 Error Scenarios and remedial measures:

An object may be sensed and modeled as smaller than the size or larger than its real size as shown in figure 6.24 . In general error is mixed positive and negative as implied by case 3 , and so for same object, both conditions can exist in different parts. Here it is assumed that the voxel coding is only for free , boundary and neighbor layers. In red zones, the Phantom system believes that object is present but actually object is not there, and so no PFFB is developed. Also SCTFB is not produced as the physical arm is acting in synchronism with master and has not yet developed contact with object in workspace. As a consequence, in this zone, neither PFFB nor SCTFB reaches the operator's hand. This can lead to operator perceiving the zone as free space

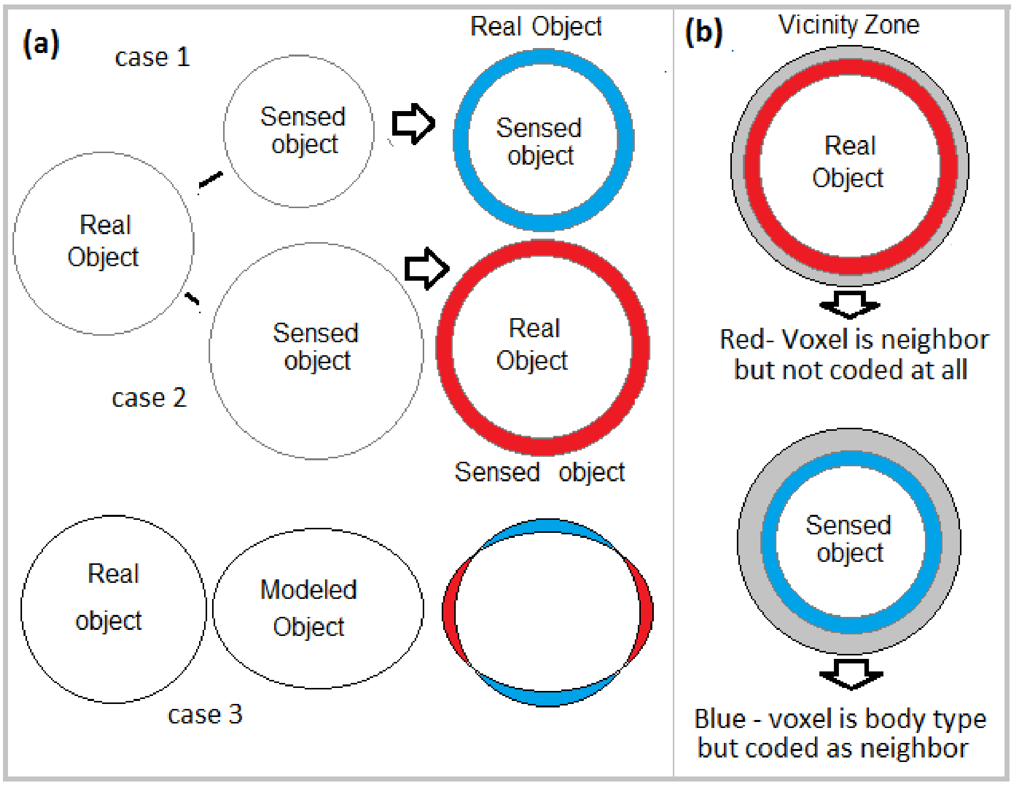



Figure 6.24: a; modeling error cases, b; error effect on coding

In blue zones the Phantom system believes object is not present but actually object is there. The phantom robot is in the neghbour zone either closest to object or at next lower order neighbor layer but SCTFB has initiated owing to object contact. As a consequence hit (though at low velocity) can happen for the real slave and SCTFB and PFFB both are produced. But PFFB effect is lesser than expected. Case 2 in effect reduces the closest layer thickness. A wider 'layer of order zero' is an effective solution as it works as guard band.

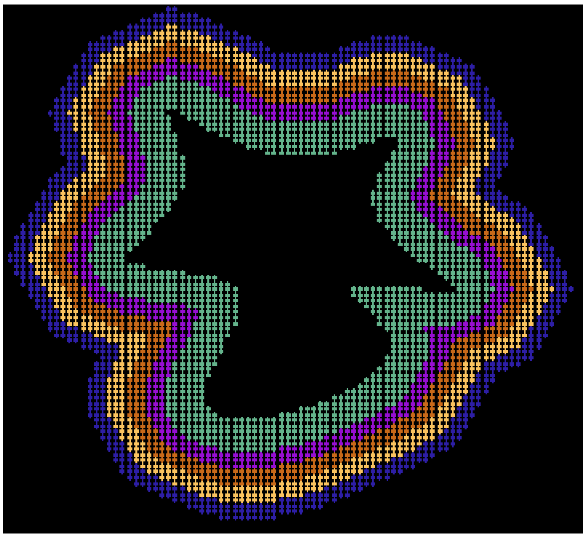


Figure 6.25: Closest vicinity (green) is wider with same μ

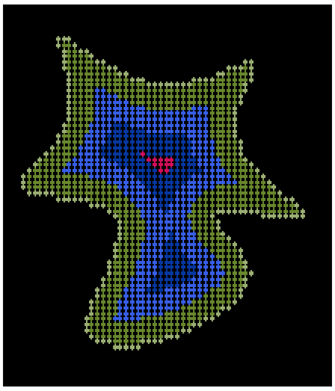


Figure 6.26: Layers inside object with 'μ' values.

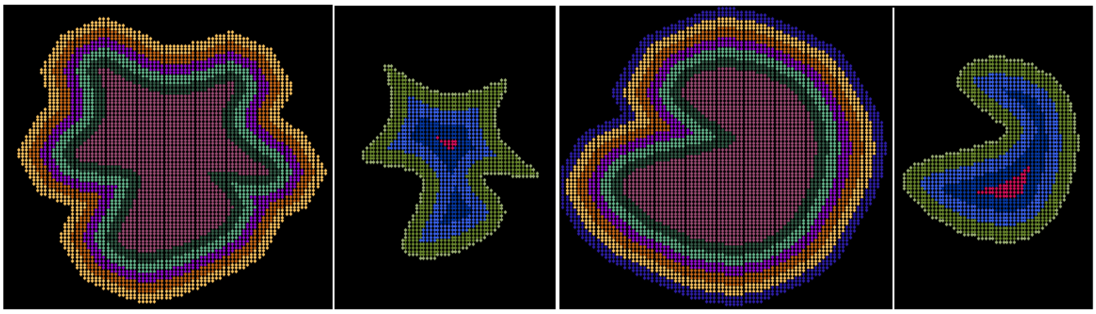


Figure 6.27: Objects coded for outer layers are coded inside by three layers.

In figure 6.26 the layers that are actually inside object are also given ‘μ’ values so that in case of object being smaller than model the MPC gives feedback without discontinuity till actual slave feedback appears. The coding inside object is as shown in figure 6.27. The core zones representing object body in two objects are converted to layered zones. This can use same haptic rendering channel as used by the phantom in chapter 5.

## 6.11 Phantom Frame Work Evolution and Hosting Multimodal Percept

The forgoing research has lead to development of a phantom framework-III (figure 6.28) that works as a comprehensive model based mediator and achieves multimodal perception enhancement by using a versatile percept capable of supporting multimodal perception that has enhanced effect from synchronized incidence on haptics, auditory and visual human sensing faculties). The novel modeling methods, evolved here, use complementing function like virtual activation of workspace and sensor equipping of robot,

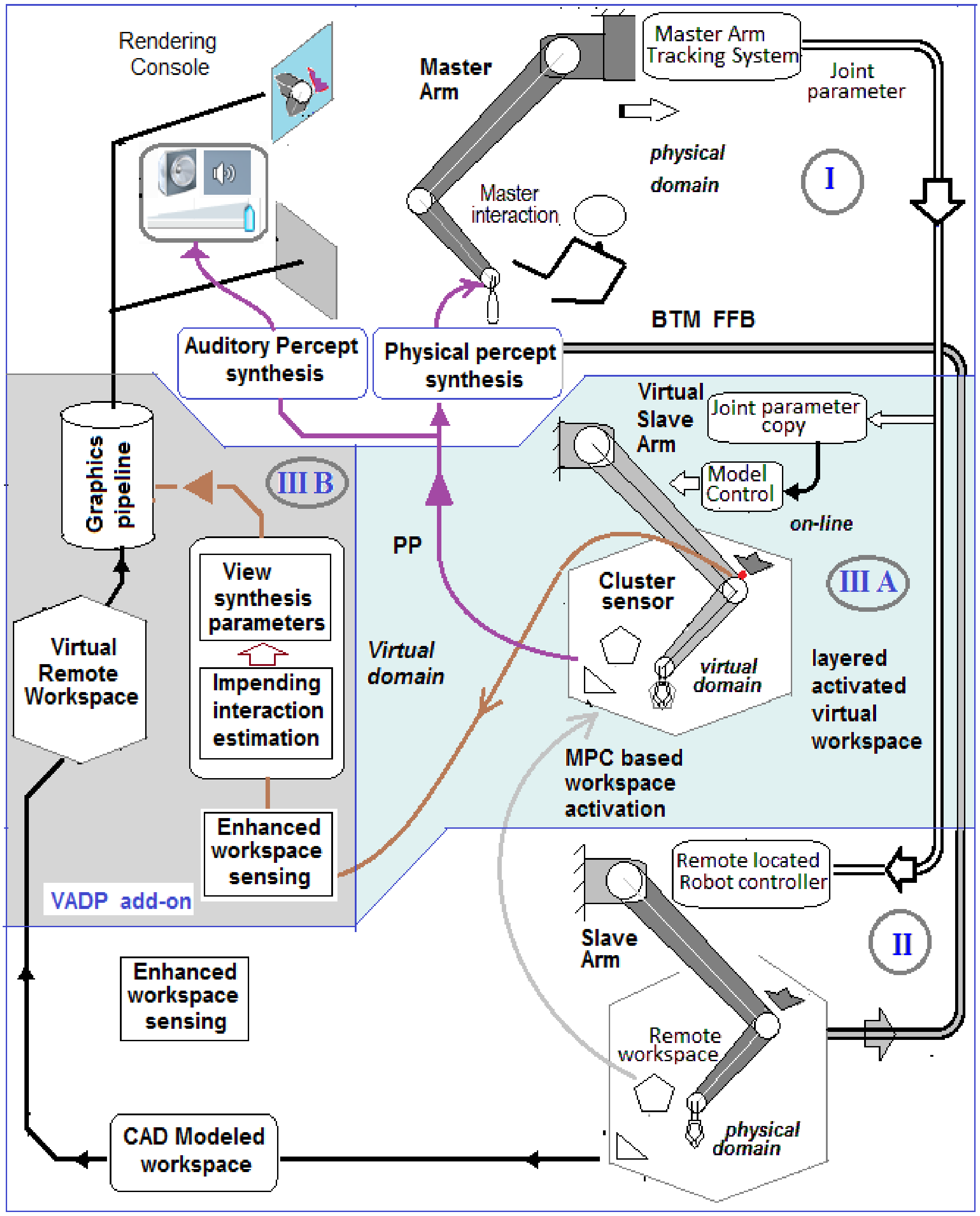



Figure 6.28: Elaborate view of Phantom III framework.

The emulated interaction synthesizes accurate haptics effect suited for creating a nearing feel in close vicinity of object, and forming a feel of presence in telecontrol systems.

**Table-6.1:** Comparison of MPC role in percept formation for different tele-control scheme

| | | Phantom-I | Phantom II | | Phantom III | | |
|---|---|---|---|---|---|---|---|
| | **Telecontrol Model** | **' Haptic' HAVP** | **'Auditory' DFAP** | **HAVP +DFAP** | **Synthesized View- VADP** | **HAVP + VADP** | **HAVP +DFAP +VADP** |
| 1 | Case (a) | - Only available haptics feedback.<br>- Useful to control master | - Effective aid for risky moves.<br>- Not essential | - Not essential.<br>- Permits, wait and act.<br>- Precursor to contact in very slow motion. | - Only visual perception method in the scheme.<br>- May trigger very close to object in slow moves. | - More like VR but lacks contact phase haptic feedback.<br>- Effective multimodal percept , permits wait & act. | - Good presence.<br>- Compatible to wait and act<br>- Improved VR |
| 2 | Case (b1) PFFB(I) Balanced | - effective | - Improves performance | - Gives good presence.<br>- Permits wait and act. | - Effective as peripheral vision substitute<br>- improve HAVP | - Good presence.<br>- Effective multimodal percept but not oriented to wait & act. | - Good presence.<br>- Compatible to wait and act<br>-Best performance as minimal sensor<br>-Augmented reality |
| 3 | Case (b2) PFFB(I) Unbalanced | - Needs DFAP support | - Essential | - Permits wait and act.<br>- Essential for effectiveness | - Effective as peripheral vision<br>- Improves HAVP | - Good presence.<br>- Effective multimodal percept but not oriented to wait and act. | - Good presence.<br>- Effective multimodal percept<br>- Compatible to wait and act<br>-can improve with electronic counter balancing |
| 4 | Case (c) PFFB(II) Sensor | - Effective | - Improves performance HAVP | - Good for wait and act | - Effective as peripheral vision substitute,<br>- improves HAVP | - Good presence.<br>- Effective multimodal percept but no wait and act. | - Good presence synthesized<br>- Connected to slave<br>-Best performance as augmented reality but uses sensor |
| 5 | Case (d) PE (d1) | - Does not stand out | - Essential | - | - Effective as peripheral vision substitute<br>- Connected to Slave | - | - Good presence VADP+DFAP (No HAVP)<br>- Connected to slave |
| 6 | Case ( d) PE (d2) Sensor | - Does not stand out | - Essential | - | - Effective as peripheral vision substitute<br>- Connected to Slave | - | Good presence VADP+DFAP (No HAVP)<br>- Connected to slave |

**Notes:-** 1. HAVP: Haptic Assisted Vicinity Percept, DFAP : Dual Function Audio Percept, VADP : Visual Assisted Distal Percept . 2.Since the VADP function does not use texture map and the magnification of view depends on $X_c$ $Y_c$ $Z_c$ (Camera location) It does not convey closeness unambiguously and needs a co-working DFAP. 3. For 'cases' in column 2 refer 'telecontrol mode' in table 5.2.

## 6.11.1 Observations on Phantom Frame Work Evolution:

The table 6.1 summarizes performance of the phantoms I to III (figure 5.28) across columns in order of their evolution from left to right. For variety of BTMs (refer to Chapter 5 table 5.1) that appear in rows. For each type of phantom its intrinsic functional performance appears on left most column, while it's combinatorial function with earlier evolved percept appear on the right columns. For example, for the phantom II that offers 'VADP' as added percept, it appears on left most (VADP) column while its combined versions with HAVP and DFAP appear on the right.

**case (a) of table 5.1**

For this case since it t uses only position forward control and has no SFFB, the HAVP adds a vicinity percept for high speed approach only while the DFAP gives approaching feel to the operator for low speed approach as well as stand still condition too. It is marked 'not essential' as otherwise HAVP is available. Note that for this type of control the 'MPC' generated feedback, PFFB is the only haptic feedback. The evolving phantoms across versions II and III succeed in providing a 'virtual-reality' VR type environment but sans contact phase SFFB. The notable factor is that it works with least on body sensor on slave. Its drawback is that in case of model error its function lacks robustness as untracked contact forces can build and damage the hardware.

**Case (b1) of table 5.1**

The BTM is position forward PF type with the arms of balanced nature. The SCTFB has minimal self loading effects other than friction loads. The HAVP is very effective by itself in creating perception of an approaching object through proximal haptic percept from the MPC. The zero speed situations need DFAP coincidence for keeping the operator aware of proximity.

The DAVP contribution is in permitting a wait-and-act type operation by operator which may be essential for preparing oneself to act towards forming impact free stable contact. The final HAVP, DFAP and VADP is the most elaborate virtual reality experience and so the best 'presence' feel occurs here without any on-body sensor. The phantom co-working with the BTM offers free-space phase, pre-contact phase as well as contact phase feel of the environment. A notable aspect is that the modality offers tolerance to some model error as in-spite of less than actual proximity feel that might appear in pre-contact phase, the contact forces are felt by the operator. Since precursor feel to contact is available, the operator is aware of impending contact qualitatively. However the model error contribution in VADP may have undesired confusing impending direction of contact input to operator. If modelling error is high Case b2- The evolution of 'perception' versatility is same as that in case b1 with the difference that while DFAP and VADP effectiveness is though same as that in (b1), their contribution is highly essential since the HAVP is less effective owing to background force variation from gravity effect. Control of SCTFB gain (attenuation) can improve the situation.

**Case (c) Table 5.1 –**

The HAVP being a PFFB can easily work alongside the feedback that is generated by sensor. It has all the positive factors of the case b1 and gives best "mixed reality" experience with phantom III as the HAVP, DFAP and DAVP create relatively more modeling error tolerant function. But a trained operator can use contact exploration by contact search by force sensor feedback. Even model correction approaches can be adopted. The only drawback is the use of extra on-body sensor. But this sensor can give tremendous help in model correction by interactive route for the part pose correction to the existing true state in the workspace. In other application like telesurgery this modality may emerge as superior solution.

**Case (d1) and Case (d2):**

In these cases the bilateral control is primarily by position exchange 'PE' method. The slave robots of large size, being relatively heavier, cause a drag on the master side. The master, in such setups, face increasing drag with increasing speed of operation. This is not very different from HAVP except that HAVP occurs near parts. Therefore the HAVP does not stand out as a well recognized percept in this BTM modality.

The DFAP produced by Phantom II has high attention drawing capability and is an effective percept conveying vicinity and also rate of approach. The phantom III function with DFAP and VADP, deliver good feel of presence by forming virtual reality like environment. In contact phase the 'PE' function develops high opposing impedance in local master loop as slave is constrained to stay at contact position. The phantom III in this case is very effective upgrade to the 'PE' type BTM except for the HAVP.

The peripheral vision offered through synthesized view of hit prone zone is quite a help to operator irrespective of the BTM modality.

## 6.11.2 Salient Capabilities of Phantom III

The salient capabilities of the phantom framework is as following:

1. Unlike the haptic model based on constraint of interpenetration and effort of restitution, there is no imposition of force on body on rest in MPC model and so a balanced robot body can rest without dissipating energy close to the object body.
2. Unlike the other haptic model using contact forces, the determination of force direction is straight forward in MPC model as it is always opposite to the motion vector.
3. The MPC model induces a 'crisp' force reaction at layer interface which is more easily sensed than gradual increase. In addition to force modulation by scale, the model permits frequency (step occurrence interval) modulation by layer width variation which is less prone to force background at a given point.
4. The MPC model is multipoint
   a. Haptics reaction sensing forms by wading through the virtual fluid layer, a modality where a multipoint contact and reaction aggregation is natural unlike rigid virtual objects where a point contact only is realistic.
   b. The probe dynamics is well defined by the robot Jacobian [40].
5. The percept's speed and vicinity sensitivity can be directly rendered in auditory mode also very effectively by using tone & volume attributes of audio sounds. The MPC model induced percept achieves high attention drawing property.
6. Susceptibility to hit 'STH' parameter formed with the MPC model is sufficient for impending hit detection based peripheral view synthesis in visual domain.

7. The PGD based relative object direction at a location works on 3D spatial DSP (convolution) method and does not use computation intensive computation geometry based computations.
8. The MPC model works through generic virtual sensing framework created in phantoms and can be automatically formed by
   a. Neighborhood that is coded simply form STL model of parts of any complexity.
   b. Sensor structured on generalized cylinder model of link elements of the robot.
9. The method is highly scalable as
   a. Activated work space that needs large memory can be partitioned and can be handled by memory management techniques.
   b. The method is amenable to parallel processing. The granularity is defined by link segments which can be formed easily.
   c. Thin and tiny parts like fingers can have more simple sensor structures.

# Closure:

The phantom sensor hosted method of force-feel development, that appears only in vicinity of object has been enhanced substantially by devising array based workspace sensing over the Phantom model II. The enhanced sensor offers method for perceiving object presence in robot vicinity and relative surface location that are well suited for graphics based on-line appreciation on effective HMI and can be depended upon for audio effect synthesis wherein audio signal’s tone and volume may be used to convey imminent contact information and form effective human perception.

The multimode sensing, developed here, undergo 'fusion' in operator brain and helps achieve superior perception of remote slave scenario in workspace. The rendering is multimodal and real-time executable as uncertain search and boundless computation have been eliminated totally.

The automation friendly framework's use in delicate tele-control for precision remote operations such as machine aided surgery with multimode operator end interface and autonomous close range viewing from most appropriate direction facilitating precise observation of tool-object interaction can also be implemented using PGD technique.

The audio percept is essential to keep the MPC based percept detectable at low velocity near objects and in case of working with BTM with unbalanced slave ends. For the PE type bilateral telecontrol the auditory version assists the operator in identifying vicinity. The VADP percept succeeds in mitigating the lack of peripheral view in cell scenarios as mentioned in chapter 1. It has high dominance among all.

Precision remote operations such as machine aided surgery and autonomous close range viewing from most appropriate direction facilitating precise observation of tool-object interaction can also be implemented using PGD.

# CHAPTER 7

# CONCLUSION, CONTRIBUTION

# AND

# FUTURE WORK

# 7.1. Conclusion

Salient conclusions drawn from the forgoing presentation are as following :-

1. In this work innovative method has been developed to assist the human operator in perceiving the pre-contact phase by haptics based effect formation. The indigenous $\mu$ coded vicinity model, allows easy conditioning of dynamics based response to operator using multimodal rendering. It has achieved enhanced perception of approaching a workspace object. Casting of the model in two parts, the scalar 'basis drag' and a phantom sensor framework resulted in low computation load as no separate computation of reaction direction is required. The innovative approach permits synthesized haptics to function together with a physical slave without necessitating prediction of future state of slave robot arm.

**2**. The MPC behaviour as a 'proximal percept' is very successful in forming remote perception of presence of object in pre-contact phase for BTMs. The operator gets an improved feel of presence of remote work environment.

**3**. It is free from force artefact and so is suited well for 'stop-watch-move' operations in manipulation phase. The supportive auditory percept DFAP, being related to proximity, continues to convey the presence of nearby object in a qualitative manner.

**4**.The technique does not take away control from the operator, as by reducing speed, he can establish contact with a part. This has a very positive implication. Free space use in constrained work environment can be maximised by operator as some part of slave arm can be in near contact condition.

**5**. The phantom sensor framework, is tolerant to measurement error inherent in the non in-cell, long standoff type laser range sensing based approach of CAD model assembly into a workspace model [Appendix A].

**6**. The augmented phantom sensor employing sensor array, supports move classification as safe or otherwise. The ‘body sensitisation’ is versatile as it achieves sensing on robot shell of large size as well as cluster based sensitization of thin parts like fingers.

**7**. Optimization features of the ‘Phantom Framework’ include variable spacing of sensors which are anchored to the link segments. The structuring of the sensor at segment granularity and their scalability by complex shell, as well as simplified fingers, make the method amenable to scale-up. The method being data parallel type, data size escalation arising from small voxel high resolution modeling may be easily addressed by employing multi-processor approach in which individual object or their cluster and one copy of phantom form independent process execution channel.

On the other hand, complex arm shapes can be handled by treating the segments of link as parallel executable asynchronous modules whose computed results are integrated by synchronous method in a resultant joint torque aggregating process.

**8.** The single generic sensing scheme, innovated in the work, uses cluster sensors for fast real time sensing to facilitate synthesis of several types of on-line perception in haptic auditory and visual domains when in close vicinity of object. The utility of ‘principal gradient sensing’ by sensor array has been established in laboratory test set-up for detection of object location *viz a viz* robot position and distal percept synthesis in the form of synthesized view.

The multimode sensing, developed here, renders several haptic, auditory and graphic percept which undergo fusion in operator’s brain and helps achieve superior perception of remote slave environment in workspace. The method is amenable to automation for deployment in cells. It supports operator in making impact minimised contact or avoid contact altogether without taking away control from him.

The research has lead to development of generic method for mitigating limitations in teleworking that arise from lack of direct and peripheral vision in hot cells.

# 7.2 Dissertation Contributions:

An innovative approach has been developed for enhancing feel of presence by devising a fluidics inspired simple but adequate model that avoids error proneness of reaction direction estimation in haptic interaction model.

The development has also formulated a compatible and scalable generic virtual sensor frame-wok for reaction sensing as well as human interpretable multimodal, proximal percept casting. It enhances sense of presence using above model for wide range of telecontrolled work of delicate as well as industrial nature, such as BTM for hot cells.

## 7.3 Suggested Future Work:

The Phantom framework can be used to improve several types of telecontrol applications. Following is intended to be taken up in recent future.

**(1).** Case studies of the phantom framework on SCARA type manipulator pair for radioactive fuel assembly in hot cell are suggested. Most of the joint axes being vertical and last prismatic stage being balanced or light weight, SCTFB background is expected to be minimal and vicinity perception is expected to improve. An experimental verification on SCARA arms shall be done.

**(2).** The phantom framework shall be applied on a telesurgery set up by tiny voxels (<1mm) based workspace formation to explore its applications for patient safety augmentation against inadvertent fast approaches to body, sensitive tissues or sensitive physiological zones. The viability of the suggested investigation rests on following strength in optimizing the sensing process in the Phantom framework.

i). The method is highly scalable since

a. Activated work space, that needs large memory, can be partitioned and can be handled by memory management techniques.

b. The method is amenable to parallel processing. The granularity for parallel processing is defined by link segments which can be formed easily.

c. Thin and tiny parts like fingers can have more simple sensor structures

ii). Modularisation for parallel computing can be done as indicated in figure 7.1. Long segment is broken down to smaller segments with different segment to joint offsets ($SOJ1_a$ and $SOJ1_b$) and top and bottom polygon definitions repeated in them. They form independently executable modules in parallel computing system. With a different set of top and bottom polygon definitions the spatial resolution too can be independently defined along the same link.

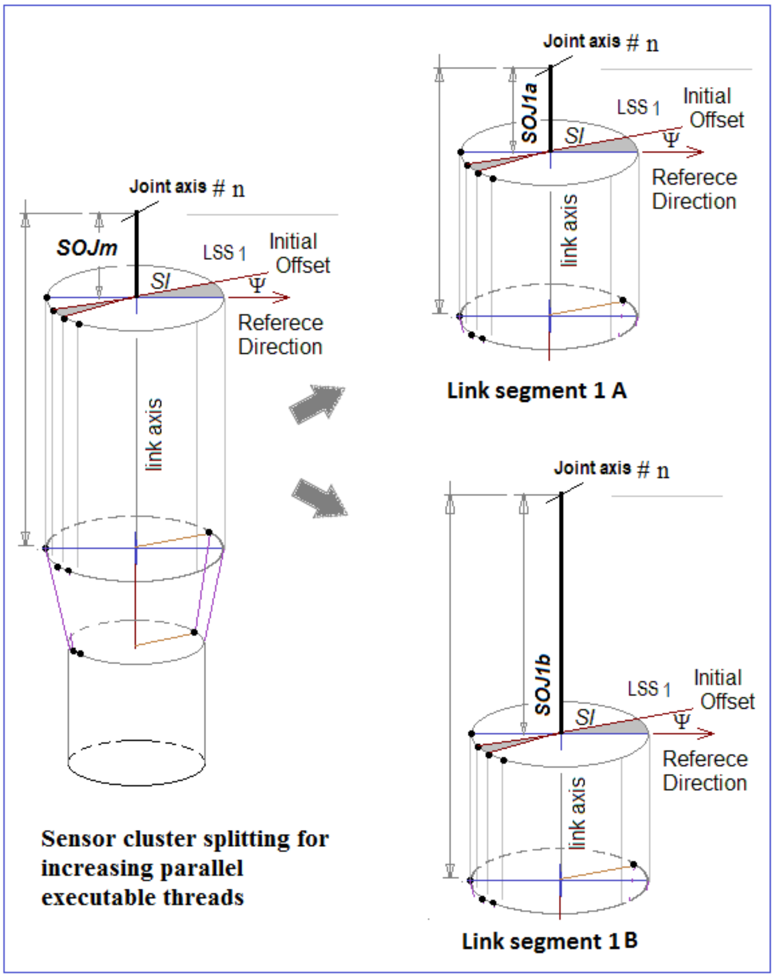



Figure 7.1: Modularisation for parallel computing - long segment is broken down to smaller segments.

iii). A multiple resolution implementation can be built with smaller voxels and fine pitched sensor meshes for different objects held in individual OVAs. While larger upper links use coarse model, the fingers can use finer model. These can be run in parallel in different threads or processes

iv). The voxel record is amenable to being coded in unsigned single integer (32bit) by assigning 1 byte to 'voxel type', 1 byte to voxel $\mu$ and 2 bytes to object index assuming a large object population in workspace. This array aligned data slightly increases sensed attribute access in runtime but reduces memory requirement effectively and permits very fine voxel definition of $1mm^3$ or smaller.

**(3).** For work space that are moderately radioactive and so are not highly adverse to simple mechanical sensors, experiments with a contact sensor 'held' by the manipulator arm is proposed for detecting and correcting model assembly error in WVA. This is expected to achieve more robustness in phantom function for such application areas.

**(4).** Development of a 'palpation' simulation for tele-medicine is suggested by using the voxels to hold appropriate reaction model of body parts and rendering the joint separable reactions (appendix c), on very light structured arm.

**(5).** Operations in water submerged radioactive material handling pools pose different operating conditions for telemanipulators. Simulation systems can be developed using the virtual sensor framework. For operations involving partial submergence where free space (in air outside water) and submerged condition

# Appendix- A

## Remote Managed Long stand-off Sensing Technique for Workspace Modeling

Model based telecontrol systems need to use body models of robot and workspace objects. The objects residing in inaccessible work space pose the challenge of accurate spatial state assignment to their CAD models. Sensing of accurate position for pose determination of the target objects while maintaining large sensor standoff distance from the cell active space is feasible by building innovative machine intelligence based system around emerging laser range sensors. Use of Human intelligence and effective system assistance results in model fitting with good error control.

The problem is dual a: Object model fitting as per its instance in real world workspace b: object model identification. To start with, the problem is only as stated by 'a' but several measurements that are needed for 'a' can support robust techniques for achieving 'identification' i.e. objective 'b' as well to the approach. Image based object identification (object recognition) and volume reconstruction, though possible, are hampered by limited direction view [104].

## A.1 Object Identification Based Approach to Model Fitting

CAD representation, such as '.X' format files, describe object as set of bounding polygon shaped surface patches with color or texture attributes [105]. The approach followed here uses measurement of geometric features in 3D space rather than in image and their search in .X object descriptor files in object collection. The challenge lies in carrying out object-model match 'in-situ' in limited view condition from safe distance (figure A.1). Laser ranging is a powerful tool for noncontact distance measurement [106].

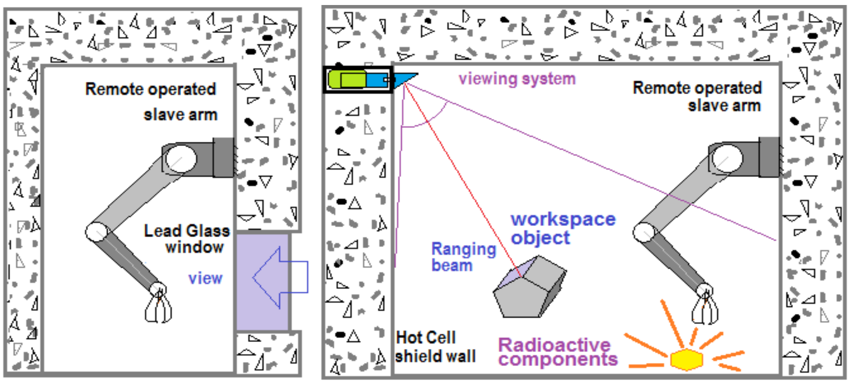



Figure A.1 (a; left) Old hot-cell working by viewing through transparency constrained radiation resistant glass window; (b; right) Long stand-off safe workspace model mapping approach.

Under highly constrained viewing conditions, sensing suitable sparse data related to salient exposed geometric feature is possible by equipping the user with remote measurement tools and object-model matching as well as 'in-situ' model fitting assistance from system.

The method is applicable for only those features that have presence in object geometric models. A feature may be related to a graphic feature too. The surface based models used for graphic rendering (like .X representation) are useful as they define volume occupancy and can create 3D virtual appearance suitable for human visualization [54]. These are necessary to assist user in result refining. Most useful measurable entities are surface patches represented by planar polygons. All nodes of polygons are point in 3D space, formed by intersection of 3 or more surfaces and are represented as data $P_{(x,y,z)}$. All nodes of an object are maintained as indexed list of $x, y, z$ coordinate data. The surface polygons are list of the indices formed in a sequence such that their forward and backward traverse form the two sides of surfaces represented by texture maps (bit patterns) [105]. The feature should be visible from outside and have definition in data form in the model. The basic measurable geometric feature is an edge as it is visible on object body. In the model it is represented by a pair of nodes. Closed set of nodes form polygon (smallest set is triangle)

### A.1.1 Spars-Data Approach for Finding Geometric Features

A single point viewing approach (figure A.1) relates 'measured point' P to sensor base $SB_{(x,y,z)}$ (figure A.2) by

$$P_{n(x,y,z)} = f(L_n, \theta_n, \phi_n) \tag{A.1}$$

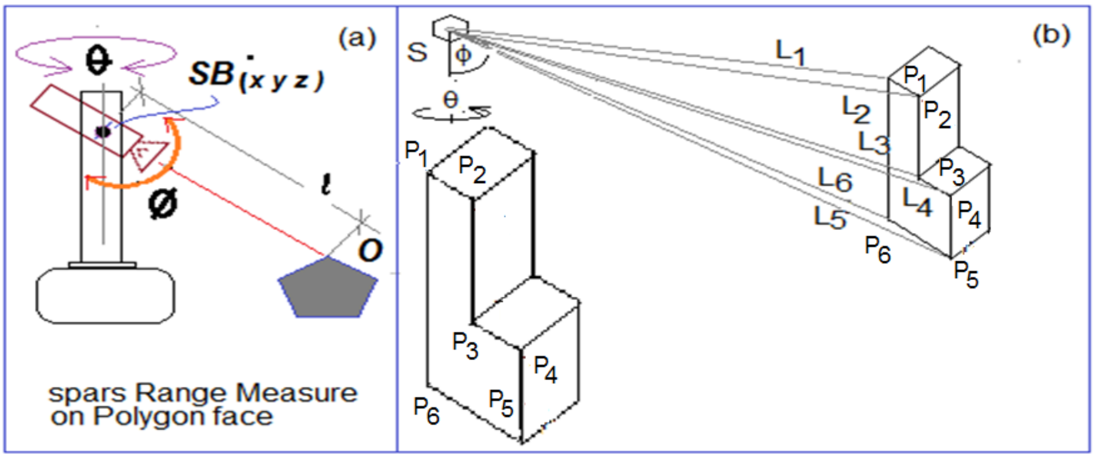



Figure A.2 a; laser targeting from same base $S_{(x,y,z)}$ , using $l$, $\theta$, ϕ $\phi$ for range to point measure and b; Feature metrics formation.

Nodes are formed by three plane (face) intersection and have definite X,Y,Z co-ordinates in space. The node's location in workspace can be found by measuring length '$SP$', angle $\theta$ and $\phi$ of vector originating at sensor location $S(x, y, z)$ and oriented to reach the node point as its target '$P$' (figure A.2a). Vector is referred as Laser measure vector (LMV) in this paper. Spatial position of '$P$' is found by

$$P_x = L \cdot cos\phi \cdot cos\theta$$
$$P_y = L \cdot cos\phi \cdot sin\theta$$
$$P_z = z_0 - L \cdot cos\phi \tag{A.2}$$

Geometric features (GF) that define the surfaces partially are two sides and included angle (for corner), set of two connected or more un connected sides, three connected edges with each edge common to pair of surface. For fully visible surface, forming match vector using basic edges is possible (figure A.2). All match and orientation vectors are formed using $P_n$.

Edges are formed by 3D vectors $V_n$ defined by node pair

$$V_n = P_k(x, y, z) - P_{k+1}(x, y, z) \tag{A.3}$$

and form the basic data for the geometric feature listed above. Angle $\beta$ between $V_n$ and $V_m$ where

$$V_n = P_k(x, y, z) - P_{k+1}(x, y, z) = \overrightarrow{e_1} \tag{A.4}$$

$$V_m = P_{k+1}(x, y, z) - P_{k+2}(x, y, z) = \overrightarrow{e_2} \tag{A.5}$$

$$\beta = \cos^{-1}(\overrightarrow{e_1} \otimes \overrightarrow{e_2}) \tag{A.6}$$

(Note: $\otimes$ is vector cross product )

### A.1.2 Sparse Data based Model fitting

Model fitting to the viewed object's instance involves i) face association to description in model(polygon match); ii) model orientation to align respective surface normals of measured faces) ;iii) rotation of model face around normal to make edges parallel and iv) translation of rotated model to superimpose measured nodes on workspace object nodes [105].

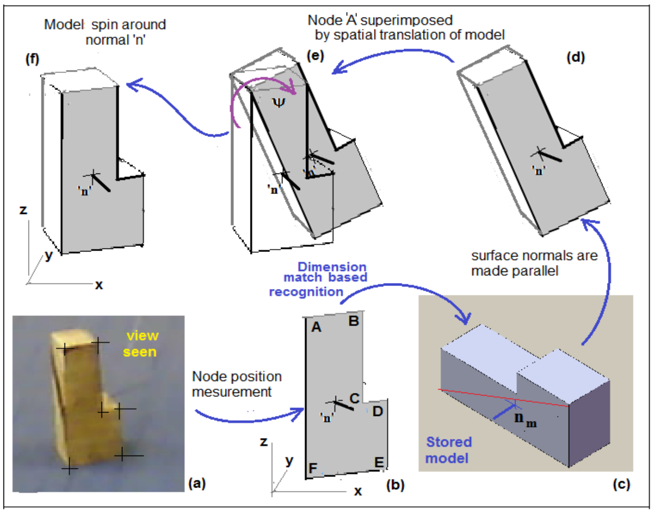



Figure A.3: steps in on-line spars data based in-situ model fitting in workspace

## A.2 Experimental Setup

An experimental setup has been built to facilitate measurement of visibly identified shape features and machine intelligence tools to test feature metrics fitting approach.

### A.2.1 System Configuration

A system (figure A.6) has been formed to implement above method as single integrated instrument. It supports feature metrics formation on object seen through viewing port, facilitates supervised measurement, provides object identification assistance, contains effective a-priori data and structured context sensitive access

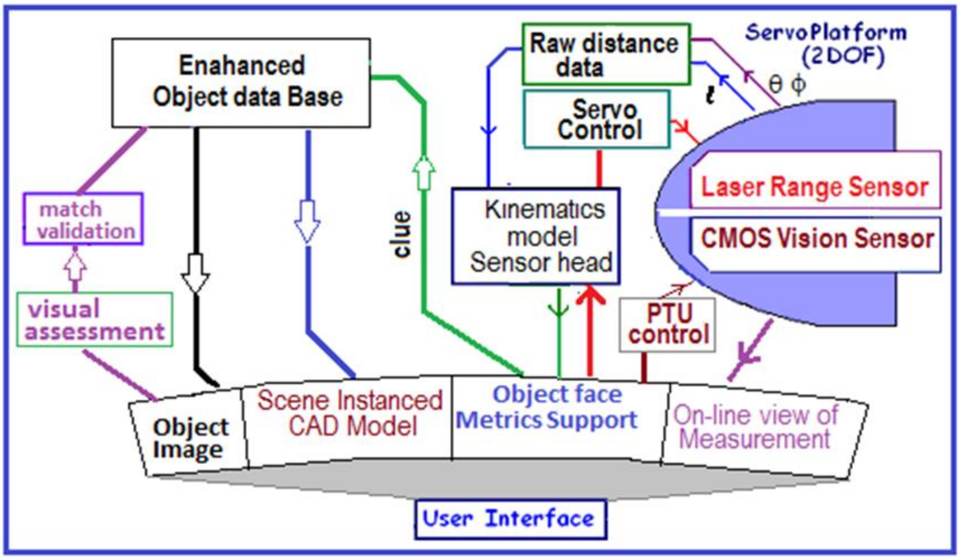



Figure A.4: System schematic of the feature metrics based model fitting system

methods and executes rendering for interactive identity and fitment validation.

In this approach employs an arrangement located at top of cell (figure A.1) can be used for 'time of flight' sensing on long range (3500mm or more). The ranging beam is visible (670nm) and has low divergence. 2DOF servo platform hosting the 'Laser Range Sensor' 'LRS' of rotation type is used. (figure A.5).

### A.2.2 Measurement Environment

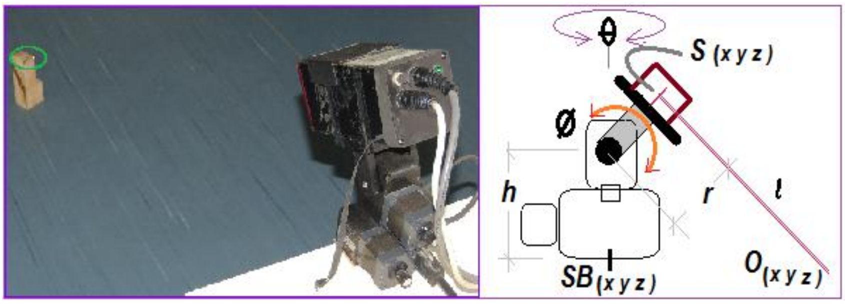



Figure A.5: (left) long stand-off range head, sensing feature distance ;(right) kinematics model for data formation.

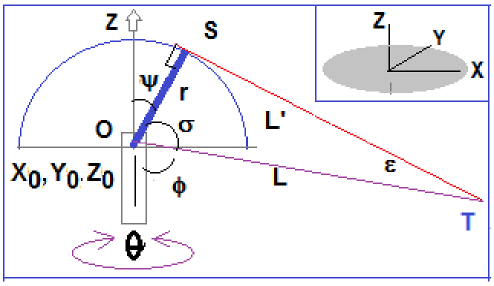



Figure A.6: Sensor head kinematics model for correcting raw range data.

The user can form a vector from the measurement base 'O' by interactively guiding the beam spot on feature by changing pan-tilt angles (figure A.6) using CMOS imager generated on-line streamed views (figure A.4). Ideally sensor location must not change during the laser targeting, but in implementation here, the platform changes position and introduces non-trivial errors. Controlled directions are $\sigma$ and $\theta$. '$r$' is distance of sensor center from rotation base, measured distance is L$'$ (figure A.6). Corrections are computed as following for use with Equation(A.2) in finding target point coordinates relative to senor base.

$$L = \sqrt{L'^2 + r^2} \tag{A.7}$$

$$\phi = 180 - \psi - \cos^{-1}\left(\frac{r}{L}\right) \tag{A.8}$$

The system computes and records the target point's position $(P_x, P_y, P_z)$ and forms sets of $P_n$. Computation modules form $P_n - P_{n+1}$vectors, compute corner angles between $(P_n - P_{n+1})$ and $(P_{n-1} - P_n)$ in plane of $P_{n-1}, P_n, P_{n+1}$ analyze polygons formed by $P_n$ set.

### A.2.3 Interactive System Support for Metrics

#### A.2.3a Assisted identification

Measurement based identification may not be possible with sparse data unless uniquely identifying feature appears in view. Partial insufficient feature may result in multiple matches and in such case all such matches are presented to user for human assisted resolving.

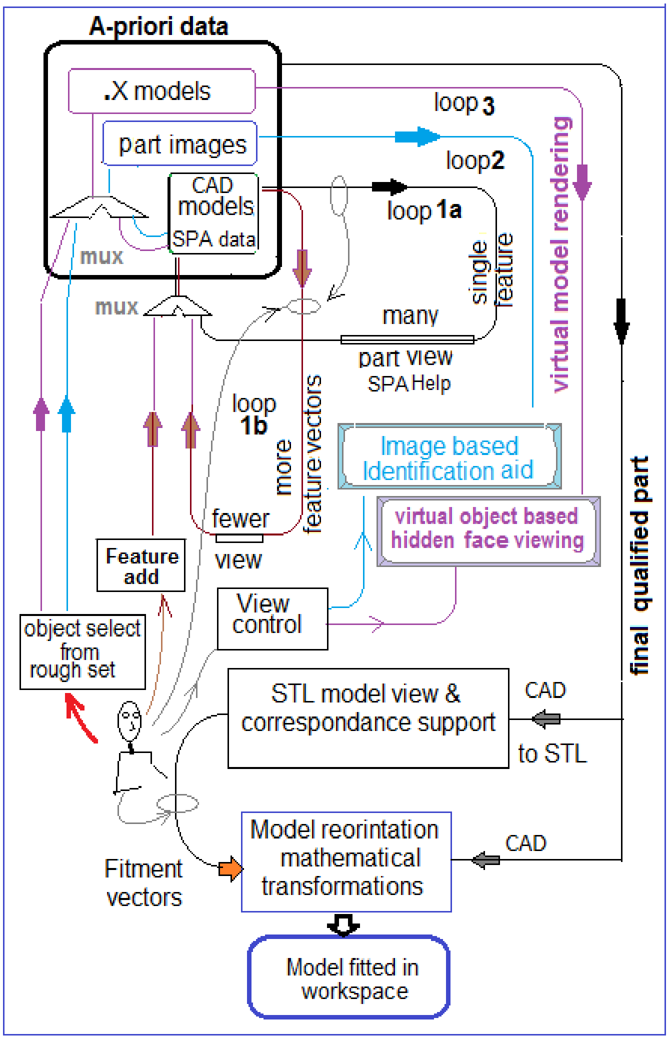



Figure A.7: System function and operator interactions

First with few measured feature search can be made (loop 1a figure A.7). The result may have many short listed objects presented on console and model can be seen as thumbnail image formed for assistance. User may either accept the object if good identification is possible with available on-line view or can add more features by measuring and forming richer match vectors for the search system to minimize the set of qualifying objects. In most cases, the list shrinks (loop 1b, figure A.7) If by either methods the list cannot be reduced to one object then user can make use of graphics rendered models that can be rotated to see objects model from same relative view direction as seen in the constrained viewing condition as prevailing in the cell. With this approach the object is identified in almost all cases.

### A.2.4 System working

Computer hosted platform employs laser range sensing for model matching and uses sparse data route. CAD data of all the parts are augmented with key identifying aspects like unique dimension, long to short side ratio (for thin long objects). Complex polygon surface and their images are maintained in a relational database connected through part name as 'key' (figure A.8). On simple match basis like 'side length', all objects with at least one dimension matching within specified error limit are short-listed. Multiple qualifications may result on such simple criteria. User is presented with all of them in order of match. For improving match the user can add metrics as suggested by system help. Based on presented images too, one may identify most fitting object. Selection may be narrowed down by interactively introducing more measured criteria.

CADData : Table

| Field Name | Data Type | |
|---|---|---|
| ObjectID | AutoNumber | Unique ID of object |
| PathImageShort | Text | Short path of object folder |
| # of Views | Number | Number of surfaces in the Body |
| PathImage | Text | Complete path of object's principal Image |
| PathSurface | Text | Body Name/Path |
| PathSPAF | Text | SPAF Name/Path |

Figure A.8: Multimode data integration in 'a-priori' data collection with their linkage for structured access and rendering

### A.3 Model Fitment Methods

A sensor hardware set up (figure A.4) is used in laboratory. A set of objects with flat surfaces were formed by choosing them such that some dimensions of them were same for different objects and few had unique surface shape (figure A.3, A.9).

#### A.3.1 Model Identification

Part association to model is established by assisted identification. For experiments, objects chosen from the samples were placed in space at a stand-off distance of approximately 3 meters. The CMOS camera view was set to cover full object in maximum space of its view by optical zooming.

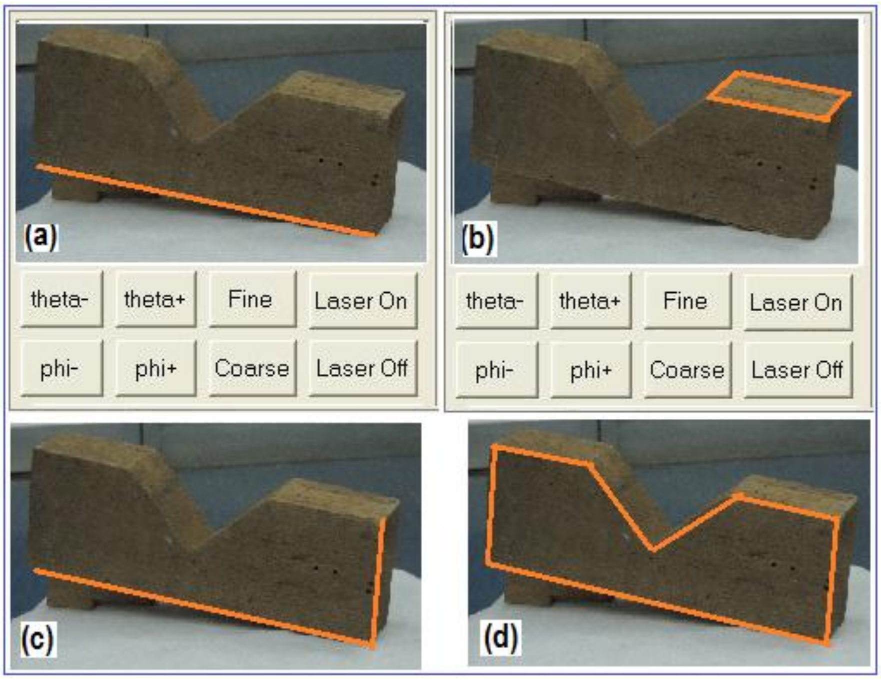



Figure A.9: a; remote measurement console showing measurement on wedge object, single edge vector measured, b; one face area measure, c; height and length measurement, d; face shape measurement.

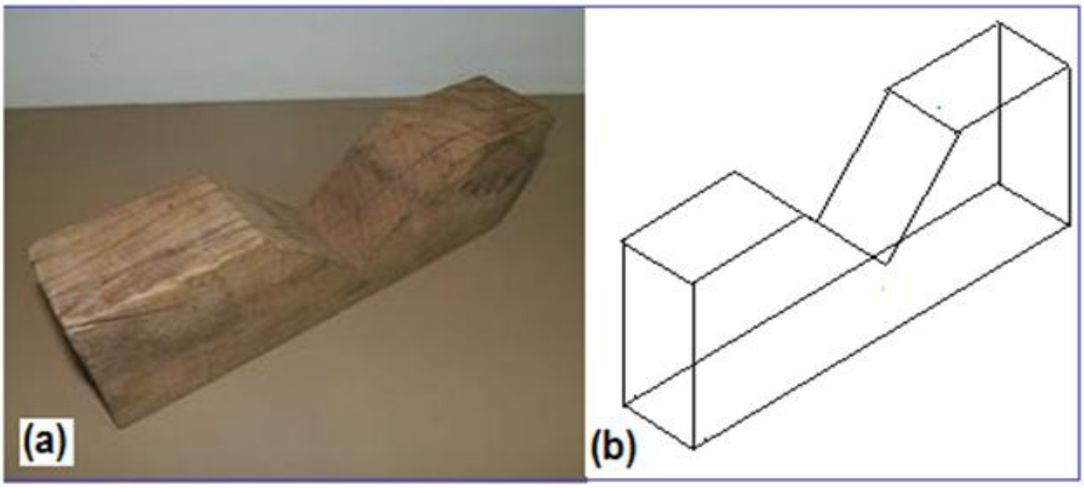



Figure A.10 Registered image of the object presented on user assistance call on multiple choice in (a), b; wire diagram that can be rotated to see details of occluded side of object .

### A.3.2 Model Fitting into Measured Data

Model is fitted in the object's position by geometrical transformation based pose imparting to model [54] as the object's measured pose in model position and then translating it to the object's measured position. The method implemented in this work (figure A.11) computes the normal vectors on the measured node based flat surface containing polygon face and identified object model's matching face. Rotation of all the nodes in the node list of the object's '.X' representation around the computed normal vector **'$\boldsymbol{n}$'** by an angle '$\alpha$' in appropriate direction. The measured face and CAD model face are then parallel to each other. The spin around normal '$n2$' is performed (figure A.12) to make all the edges parallel to the corresponding edges of measured edges. Last step is to translate the nodes by a translation vector formed by object node to corresponding model node. Since entire translation accuracy depends on this operation a precisely measurable, well viewed corner should be used for this operation.

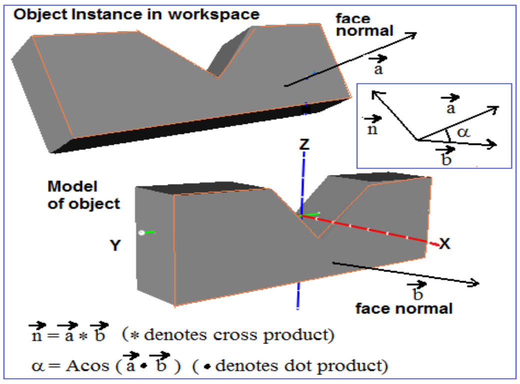



Figure A.11: Transformation for making identified surface parallel to measured object instance

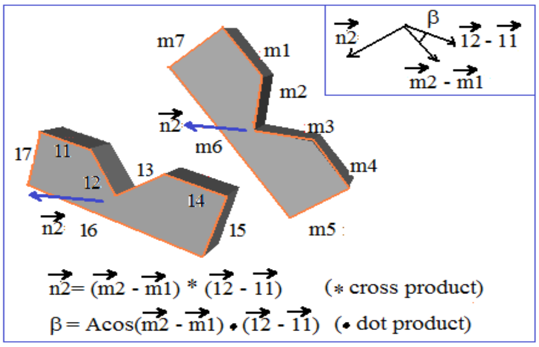



Figure A.12: Model edges set parallel to sensed object's edges.

### A.3.3 Assisted Orientation

For symmetric objects visual resolving is carried out by rendering all possible model instances, on line view and pre-registered images. Textured virtual views too are offered for maximum assistance to the user. For handling symmetrical objects such as cube (6 equal faces) and rectangular bar (3 pairs of equal faces) image data is most important requirement. Note that volume occupancy model is unaffected for 3D symmetric objects but for achieving consistency in view, resolving correct orientation is preferable.

## A.4 Sources of Error

### A.4.1 Instrument error

Though careful measurement to reduce error by meticulous approaches indicated above improves working, it is difficult to mitigate the effect of instrument error generated problems. The target range based dimension measurement suffers errors owing to the instrument limitations [107]. For angular resolution of $\delta\theta$ and $d\phi$ of servo platform and ranging error of $\pm\delta r$. A range vector originating at $P$ (figure A.13) may actually end anywhere on a spherical surface segment spanning over $2\delta\theta$ and $2\delta\phi$ angle at radius $r - \delta r$ in maximum negative range error condition. It may be on a spherical surface segment spanning over$2\delta\theta$ and $2\delta\phi$ angle at radius $r + \delta r$ in maximum positive range error condition. In effect the target

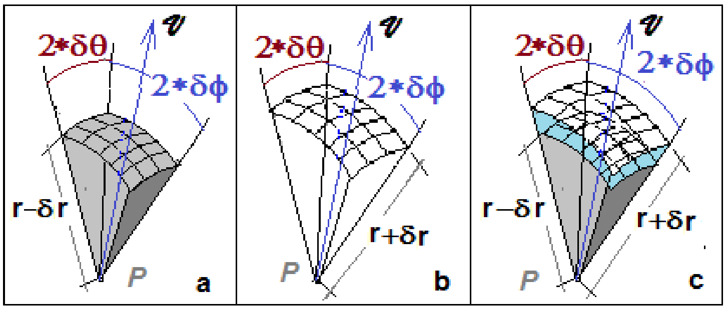


Figure A.13: errors in target position measurement

point formed by the vector end may actually lie within the space bound by the two error surface segments. Another notable aspect is that the error $\delta r$ is also dependent on $r$ as it is higher for longer r, resulting in larger uncertainty space for the vector end's position. An edge represented by 3D vector $V(P_k(x,y,z) - P_{k+1}(x,y,z)$ formed by long '$L$' laser vectors (figure A.5) will have higher length and orientation errors as $P_k$ and $P_{k+1}$ may lie in larger uncertain spaces. Also if $P_k$ and $P_{k+1}$ lie closer to each other the % errors (important in matching) will be higher. This is the cost of higher stand-off distance on performance.

### A.4.2 Human Error

The laser spot placement on a feature is difficult in insufficiently illuminated zones of scene. Also the spot formation on inclined surfaces where feature is on such a surface, results in deformed shapes of spot. Error in targeting the features at spot sensor can be erroneous. In such cases the route measure is on adjacent location and not on a corner or edge.

### A.4.3 Error Effect on Object Model Identification

Features like edge length, angle between vectors and area have same effect in identification stage as that caused by other registration errors. It is nullified on validation unless highly similar parts in size shape exist as members in database. But for model fitting purposes the error effects may be profound and difficult to minimize.

### A.4.4 Orientation error source, its behavior and its minimization

On a face ranged by near normal measuring laser vectors the $\delta\theta$ and $\delta\phi$ errors cause area change and range error '$\delta r$' cause orientation errors. Node location errors lead to position and orientation errors in fitted model. The range sensing error caused orientation inaccuracies are difficult to correct.

Unless care is taken, errors can even scale up for larger object as evident from following discussions.

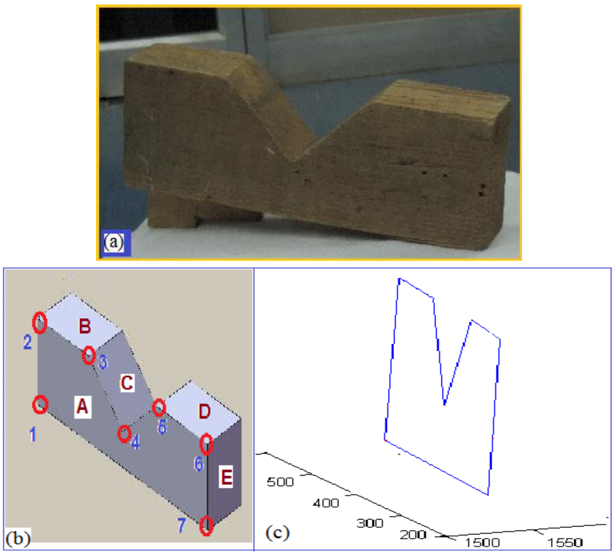



Figure A.14 a; object in view; b; targeted nodes for face reconstruction, c; result as face polygon in 3D

### A.5 Error Correction Aid

For generating a face measurement based data user selects a face ‘A’ and targets nodes 1-7 (figure A.14a and A.14b) to form a polygon for use by the process (figure A.14c). To understand the range error effect we introduce errors in range data ‘l’ of the two targets nodes 3 and 5 (figure A.15a). The error is not easily perceivable. As detection tool, normals at these nodes are drawn with fixed length (red) at nodes using two adjacent edge vectors and shown in 3D views. Correct registration of node s must result in parallel normals for nodes on planar surface. Conversely their non parallelism indicates registration error. Identification of the erroneous node suffering from range error effect allows improved working, as user can correct it by re-measuring range data on the node.

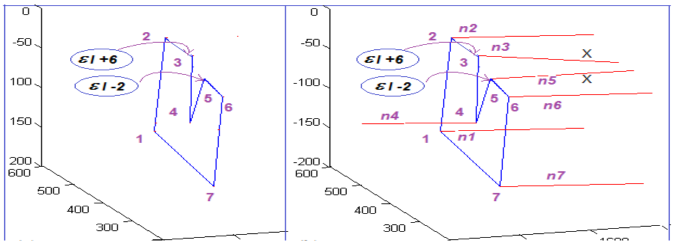


Figure A.15 a; polygon with erroneous nodes 3 (positive range error 6mm) and 5 (negative range error 2mm), b; normals at nodes(red).

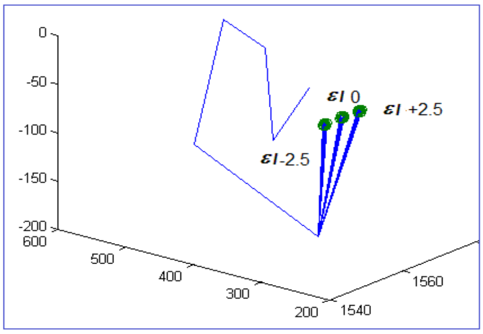


Figure A.16: Possible location of a node owing to instrument errors. Note magnified 'y' axis to emphasize error in node location viz a viz other nodes. Note cluster formation from directing $(\theta, \phi)$ errors

### A.5.1 Reduction of Model Fitting Error

A node measurement may place the node in space with errors (figure A.13). The servo platforms orienting errors, form a point cloud if repeated targeting is done for some target features and $T_n$ may appear in clusters (figure A.16) for maximum range $\ell$+2.5 mm or minimum $\ell$-2.5 mm (specified for instrument). Hence a node may appear in any cluster or in between them. This is common phenomena for all the nodes. ). Use of nodes 6, 7, 1 limits the error to cluster shown in figure A.18.Use of node 5, 6, 7 (figure A.17) for determining normal of the surface polygon 1, 2, 3, 4, 5, 6 and 7 results in higher displacement of other node farther from 2 (i.e. node 6) after model fitting (figure A.19).

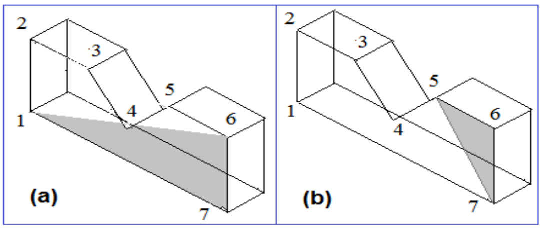



Figure A.17: use of surface segments (triangle) for computing face orientation for entire face

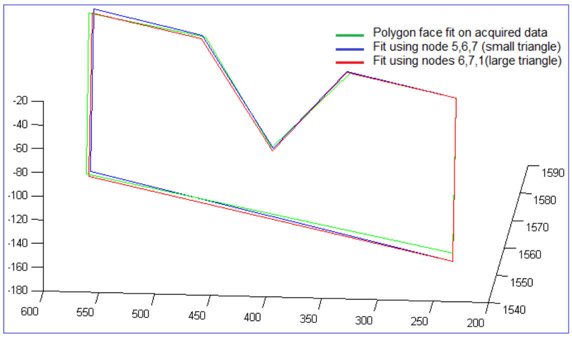



Figure A.18: No error introduced at node 6 but the actual error in data play their role in fitment. The fitment error is less if nodes 1, 6, and 7 are used as compared to that when nodes 6,7 and5 are used. Note- plot axes are not equal.

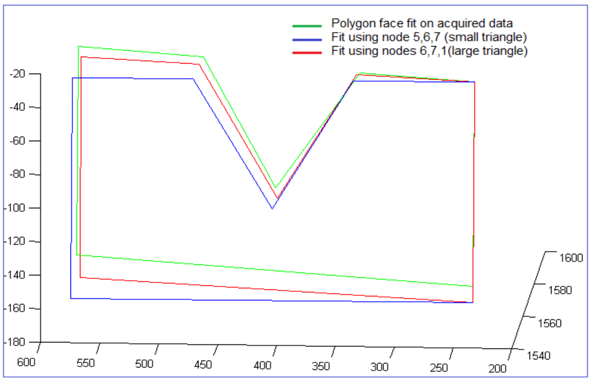



Figure A.19: Case with +ve range error introduced at node Fit using smaller triangle node set 6.7 and 5 is more erroneous. Note- plot axes are not equal

Therefore there is definite benefit of choosing larger triangle forming nodes' data use in fitment of model (figure A.20). The observations are reinforced by the fitment experiments' results (figure A.20 to A.22). Note that node registration offset for node 6 has resulted in translation error in body too. The ranging error chiefly effects model orientation accuracy.

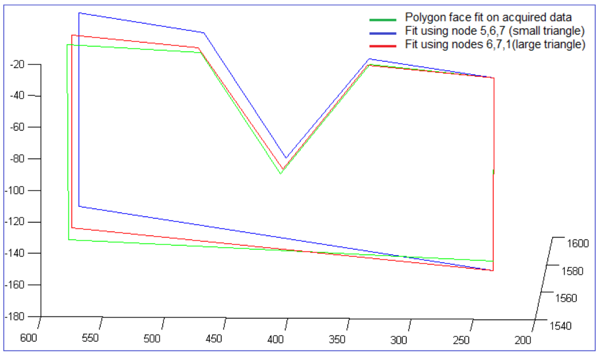



Figure A.20: Case with -ve range error introduced at node 6. Note: for accentuating error rendering, different plot axes scale have been used.

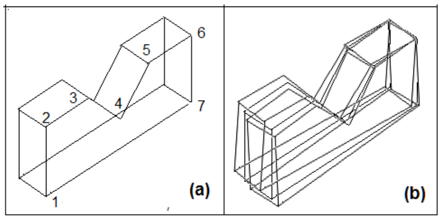



Figure A.21: Error after fitting a model at node 2 owing to possible ranged positions of node 6 resulting from ranging error of LRF.

### A.5.2 Nullification of errors in applications of the model

The workspace models are used for mapping of volume occupancy where the remote operator focuses on tele-manipulation of end-effector while the environment aware machine intelligence system implements haptic percept creation. The workspace model created here (figure A.22) forms input for voxelised representation formation. Voxel size is determined by application. The maximum node displacement errors, in the voxel modeled workspace, may be less than an integer $(j)$

number of voxels (generally$j \leq 2$). As many as '$j$' layers of voxel can be wrapped on object's work-space instantiated voxel model to work as protective skin (figure A.23). This effectively nullifies the errors in long standoff approach related errors as brought out here.

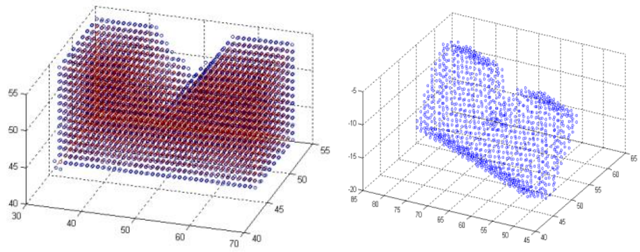


Figure A.22: (left) voxel model of wedge object surrounded by skin voxels (blue) to compensate error on safe side, (right) shows voxelised of fitted model on object instance as detected by spars data method.

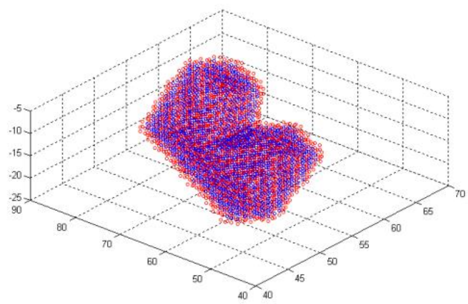


Figure A.23: Voxel model of wedge object as detected by spars data method and then wrapped by skin voxels (blue) fitted model on object instance.

# Appendix B

## Lean Computation For Sensor Location Accessing

**Introduction:**

The environment sensing used in HAVP , DFAP and VADP need computation of large number of sensor positions in workspace that dynamically change with robot arm movement. The computations have to be completed in short time to sustain the haptic rendering loop run by the phantom sensor system for rendering the percepts above a rate of 20 Hz. The number of sensors depend on the robot arms surface area and $WVA$ voxel dimensions that dictate sensor placement pitch (chapter 5). The computation linearly increases i.e. $\boldsymbol{\mathcal{O}}^{1}$ with number of sensors. In the simplest approach each sensor can be treated as connected to the link axis(figure C.1) at a distance from joint location with a rigid link of length '$r_p$' at angle $\Psi_p$ as mentioned in chapter 5 section 'sensor structuring'. Here an innovative approach is introduced to reduce computations.

### B.1 General kinematics approach

Since joint axis needs to be computed always, we assign Link Axis Start Point $LASP$ and Link Axis End Point $LAEP$ to each link . Forward kinematics transforms starting from base and proceeding through the joints when applied on a robot (figure B.1) give positions of $LASP(n)$ and $LAEP(n)$ .

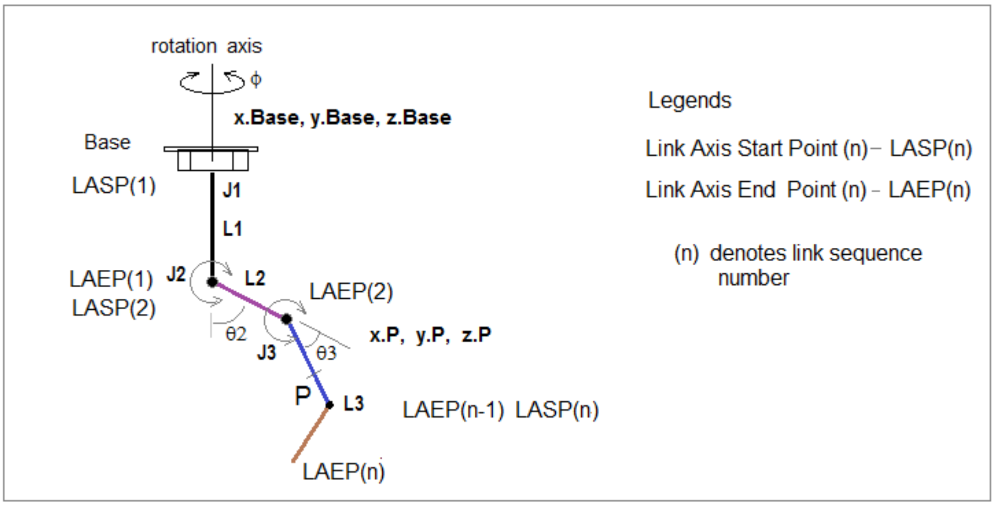



Figure B.1: Link axis based kinematics computations for lean computation based sensor location determination

The transformations effectively cause 3 degree of rotations and 3 degree of translations to the rigid bodies representing links and orient and position the axes in space as shown in figure B.2 here.

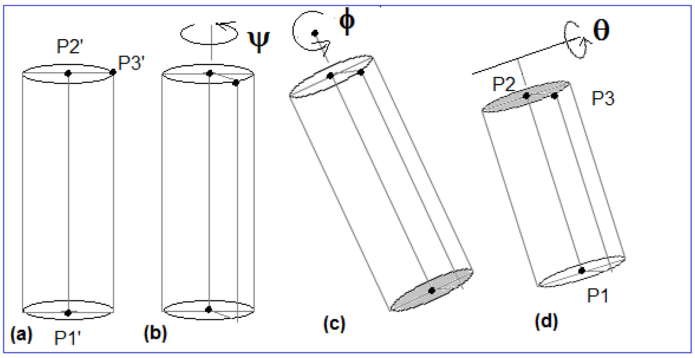



Figure B.2: Translation, Rotation on robot link

A link body that is a child of rotation joint undergoes 3 types of rotations depending on joint function. It can be translation too. These may undergo a combination of these depending on the robot DOFs . The sensor point on link shell also undergo the same rigid body motions. Therefore the sensors are defined in relation to the $LASP - LAEP$ vector (figure B.3).

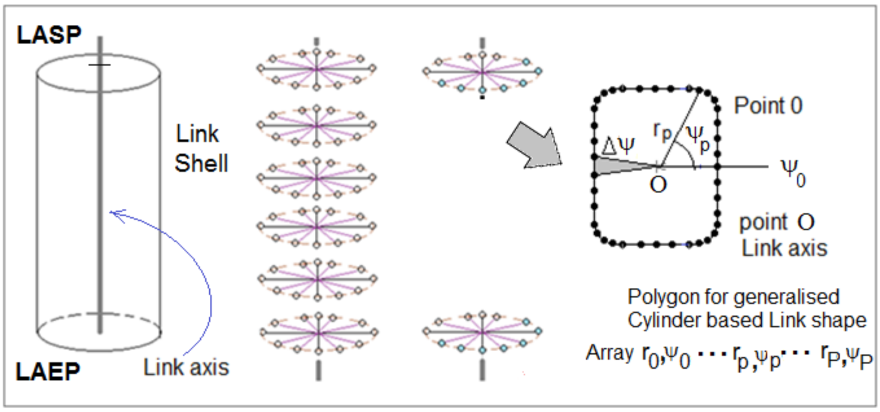



Figure B.3: Sensor locations on a link shell.

For a circular cylinder they appear like spokes on axle. For generalized cylinder they are defined by an array of data sets representing distance of $p^{th}$ point at distance $r_p$ from link axis '$O$' at angle $\Psi_p$ from reference start angle $\Psi_0$. The Sensor locations may be treated as nodes connected by rigid links to the link axis vector $LASP - LAEP$ and subjected to same transformations as the link body is. All the sensors along the body shell (figure B.3) can be assigned $x, y, z$ coordinates with respect to the robot base in $WVA$ Cartesian space and base location $X, Y, Z$ being known, location in workspace is found.

**Computationally lean approach to locate sensors:**

Let there be two vectors rigidly connected orthogonally to each other as features of a physical object (here link). The vectors are defined by three points $P1', P2'$ and $P3'$ Assume that owing to general transformation like translation, rotation $\theta$, rotation $\phi$ of unknown magnitudes the points become $P1, P2$ and $P3$.

With translation $T$ ,where

$$T = \ P2' - P2$$

$P2$ shall coincide with $P2'$ (figure B.4a)

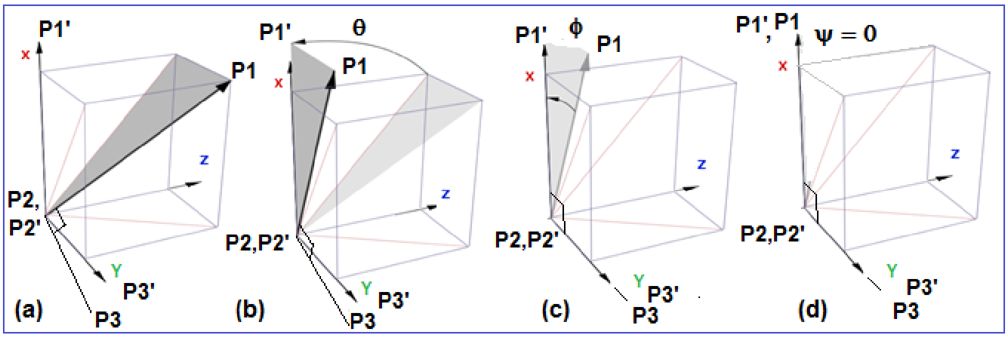



Figure B.4: Aligning vector set for finding 2D rotations

Now the problem remains to align the two vectors.

Align the $P2P1$ vector along $P2'P1'$ vector

Let $P2P1$ be $(u_1, v_1, w_1)$ and $P3P2$ be $(u_2, v_2, w_2)$ where

$$u_1 = x_2 - x_1;\ u_2 = x_3 - x_2;$$

$$v_1 = y_2 - y_1;\ v_2 = y_3 - y_2;$$

$$w_1 = z_2 - z_1;\ w_2 = z_3 - z\_2\,;$$

Two rotations are required for a vector to be aligned along x axis.

i. Rotate around y axis by $\theta$ to bring vector along x-y plane.

ii. Rotate around z axis by $\phi$ to map vector from x-y plane onto x axis.

**Step 1:**

Rotation around y axis:

$$\Theta = \cos^{-1}\left(\frac{u_1}{\sqrt{(u_1{}^2 + w_1{}^2)}}\right)$$

The matrix required will be then

$$Ry = \begin{bmatrix} \frac{u_1}{\sqrt{(u_1{}^2+w_1{}^2)}} & 0 & \frac{-w_1}{\sqrt{(u_1{}^2+w_1{}^2)}} \\ 0 & 1 & 0 \\ \frac{w_1}{\sqrt{(u_1{}^2+w_1{}^2)}} & 0 & \frac{u_1}{\sqrt{(u_1{}^2+w_1{}^2)}} \end{bmatrix}$$

**Step 2:**

Rotation around z axis:

$$\Phi = \cos^{-1}\left(\frac{\sqrt{({u_1}^2 + {w_1}^2)}}{\sqrt{({u_1}^2 + {v_1}^2 + {w_1}^2)}}\right)$$

The matrix required will be

$$Rz = \begin{bmatrix} \frac{\sqrt{({u_1}^2+{w_1}^2)}}{\sqrt{({u_1}^2+{v_1}^2+{w_1}^2)}} & \frac{v_1}{\sqrt{({u_1}^2+{v_1}^2+{w_1}^2)}} & 0 \\ \frac{-v_1}{\sqrt{({u_1}^2+{v_1}^2+{w_1}^2)}} & \frac{\sqrt{({u_1}^2+{w_1}^2)}}{\sqrt{({u_1}^2+{v_1}^2+{w_1}^2)}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With rotation of joints the point $P1'$, $P2'$ and $P3'$ undergo same translation and rotations together being on a rigid body . $P1'$ and $P2'$ are the ends of a link or segment and P3' is the section polygon's 1$^{st.}$ point on PBP. These can be computed by Forward kinematics computation (C-1 )[40] for the link.

For given $P1, P2, P3$ resulting from transformations on vector $P1'P2'$ and $3'P2'$ , as shown in figure B.3 and described in previous section, T, θ and ϕ can be computed by using step 1 and 2.

If vector $P1P2$ has undergone spin then the point $P$ shall not coincide with $P3'$ and $\psi$ is computed by following method:

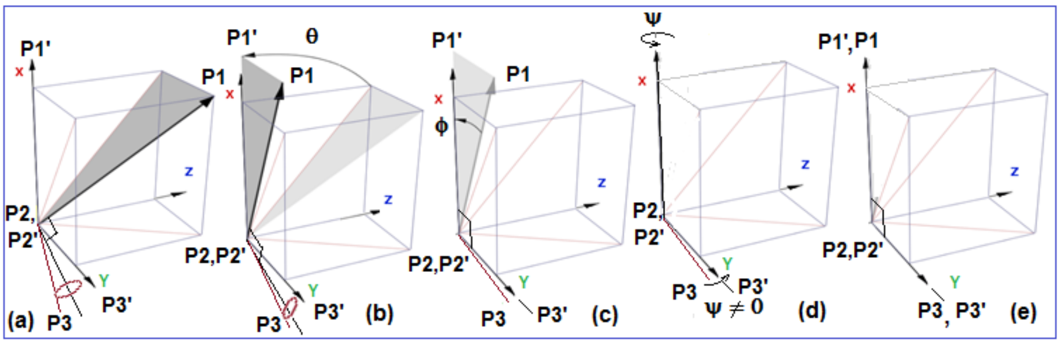



Figure B.5: Vector set alignment for 3D rotations

**Step 3:**

Rotate around x axis by angle $\psi$:

$$\psi = \cos^{-1}\left(\frac{\sqrt{(v_2{}^2 + w_2{}^2)}}{\sqrt{(u_2{}^2 + v_2{}^2 + w_2{}^2)}}\right)$$

The matrix required will be

$$Rx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{(v_2{}^2+w_2{}^2)}}{\sqrt{(u_2{}^2+v_2{}^2+w_2{}^2)}} & \frac{u_2}{\sqrt{(u_2{}^2+v_2{}^2+w_2{}^2)}} \\ 0 & \frac{-u_2}{\sqrt{(u_2{}^2+v_2{}^2+w_2{}^2)}} & \frac{\sqrt{(v_2{}^2+w_2{}^2)}}{\sqrt{(u_2{}^2+v_2{}^2+w_2{}^2)}} \end{bmatrix}$$

The dataset $T, \theta, \ \phi \ and \ \psi$ fully define the transformations required to locate the vector $P1P2$ and $P2P3$ on $P1', P2'$ and $P3'$

Following the procedure of determining $\theta, \phi, \psi$ and T by reverse mapping of $P1, P2$ and $P3$ on $P1'$, $P2'$ and $P3'$ respectively the applicable parameters are computed for the link. Using these translation and rotation parameters all $PTP$s are computed using their intrinsic spatial coordinates at joint parameters corresponding to its home state. For $\omega$ <> 0 the $PTP(0)$ is first rotated by $\omega$ and then points are computed for $PBP$s using same transformations parameters $\theta, \phi, \psi \ and \ T$ (figure B.6).

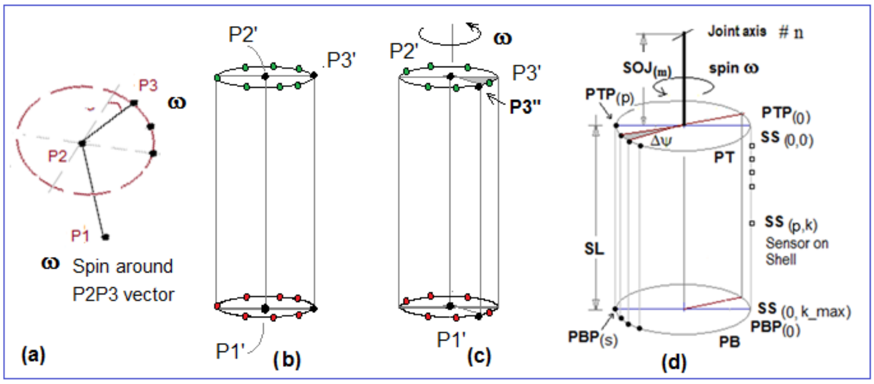



Figure B.6: Determining $PTP$s from vector set $P1P2$ and $P3P2$

**Fast computation of PBPs**

Case 1: For link type 1 apply translation $T = \boldsymbol{P2P1}$ on the above computed PTPs.

Form sets $PTP$s and $PBP$s

Intermediate sensor points $SS_{p,k}$ are found by simple interpolations between $PTP_n$ and $PBP_n$ on the link shell at present state.

Case 2: For link type 2 computation of $PBP$ is done by transformation as used for $PTP$s. Note that this is double computation as compared to that for link type 1. Subsequently interpolation technique is used between $PTP_n$ and $PBP_n$

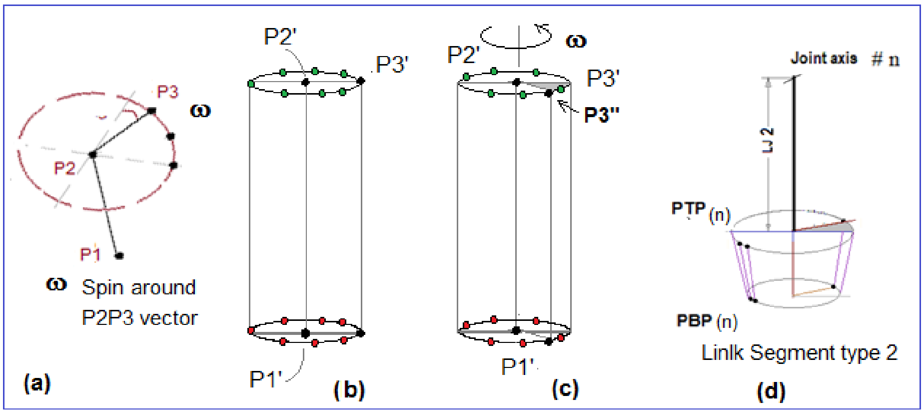



Figure B.7: Determining PTPs from vector set $P1P2$ and $P3P2$

**<u>Calculations Involved</u>**

Add/Sub=12***1** = 12 FPC

Multiply=12***2** = 24 FPC

Divide = 3***23** = 69 FPC

SQRT = 5***23** =115 FPC

TRIG = 3***1** =3 FPC

Total FPC required=**223**

**Note:** Here S=Subtraction (1 FPC), A=Addition (1 FPC), E=Exponent, SR=Square Root (23 FPC), M=Multiplication (2 FPC), T=Trigonometric (1FPC, considering lookup tables), C=Comparator (1 FPC), D=Division (=23 FPC), L =Logical Operation (1 FPC), CG=Centre of Gravity

**Computation reduction in torque computation:**

This needs the move vector computation and the $\mu$ value in WVA at the sensor position.

**i.** For the case where joint operation is translation the required computation of direction and magnitude of motion vector needs only single computation as explained in figure B.8. The μ values will be applicable at each $P_n'$ and need to be accessed from WVA model for which linear interpolation based computation is already explained.

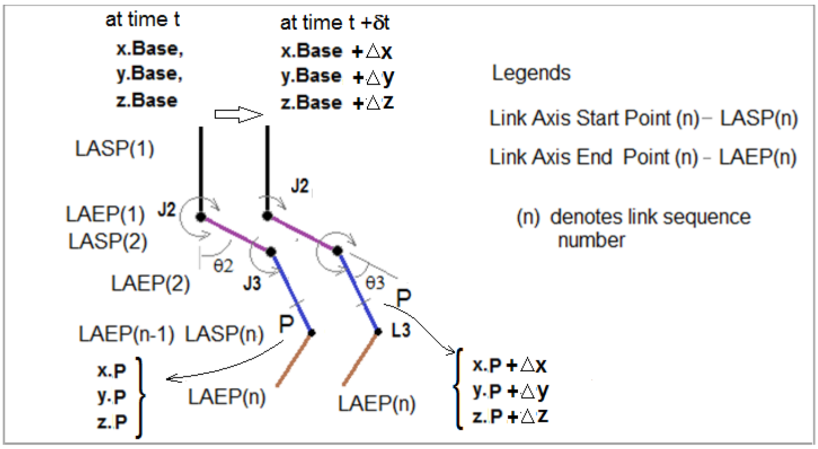



Figure B.8: Computation reduction in translation of a parent joint (here base joint).

**ii.** For the case where joint operation is rotation the required computation of direction and magnitude of motion vector needs computation of a: sensor distance from axis of rotation b: motion vector magnitude computation. The vector direction is not computed as the effective torque is in direction opposite to rotation caused by the controlled operation.

Intermediate sensor points $SS_{p,k}$ are found by simple interpolations between $PTP_n$ and $PBP_n$ on the link shell at present state.The $\mu$ values will be applicable at each

$SS_{p,k}$ and need to be accessed from $WVA$ model for which linear interpolation based computation is already explained.

Typical examples are provided in appendix – C.

# Appendix C

## Drag Effect for a Joint Motion

Position of end-effecter in the inertial frame is given by kinematics transformations:

$$\mathbb{H} = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} = \mathbb{T}_{0n} = A_1(q_1) \cdot A_2(q_2) \cdots A_n(q_n) \qquad \text{(c.1)}$$

Where $A_n$ are the kinematics operations and $q_n$ are the robot parameters [40] for a typical robot arm (figure C.1).

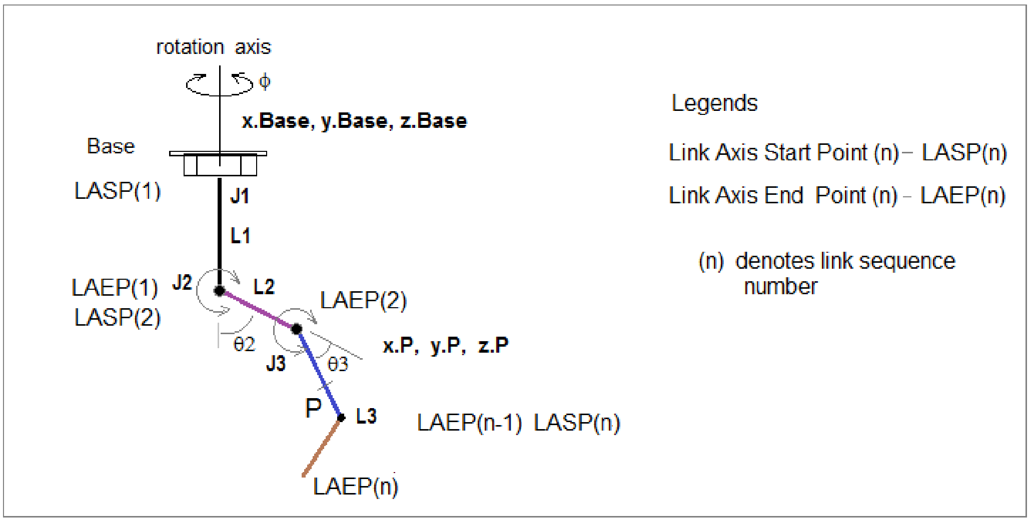



Figure C.1: Link axis based kinematics computations for lean computation based sensor location determination

The $LASP(n)$ and $LAEP(n)$ are respectively Link Axis Start Point and Link Axis End Point of link number '$n$'. These can be treated as the points $P_1, P_2$ (Appendix B) and used for determining positions of Link Shell Sensors of given link from which $PTP$s and $PBP$s can be located in workspace $WVA$. Since $LASP$ and $LAEP$ are computed by using the kinematics transformation chains (relation c.1), the sensor positions in $WVA$, velocity estimation and applicable $\mu$ access from $WVA$ property is feasible.

For finding drag seen for a particular joint operation following approach is developed.

For the $l^{th}$ link joint $LJ_l$ the sensor $S_{s,k}$ on child link of Joint $LJ_u$ develops reaction only if its host link $LJ_u$ is not above the $l^{th}$ joint i.e. $u \geq l$.

Note, for base joint $n = 0$ , the relation (c.1) modifies to

$$\hat{T}_{0lu} = A_1\big(q_{1(t-\delta t)}\big) \cdot A_2\big(q_{2(t-\delta t)}\big) \dots A_l\big(q_{l(t)}\big) \dots A_u\big(q_{u(t-\delta t)}\big) \tag{c.2}$$

For computing sensor contribution to the joint $LJ_l$, forward kinematics for a single active joint is done at a time for assessing effect on its motion. For $(0 \leq n < l)$ the shell sensor points on link $LJ_n$ at time $(t - \delta t)$ is at same location as at time $(t)$ and so their contribution in drag is zero.

Let $\Pi_{\mathcal{M}}$ be plane of motion. Workspace position of a sensor $SS_{p,k}$ on segment $m$ on Joint $LJ_u$ is denoted as

$$\boldsymbol{u\_m\_CSS_{p,k}}$$

Effective distance $EDL$ of sensor on $m^{\text{th}}$ segments of $u^{\text{th}}$ link from rotation axis of $LJ_l$ is denoted by

$\boldsymbol{EDL\_\Pi_{\mathcal{M}}\_l\_u\_m\_(p,k)}$ and referred further as $\mathbb{D}_{uml}\,(p,k)$.

Effective position displacement in $\Pi_{\mathcal{M}}$ is denoted by

$\boldsymbol{EPD\_\Pi_{\mathcal{M}}\_l\_u\_m\_(p,k)}$ and referred further as $\mathbb{P}_{um}\,(p,k)$

$\mathbb{P}_{um}\,(p,k)$ has 3 components along $x, y, z$ computed as in section 2.

For all points on link '$n$' where $(l \le n < u)$the points shall be at different locations owing to actuation of joint $LJ_l$ . These can be computed by same procedure based on LASP and LAEP computations and the sensor locations estimation following (c.1) and method referred in Appendix B. We use ^ to identify these computed sensor location points.

Effective velocity in $\Pi_{\mathcal{M}}$ or parallel planes are denoted by

$\boldsymbol{EV\_\Pi_{\mathcal{M}}\_l\_u\_m\_(p,k)}$ and referred further as $\mathbb{V}_{um}(p,k)$

$$\mathbb{V}_{um}(p,k) = \frac{\mathbb{P}_{um}(p,k)}{\delta t}$$

The velocity of sensors applicable to the joint 'l' operation can be computed as following –

$$x\_\mathbb{P}_{um}(p,k) = \begin{array}{l} \left[\widehat{LJ}_u(x) + \widehat{SOJ_m}(x) + \frac{k}{K} * \left(\widehat{PBP_p}(x) - \widehat{PTP_p}(x)\right)\right]_{(t)} \\ -\left[LJ_u(x) + SOJ_m(x) + \frac{k}{K} * \left(PBP_p(x) - PTP_p(x)\right)\right]_{(t-\delta t)} \end{array}$$

$$y\_\mathbb{P}_{um}(p,k) = \begin{array}{l} \left[\widehat{LJ}_u(y) + \widehat{SOJ_m}(y) + \frac{k}{K} * \left(\widehat{PBP_p}(y) - \widehat{PTP_p}(y)\right)\right]_{(t)} \\ -\left[LJ_u(y) + SOJ_m(y) + \frac{k}{K} * \left(PBP_p(y) - PTP_p(y)\right)\right]_{(t-\delta t)} \end{array}$$

$$z\_\mathbb{P}_{um}(p,k) = \begin{array}{l} \left[\widehat{LJ}_u(z) + \widehat{SOJ_m}(z) + \frac{k}{K} * \left(\widehat{PBP_p}(z) - \widehat{PTP_p}(z)\right)\right]_{(t)} \\ -\left[LJ_u(z) + SOJ_m(z) + \frac{k}{K} * \left(PBP_p(z) - PTP_p(z)\right)\right]_{(t-\delta t)} \end{array}$$

(c.3)

Here quantities marked ^ are resulting from only joint $LJ_l$ actuation occurring at present *'t'* from the previous position at *'t-δt'* that resulted from all the 'n' joints that actuated at the previous time interval.

$$|\mathbb{P}_{um}(p,k)| = \sqrt{(x\_\mathbb{P}_{um}(p,k))^2 + (y\_\mathbb{P}_{um}(p,k))^2 + (z\_\mathbb{P}_{um}(p,k))^2} \tag{c.4}$$

$$x\_\mathbb{V}_{um}(p,k) = \left\{\begin{matrix} \left[\widehat{LJ}_u(x) + \widehat{SOJ_m}(x) + \frac{k}{K} * \left(\widehat{PBP}_p(x) - \widehat{PTP}_p(x)\right)\right]_{(t)} \\ -\left[LJ_u(x) + SOJ_m(x) + \frac{k}{K} * \left(PBP_p(x) - PTP_p(x)\right)\right]_{(t-\delta t)} \end{matrix}\right\} \Big/ \delta t$$

$$y\_\mathbb{V}_{um}(p,k) = \left\{\begin{matrix} \left[\widehat{LJ}_u(y) + \widehat{SOJ_m}(y) + \frac{k}{K} * \left(\widehat{PBP}_p(y) - \widehat{PTP}_p(y)\right)\right]_{(t)} \\ -\left[LJ_u(y) + SOJ_m(y) + \frac{k}{K} * \left(PBP_p(y) - PTP_p(y)\right)\right]_{(t-\delta t)} \end{matrix}\right\} \Big/ \delta t$$

$$z\_\mathbb{V}_{um}(p,k) = \left\{\begin{matrix} \left[\widehat{LJ}_u(z) + \widehat{SOJ_m}(z) + \frac{k}{K} * \left(\widehat{PBP}_p(z) - \widehat{PTP}_p(z)\right)\right]_{(t)} \\ -\left[LJ_u(z) + SOJ_m(z) + \frac{k}{K} * \left(PBP_p(z) - PTP_p(z)\right)\right]_{(t-\delta t)} \end{matrix}\right\} \Big/ \delta t \tag{c.5}$$

From this, the velocity of the sensor point $SS(p,k)$ on segment '$m$' of link '$u$' is

$$|\mathbb{V}_{um}(p,k)| = \sqrt{(x\_\mathbb{V}_{um}(p,k))^2 + (y\_\mathbb{V}_{um}(p,k))^2 + (z\_\mathbb{V}_{um}(p,k))^2} \tag{c.6}$$

Also

$$|\mathbb{V}_{um}(p,k)| = \frac{|\mathbb{D}_{uml}(p,k)|}{\delta t} \tag{c.7}$$

Drag effect owing to joint '$l$' operation on sensors on link '$u$' is computed as following.

Effective drag basis is denoted by

$\boldsymbol{EDB\_l\_u\_m\_(p,k)}$ and referred further as $\boldsymbol{\Omega}_{um}(p,k)$

$$\Omega_{um}(p,k) = \mu_e \cdot WVA\left(CSS_{p,k}(x), CSS_{p,k}(y), CSS_{p,k}(z)\right) \tag{c.8}$$

Torque on $LJ_l$ from a sensor $SS_{p,k}$ on segment $m$ of link $u$ is denoted as

$$\mathbb{T}_{lum}(p,k) = \mathbb{V}_{um}(p,k) * \Omega_{um}(p,k) * \mathbb{D}_{uml}(p,k) \tag{c.9}$$

the Drag integration may need some of the components depending on the joint configurations and is highly robot configuration dependent.

The torque arising from movement of joint *'l'* and sensed by all segments on $u^{\text{th.}}$ link is -

$$\mathbb{T}_{lu} = \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{k=1}^{K} \mathbb{V}_{um}(p,k) * \Omega_{um}(p,k) * \mathbb{D}_{uml}(p,k) \tag{c.10}$$

Range of $k$ is 1 to $K$, $p$ is from 1 to $P$ and that of $m$ is 1 to $M$.

The total drag on joint *'l' from* all sensors on links below joint $LJ_l$ is computed as following

$$\boldsymbol{B}_l = \sum_{u=l}^{N} \sum \mathbb{T}_{lu} \tag{c.11}$$

Range of $u$ is $l$ to $N$ ( $N$ is segment number of lowest child link).

**Drag computation for cases where Joint '*l*' undergoes translation motion.**

In cases, where the element $A_l\left(q_{l(t)}\right)$ in equation (c.2) is a translation, for all values of $n \geq l$, the $LASP(n)$ as well as $LAEP(n)$ undergo same translation motion. The motion vectors of all sensor points are parallel to each other and are of same magnitude. Therefore the computation of $|\mathbb{V}_{\boldsymbol{um}}(\boldsymbol{p},\boldsymbol{k})|$ in (c.6) needs to be done only once. We use *'V'* to represent single speed of translation. Only $CSS_{p,k}$ needs to be computed for evaluating $\boldsymbol{\Omega}_{um}(p,k)$ in (c.8) for sensors on links that have undergone translation. The drag direction is opposite to the translation vector. The force arising from movement on joint *'l'* and sensed by all segments on u$^{\text{th.}}$ link is –

$$F_{lu} = V * \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{k=1}^{K} \Omega_{um}(p,k) \tag{c.12}$$

Range of $k$ is 1 to $K$, $p$ is from 1 to $P$ and that of $m$ is 1 to $M$.

The total force on joint *'l'* from all sensors on links below is computed as following

$$F_l \ = \sum_{u=l}^{N} \sum F_{lu} \qquad \text{(c.13)}$$

Where, N is the lowest joints on the serial link type arm under consideration here.

**Drag computation optimisation for rotation of arm:**

Several configuration specific optimisation of computation of integrated sensor drag effects is possible. In articulated arms, vectors representing the rotation axis and the link axis can be in three distinct angular relationship, i; co-axial, ii; orthogonal and iii; at an angle other than these. We call case iii as 'spinning around a generalised rotation axis'.

The case (i) is depicted below and has been treated in chapter 5. The other two cases are treated here.

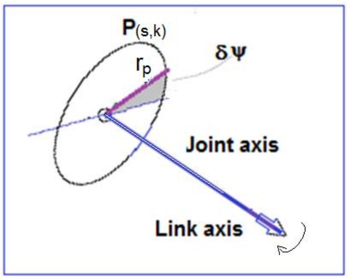



Figure C.2: Spinning Rotation –link axis and joint axis coincide.

**Drag contribution by sensor spinning around a generalised rotation axis:**

For cases where Joint '$l$' undergoes rotation motion (spin around link axis) the element $A_l(q_l)$ in equation (c.2) is rotation. The sensors $SS_{p,k}$ on links on lower joints below the immediate child link i.e. joints '$l + 1$' onwards can be treated in a generic manner .

The rotation axis $\zeta$ of joint '$l$' is a vector defined by the robot kinematics model

[40] and the joint parameters of all higher joints affecting the position of joint '$l$'. In 3D Cartesian space it is a line defined by $\overrightarrow{P_1P_2}$ on the joint axis. For rotation around this axis, the torques caused by the drag sensed by $SS_{p,k}$, here represented by $P_i$, needs computation of $P_i'$, which is the nearest point to $P_i$ on vector $\overrightarrow{A_1A_2}$ or its extension. The length of $P_iP_i'$ is 'torque exerting length'(figure C.4b).

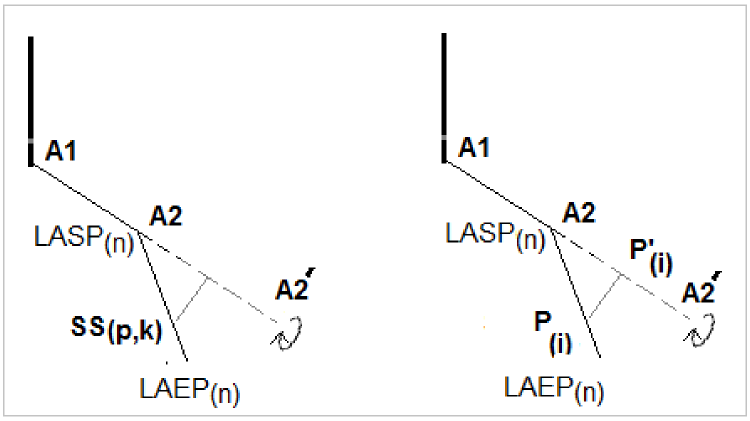



Figure C.3: sensor spinning around a generalised rotation axis

**Optimization in this computation includes**

i. Reuse of normal vector of plane $\Pi_k$ ( rotation plane of sensor point $SS_{p,k}$ )or the direction vector of $P_iP_i'$ for all $SS_{p,k}$

ii. The sensors $SS_{p,k}$ are structured . For groups of sensors on a single link a $sin(\xi)$ , where $\xi$ is the angle between vector $\overrightarrow{A_1A_2}$ and vector $\overrightarrow{LASP(n)LAEP(n)}$. $\overrightarrow{A_1A_2}$ can be used for computing $P_iP_i'$ corresponding to $SS_{p,k}$ on link defined by the axis $LASP(n)$ and $LAEP\ (n)$

For case (figure C.4(b) ) drag is computed as in relation 5-32 . $A_2'$ is extension of vector $\overrightarrow{A_1A_2}$ and so has same direction coefficients as $\overrightarrow{A_1A_2}$ . $DSS_{p,k}$ for point $P_i$

is vector normal to vector $\overrightarrow{A_1A_2}$ denoted as $P_iP_i'$ .

$$DSS_{p,k} = |\overrightarrow{P_iP_i'}| = \sin(\xi_1)\frac{k}{K} * L_n \tag{c. 14}$$

$L_n$ is length of $n^{th}$ link with end points $LASP(n)$ and $LAEP(n)$. $\frac{k}{K}$ is defined in chapter 5.

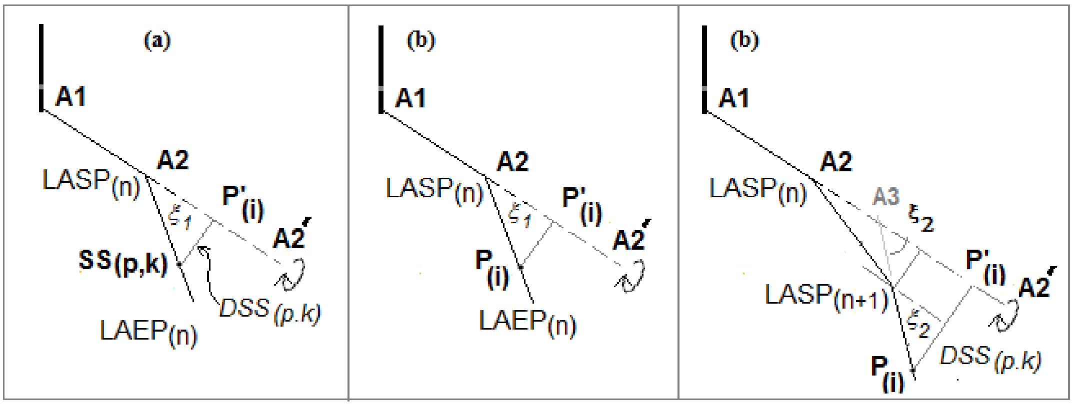


Figure C.4: Spinning rotation of link around a predecessor link axis.

In case of the generalised approach for $SS(p,k)$ i.e. points $P_i$ on next lower link $\overrightarrow{LASP(n+1)LAEP(n+1)}$ . Intersection of $\overrightarrow{LASP(n+1)LAEP(n+1)}$ with $\overrightarrow{A_1A_2}$ gives point $A_3$ and angle $\xi_2$

$$DSS_{p,k} = |\overrightarrow{A_3P_i'}| * \sin(\xi_2) \tag{c. 15}$$

In incremental computation from the joint *'1'* downward, it is beneficial to use previous computation as it reduces computation as shown below

For points $P_i$ on next lower link $\overrightarrow{LASP(n+1)LAEP(n+1)}$ which makes angle $\xi_2$ with the rotation vector $\overrightarrow{A_1A_2}$'

$$DSS_{p,k} = \sin(\xi_1) * L_n + \sin(\xi_2)\frac{k}{K} * L_{n+1} \tag{c.16}$$

Note that $L_n$ are persisting constatnt as they are robot design parameters. In equation above, the computation of $A_3$ is not required.

**Very Effective Optimisation:**

The structuring of sensors (chapter 5 ) is instrumental in optimisation of computation in very effective manner. For shell sensors $SS_{p,k}$ on a single '$p$' sensor at $k = 0$ and that at $k = K$ are at $PTP$ and $PBP$ rings , i.e. analogous to $LASP$ and $LAEP$. Computation of $P_i P_i'$ can be done by using the following method of computing minimum distance of $CSS_{p,k}$ from vector $A_1 A_2$

i. treat $A_1 A_2$ as normal vector to a plane $\Pi_H$ having point $P_i$ on it
ii. use $x, y, z$ coordinates of $CSS_{p,k}$ at $k = 1$ to find equation of plane $\Pi_H$
iii. Find intersection point $\boldsymbol{I}_1$ of vector $A_1 A_2$with this plane $\Pi_H$
iv. Distance $CSS_{p,k}$ to $\boldsymbol{I}_1$ is $P_1 P_1'$
v. use $x, y, z$ coordinates of $CSS_{p,k}$ at $k = K$ to find plane $\Pi_{HK}$
vi. Find intersection point $\boldsymbol{I}_K$ of vector $A_1 A_2$ with this plane $\Pi_{HK}$
vii. Distance $CSS_{p,k}$ to $\boldsymbol{I}_K$ is $P_K P_K'$
viii. Interpolate linearly between $P_1 P_1'$ to $P_K P_K'$ for finding $P_k P_k'$ as following

$$P_k P_k' = \mathrm{P_1 P_1'} + \frac{\mathrm{k}}{\mathrm{K}} (\mathrm{P_K P_K'} - \mathrm{P_1 P_1'}) \qquad \text{(c.17)}$$

**Link rotation around joint location '$A$' with joint axis orthogonal to Link axis-**

It is common to find link rotation around joint location '$A$' with joint axis orthogonal to Link axis in plane $\Pi_{\mathrm{v}}$ , the vertical plane containing links $n-1, n$ and

$n + 1$. Therefore the distance of a sensor $SS_{p,k}$ from the rotation axis is same as that of the plane of sensor ring $\Pi_s$ of the link at a given '$k$'.

In this approach Link rotation is around joint location '$A$' with joint axis orthogonal to Link axis, the vertical plane $\Pi_v$ containing links $n-1, n$ and $n+1$.

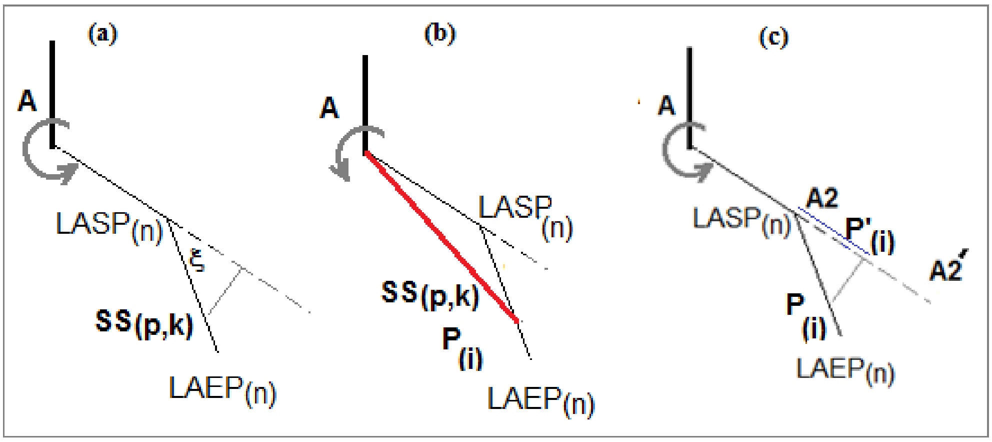



Figure C.5: Link axis orthogonal to joint axis

In a generalised approach one can find $DSS_{p,k}$ defined in chapter 5 as indicated in figure (C.5) by using Cartesian position of $A$ as $CA(x), CA(y), CA(z)$. For slender shaped long arms, if distance of $LASP(n)$and $LAEP(n)$ from '$A$' is >> .'$r$' of the sensor ring, then computation can use interpolated position on link axis as the Cartesian position of $SS_{p,k}$. Here $CA(x), CA(y), CA(z)$ are the coordinates of point of intersection between $\Pi_v$ and the joint axis $A_1A_2$.

$$DSS_{p,k} = \sqrt{\left(CASS_{p,k}(x) - CA(x)\right)^2 + \left(CASS_{p,k}(y) - CA(y)\right)^2 + \left(CASS_{p,k}(z) - CA(z)\right)^2}$$

(c.18)

In above equation $CASS_{p,k}$ is the sensor ring's plane position on axis of the host link and expressed in Cartesian form by using similar method as in chapter 5.

**Computation optimisation for base joint:**

For the base joint , the computations can follow the case of generalised rotation around a general axis of rotation in Cartesian space (5-34) relation, the feature that the rotation axis is always vertical , leads to simplification. Note that this is special condition of case where a spinning rotation of link around a predecessor link axis occurs but the rotation axis is vertical and not in general orientation.

By using projection of point $SS_{p,k}$ on horizontal base plane , which can be effected simply by using $CSS_{p,k}(x)$ and $CSS_{p,k}(y)$ coordinates only and discarding '$z$' coordinate and similarly using Cartesian position $CA2'(x)$ and $CA2'(y)$ as projection of vector $A_1A_2$ on horizontal plane. Here, only two terms being used, computation is less.

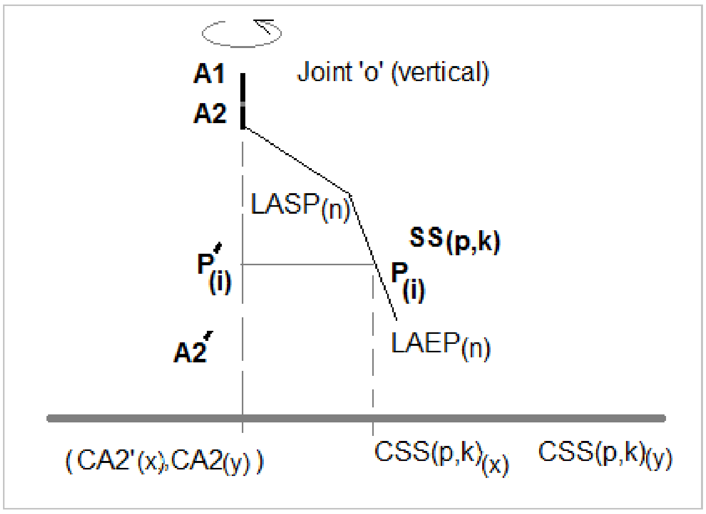



Figure C.6: Lean computation of $DSS_{p,k}$ by projection on floor plane of $WVA$.

$$DSS_{p,k} = \sqrt{\left(CSS_{p,k}(x) - CA2'(x)\right)^2 + \left(CSS_{p,k}(y) - CA2'(y)\right)^2} \quad \text{(c.19)}$$

A very important point to note is that in all rotational torque computations, $DSS^2_{p,k}$ is used and therefore the eucledean distance measured from point '$A$' in cases above, the square route calculation is not required hence more optimum computation is

$$DSS^2_{p,k} = \left(CSS_{p,k}(x) - CA2'(x)\right)^2 + \left(CSS_{p,k}(y) - CA2'(y)\right)^2 \quad \text{(c.20)}$$

Further the $DSS^2_{p,k}$ computation needs to be done for two ends and interpolation can be used similar to that by equation c.16.

$$DSS^2_{p,k} = \left(CSS_{p,k}(x) - CA(x)\right)^2 + \left(CSS_{p,k}(y) - CA(y)\right)^2 + \left(CSS_{p,k}(z) - CA(z)\right)^2$$

(c.21)

**Computation of trigonometric values for ξ**

$$Cos\ \xi = \frac{\overrightarrow{LASP(n)LAEP(n)} \cdot \overrightarrow{A_1A_2}}{\left|\overrightarrow{LASP(n)LAEP(n)}\right| \left|\overrightarrow{A_1A_2}\right|}$$

(c.22)

$$sin\xi = \sqrt{1 - \cos^2 \xi} \quad \text{(c.23)}$$

# Appendix D

# Equipments Used In Experiments

**1. CARTESIAN ROBOT**

It is a 5-DOF robot with three linear motions and two rotary motions. Linear motions are used for positioning the probe, while rotary motions are useful in orienting the probe at required position, generally normal to the test specimen. The system consists of five axes in all, where three axes are actuated by ac servomotor drives and remaining two by dc stepper motor drives. This is operated in position control mode and executes the position command issued by host controller (Computer/PLC). The system uses a real-time loop where the embedded controller controls the axes in accordance to the commands sent by host controller (PC) and the feedback received from the drives of each axes. Host controller gives position commands to the embedded controller via Ethernet. Hence the system can also be configured as a remote managed system with wireless communication between the host and embedded controller. In the system, X-, Y-, and Z-motions are linear motions white $\theta$ and $\Phi$ are rotary motions. The axes parameters are given in table 1.

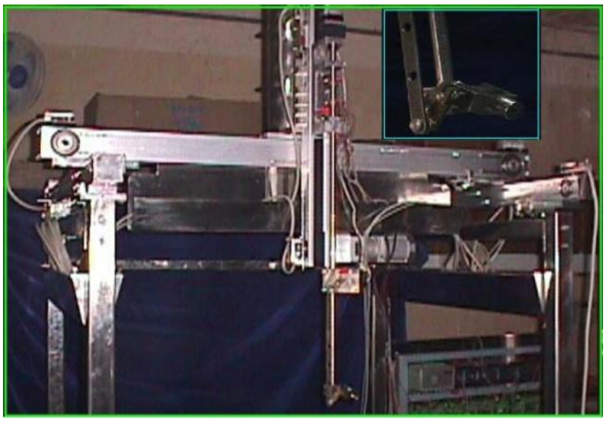


Figure D.1: 5 DOF Cartesian Robot with probe orientation stage (inset)

**Table D.1:** Axes parameters for 5 DOF probe robot.

| AXES | REPEATABILITY | RANGE | MAX. SPEED |
|---|---|---|---|
| X | 0.1mm | 72cm | 24cm/s |
| Y | 0.1mm | 67.6cm | 15.6cm/s |
| Z | 0.1mm | 44.2cm | 15.6cm/s |
| θ | 3 arc min | $-75^o$ to $180^o$ | $50^o$/s |
| Φ | 3 arc min | $0^o$ to $90^o$ | $50^o$/s |

Details of AC servo-motor and drives are as given below:

| Motor Specifications: | | |
|---|---|---|
| Make | : | Panasonic |
| Model No | : | MSMD082P1U |
| Input | : | 3ΦAC, 120V, 4A |
| Rated Output | : | 0.75kW |
| Rated Frequency | : | 200Hz |
| Rated Speed | : | 3000RPM |
| Cont. Torque | : | 2.4N-m |
| Torque constant | : | 0.6N-m/A |
| Connection | : | Star |

| Driver Specifications: | | | |
|---|---|---|---|
| Make | : | Panasonic | |
| Model | : | MCDDT3520 | |
| | | INPUT | OUTPUT |
| Voltage | : | 200-240V | 126.9V |
| Phase | : | 1Φ/3Φ | 3Φ |
| Full Load current | : | 6A/3.3A | 4.0A |
| Frequency | | 50/60Hz | 0~333.3Hz |
| Power | : | | 750W |

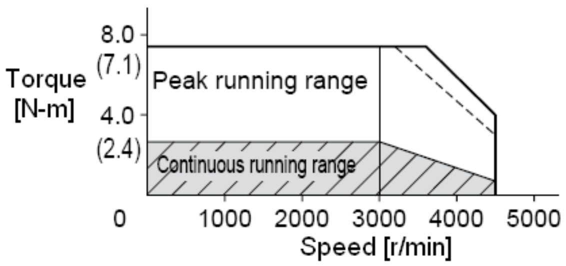



Figure D.2: Speed-Torque Characteristics

Gear Box: 1:5 (Low Backlash, High efficiency type) Model No: VRSF-LB-5C-750

## 2. GANTRY MOUNTED SUSPENDED ARM ROBOT

The system consists of a Top-down articulated robot arm mounted on a rail guided overhead gantry. The system consists of a total of 7DOFs. Gantry motion is linear position controlled servo and in experiments here (chapter 5) used as locator of robot base only. Revolute axes are operated by 400W and 100W AC servo-systems. The system is operated in position control mode .

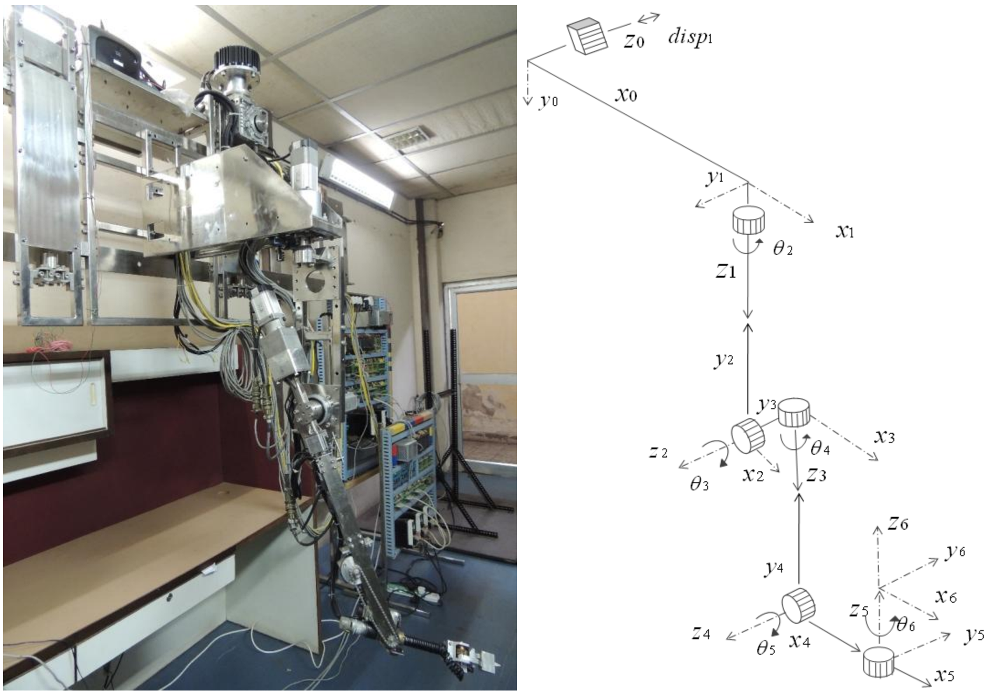



Figure D.3: Gantry based Suspended arm robot and its DH representation

**Table D.2:** Axes parameters for Gantry based Suspended Arm Robot.

| AXES | REPEATABILITY | RANGE | MAX. SPEED |
|---|---|---|---|
| Disp1 | 1mm | 180cm | 2cm/s |
| Theta2 | 3 arc min | $-180^o$ to $180^o$ | $10^o$/s |
| Theta3 | 3 arc min | $-60^o$ to $60^o$ | $10^o$/s |
| Theta4 | 3 arc min | $-90^o$ to $90^o$ | $10^o$/s |
| Theta5 | 3 arc min | $-45^o$ to $60^o$ | $10^o$/s |
| Theta6 | 10 arc min | $-180^o$ to $180^o$ | $20^o$/s |

100W Servo-system

| Motor Specifications: | | |
|---|---|---|
| Make | : | Panasonic |
| Model No | : | MSMD012P1T |
| Input | : | 3ΦAC, 73V, 1.1A |
| Rated Output | : | 0.1kW |
| Rated Frequency | : | 200Hz |
| Rated Speed | : | 3000RPM |
| Cont. Torque | : | 0.32 N-m |
| Torque constant | : | 0.3 N-m/A |
| Connection | : | Star |

| Driver Specifications: | | | |
|---|---|---|---|
| Make | : | Panasonic | |
| Model | : | MADDT1205 | |
| | | INPUT | OUTPUT |
| Voltage | : | 200-240V | 69.4V |
| Phase | : | 1Φ | 3Φ |
| Full Load current | : | 1.3A | 1.2A |
| Frequency | | 50/60Hz | 0~333.3Hz |
| Power | : | | 100W |

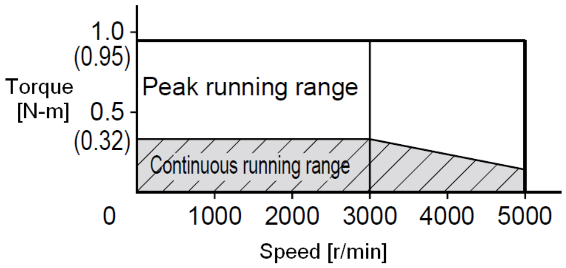



Figure D.4: Speed-Torque Characteristics

400W Servo-system

| Motor Specifications: | | |
|---|---|---|
| Make | : | Panasonic |
| Model No | : | MSMD042P1U |
| Input | : | 3ΦAC, 105V, 2.6A |
| Rated Output | : | 0.4kW |
| Rated Frequency | : | 200Hz |
| Rated Speed | : | 3000RPM |
| Cont. Torque | : | 1.3 N-m |
| Torque constant | : | 0.5N-m/A |
| Connection | : | Star |

| Driver Specifications: | | | |
|---|---|---|---|
| Make | : | Panasonic | |
| Model | : | MBDDT2210 | |
| | | INPUT | OUTPUT |
| Voltage | : | 200-240V | 107.9V |
| Phase | : | 1Φ | 3Φ |
| Full Load current | : | 3.7A | 2.6A |
| Frequency | | 50/60Hz | 0~333.3Hz |
| Power | : | | 400W |

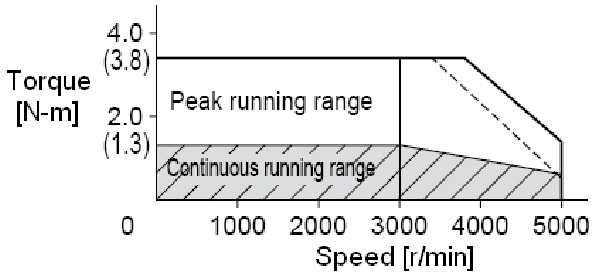



Figure D.5: Speed-Torque Characteristics

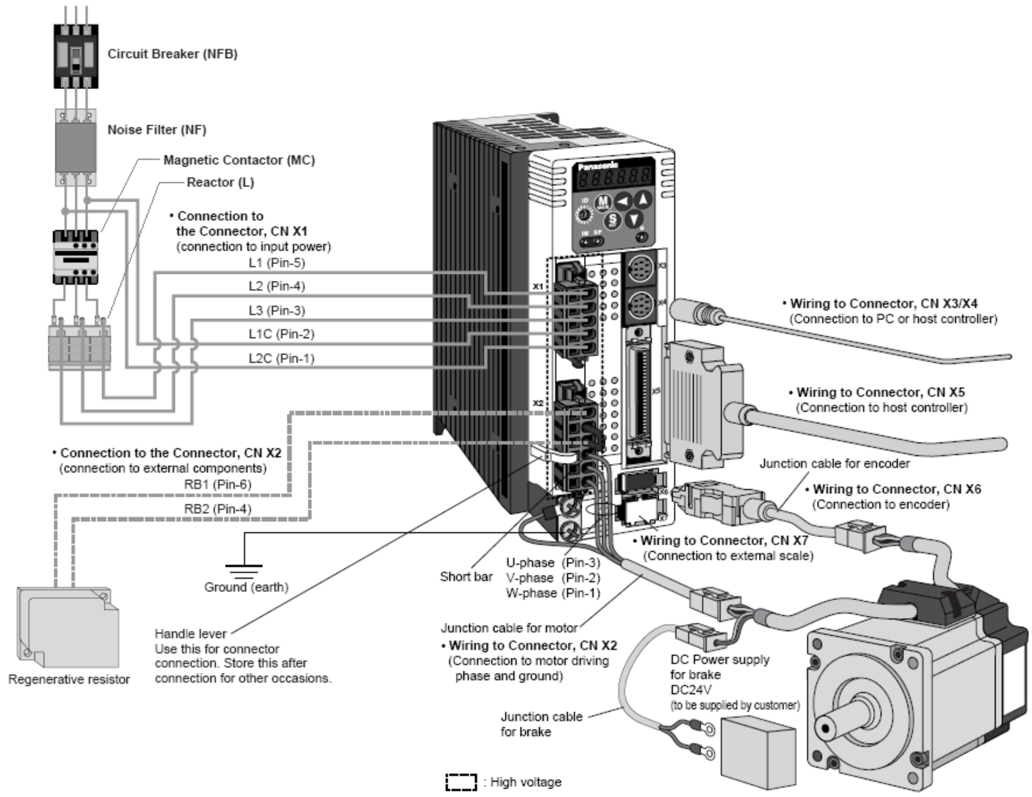



Figure D.6: Wiring example for AC Servo-motor

## 3. ACTIVE JOYSTICK SYSTEM AJS

Joystick consists of 2-DOF (viz. θ and Φ) system which is useful as master side controlling agent in various master-slave experiments. Both the axes in joystick are actuated by AC servo-systems which are tracked in real time to obtain its orientation. Servo-systems for the axes are configured in torque control mode for interactive force feedback operation. The joystick incorporates real-time joint/orientation tracking control system, has user configurable torque-setting, program configurable force feedback. It has a span of $\pm 150^{o}$ θ and $\phi$-$180^{o}$ to $90^{o}$ .

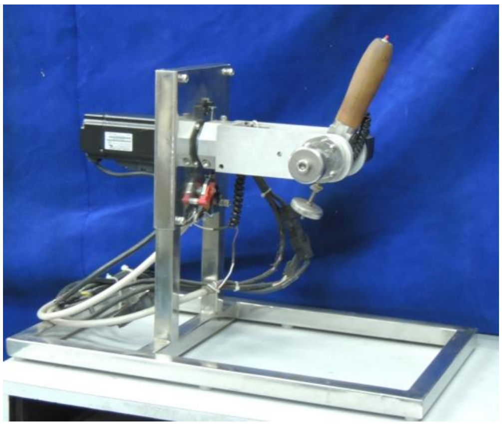


Figure D.7: 2-DOF Interactive Joystick

100W Servo-system

| Motor Specifications | | |
|---|---|---|
| Make | : | GE-Fanuc VersaMotion |
| Model No | : | IC800VMM01LN KSE25A |
| Input | : | 3ΦAC, 110V, 0.9A |
| Rated Output | : | 0.1 kW |
| Rated Speed | : | 3000RPM |
| Cont. Torque | : | 0.32 N-m |
| Torque constant | : | 0.36N-m/A |

| Driver Specifications | | | |
|---|---|---|---|
| Make | : | GE-Fanuc VersaMotion | |
| Model | : | IC800VMA012-AA | |
| | | INPUT | OUTPUT |
| Voltage | : | 200-230V | 110V |
| Phase | : | 1Φ/3 Φ | 3Φ |
| Full Load current | : | 1.0A/0.8A | 1.1A |
| Frequency | | 50/60Hz | 0~200 Hz |
| Power | : | | 100 W |

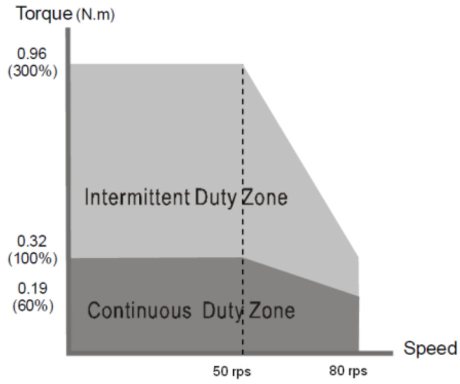



Figure D.8: Speed-Torque Characteristics

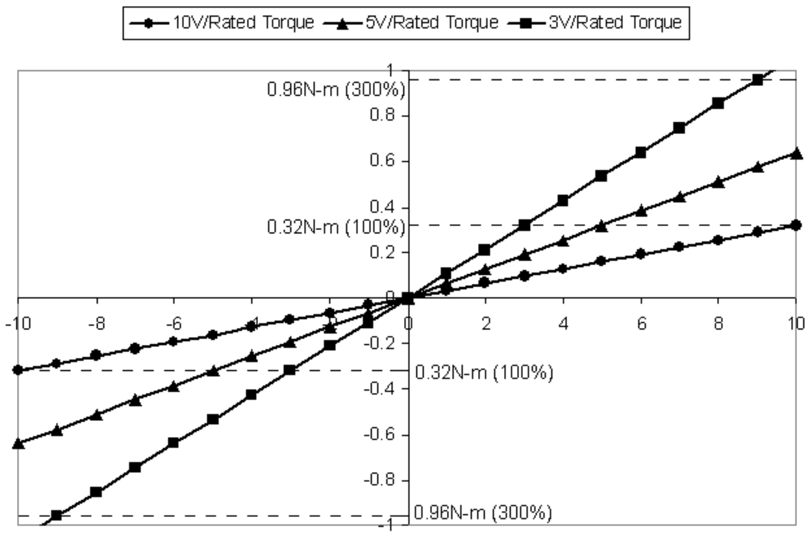



Figure D.9: Command Voltage-Torque Characteristics (Torque Control Mode)

400W Servo-system

| Motor Specifications | | |
|---|---|---|
| Make | : | GE-Fanuc VersaMotion |
| Model No | : | IC800VMM04LB KSE25A |
| Input | : | 3ΦAC, 110V, 2.6A |
| Rated Output | : | 0.4 kW |
| Rated Speed | : | 3000RPM |
| Cont. Torque | : | 1.27 N-m |
| Torque constant | : | 0.49 N-m/A |
| Holding Brake | : | Yes (24V supply) |

| Driver Specifications | | | |
|---|---|---|---|
| Make | : | GE-Fanuc VersaMotion | |
| Model | : | IC800VMA042-AA | |
| | | INPUT | OUTPUT |
| Voltage | : | 200-230V | 110V |
| Phase | : | 1Φ/3 Φ | 3Φ |
| Full Load current | : | 3.4A/2.6A | 3.3A |
| Frequency | | 50/60Hz | 0~200 Hz |
| Power | : | | 400 W |

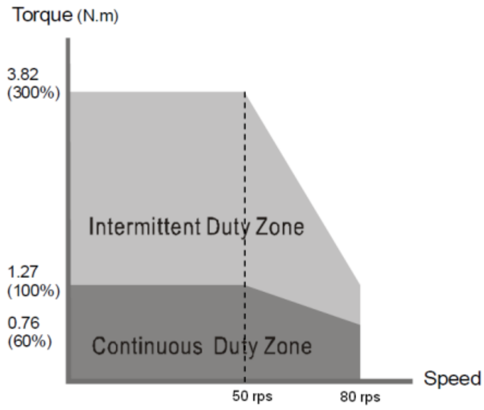



Figure D.10: Speed-Torque Characteristics

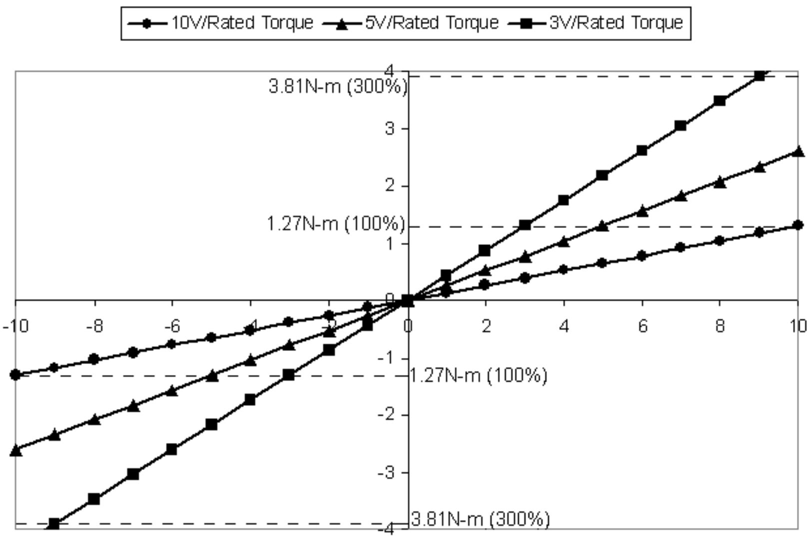



Figure D.11: Command Voltage-Torque Characteristics (Torque Control Mode)

**SERVO SYSTEMS :**

**PANASONIC MINAS A-4 Series AC Servo-Systems**

The Panasonic A4 series, digital AC servo drives have a slim profile/power ratio, carry global certification such as UL and CE, and cover a power range from 50W to 5.0kW. Along with a velocity response (bandwidth) of 1 kHz, the drives offer a high-functionality real-time auto-gain tuning, to react to even variations of load inertia. The real-time auto-gain tuning gives the user the ability to select the stiffness of the machines from low to high with a combination of an adaptive filter, 2-channel notch filters, damping control and supports vertical axis applications. Setup can be performed by keypad. With ‘Position’, ‘Velocity’ and ‘Torque’ command control modes, the A4 series are truly multi-functional. In ‘Position’ control mode, the drive accepts up to 2Mpps of pulse input and can be used with PLC for outstanding performances. In ‘Velocity’ and ‘Torque’ control modes, the drive accepts ±10V and produces corresponding control commands.

The A4 series motors have metric mounting flanges and shafts dimensions, and are capable to produce speed up to 5,000 rpm. IP65 rating is standard on all the motors and any motor can be ordered with an optional oil seal and holding brake. The use of serial encoders reduces the numbers of wire from 11 to 5 for the standard incremental encoder (2500 pulse/rev) and from 14 to 7 for the optional 17-bit absolute encoder

**Operation of Servo-systems in *Position Control Mode***

In position control mode, pulse train is applied to pins 4 and 6 of connector CN-X5 of servo-driver. In addition to pins 4 and 6, the driver has a provision for high speed differential pulse input feature which is available on pins 44 to 47. Position and speed commands are decoded by the position/speed decoder of the driver. Position is fed to the position comparator and speed is fed forward to the speed comparator. Based on position error and speed command, driver will generate torque command to drive the motor.

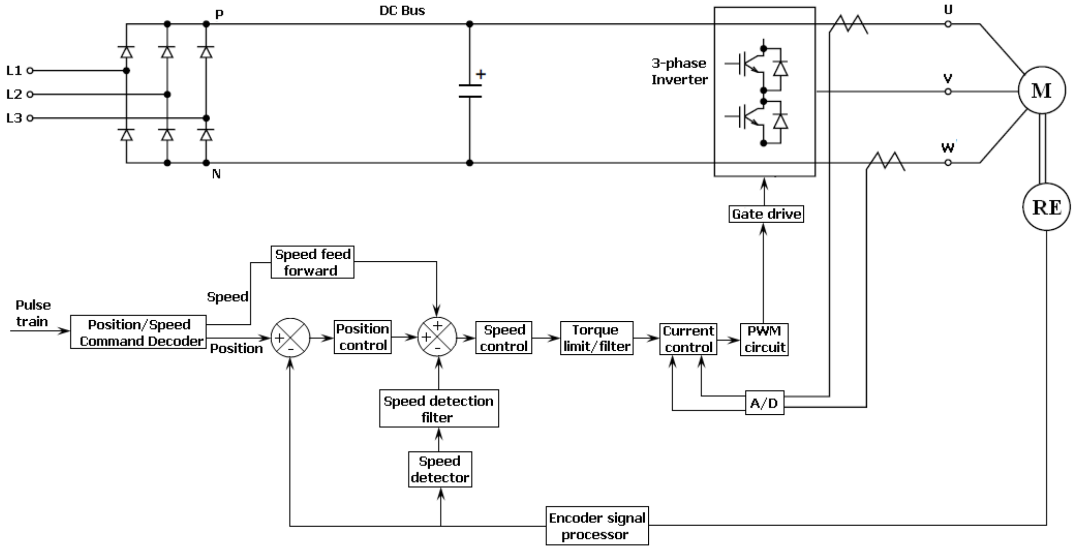


Figure D.12: Block diagram of Servo-driver in Position Control Mode

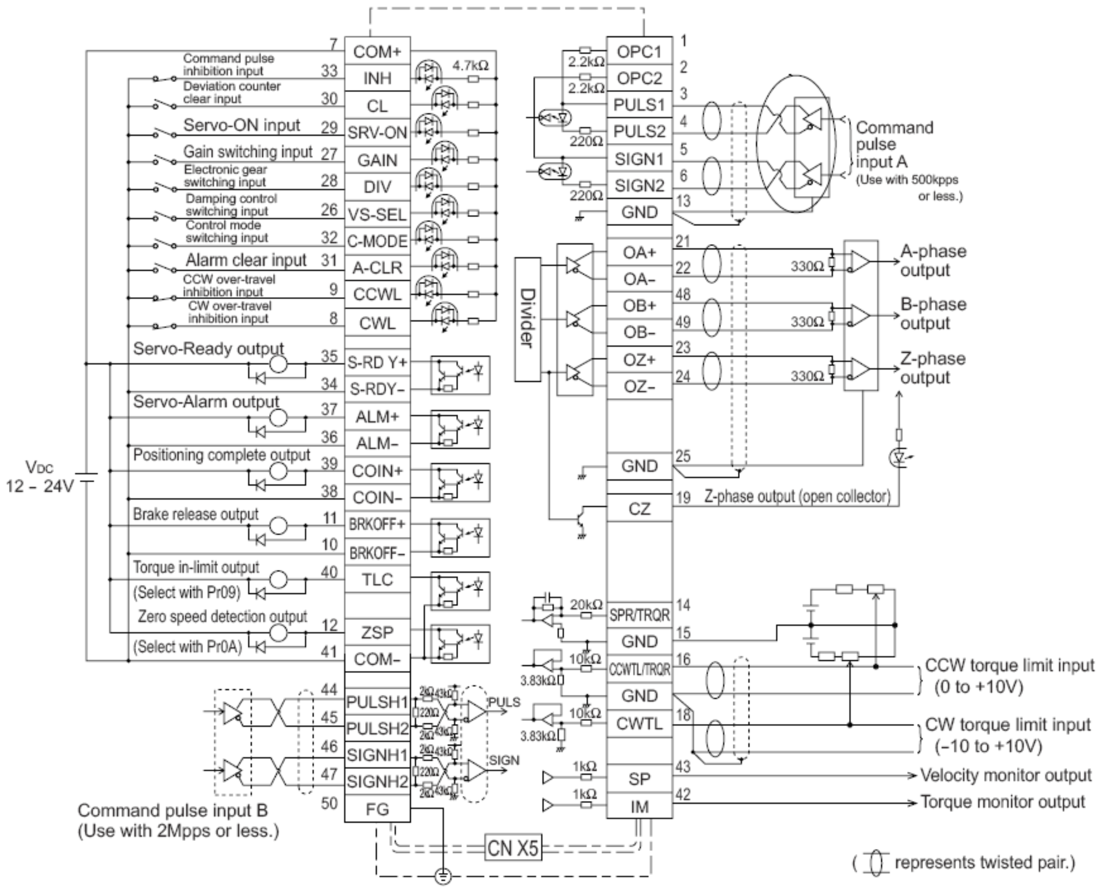


Figure D.13: Various input and output signals for Position Control Mode

**Operation of Servo-systems in *Speed Control Mode***

In speed control mode, analog voltage signal (-10V to 10V) is applied to pin 14 of connector CN-X5 of servo-driver. The driver offers a provision to have input in range of 0.0005V/rpm to 0.1V/rpm. The voltage is fed to 16 bit A/D converter in the driver and speed command is generated using the gain settings. Speed command is fed to speed control loop and corresponding torque command is generated to drive the motor.

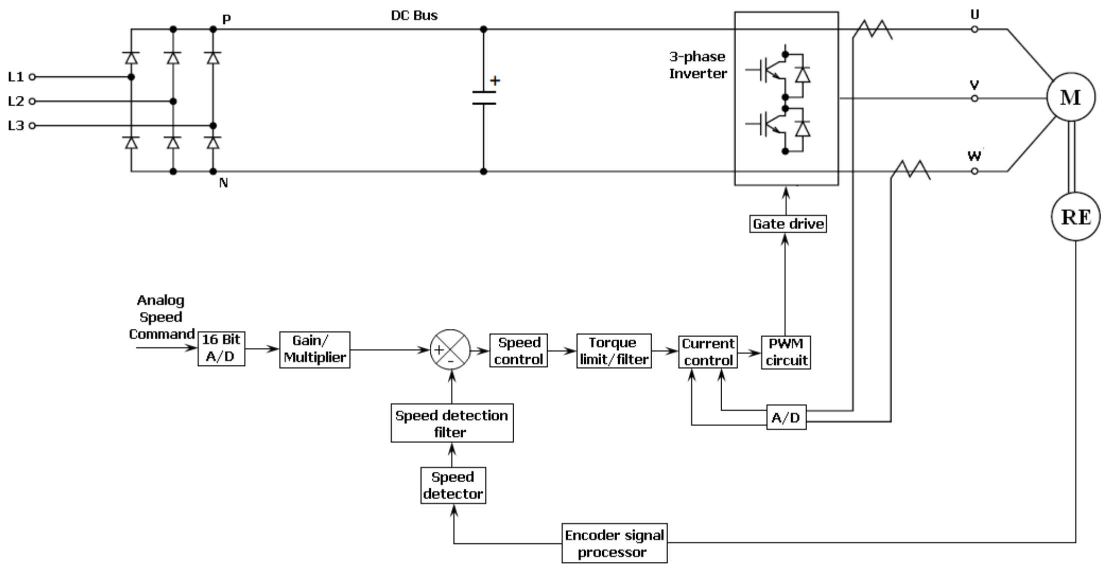



Figure D.14: Block diagram of Servo-driver in Speed Control Mode

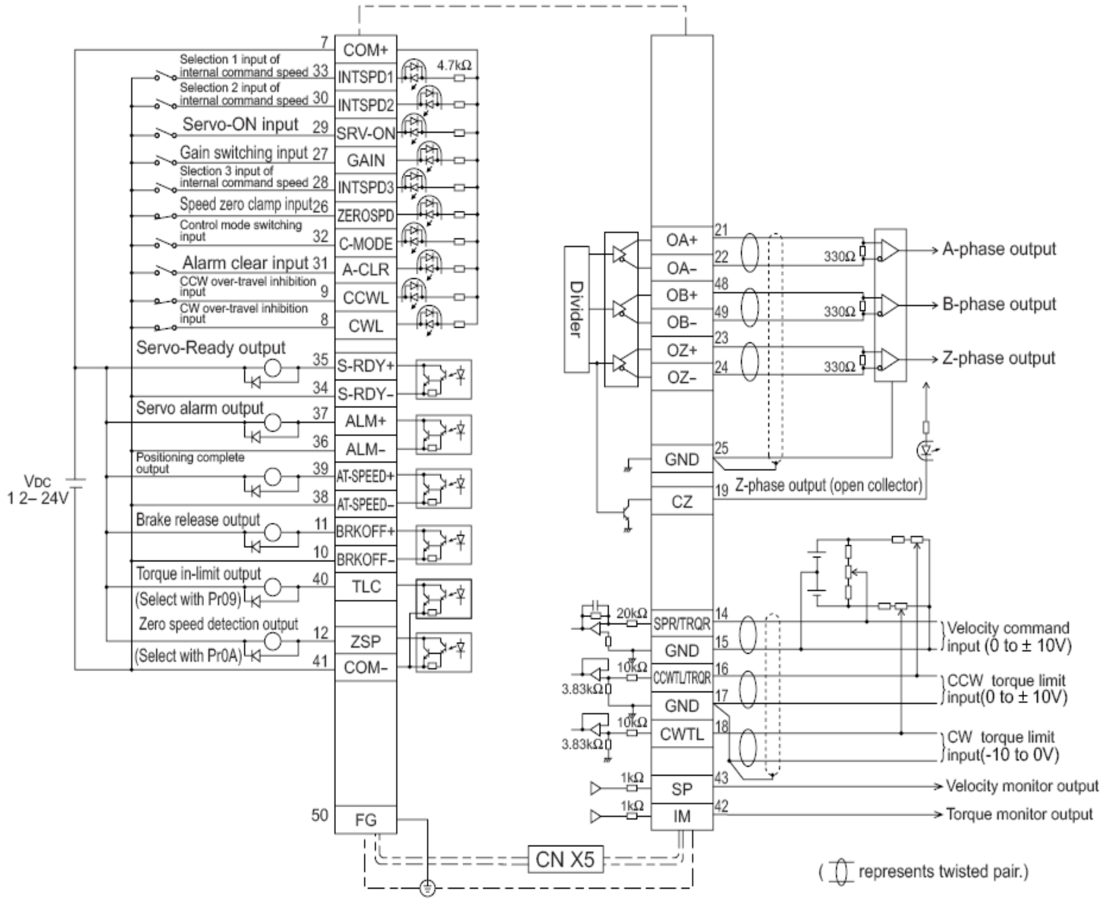



Figure D.15: Various input and output signals for Speed Control Mode

### Operation of Servo-systems in *Torque Control Mode*

In torque control mode, analog voltage signal (-10V to 10V) is applied to pin 14 of connector CN-X5 of servo-driver. The driver offers a provision to have input in range of 1V to 10V for rated torque. The voltage is fed to 16 bit A/D converter in the driver and torque command is generated using the gain settings. Maximum speed is internally selected through a user programmable register. Function of the speed loop is to limit the speed to maximum speed level set by the user. Torque command is fed forward to the torque controller and based on the torque command, the driver actuates the motor.

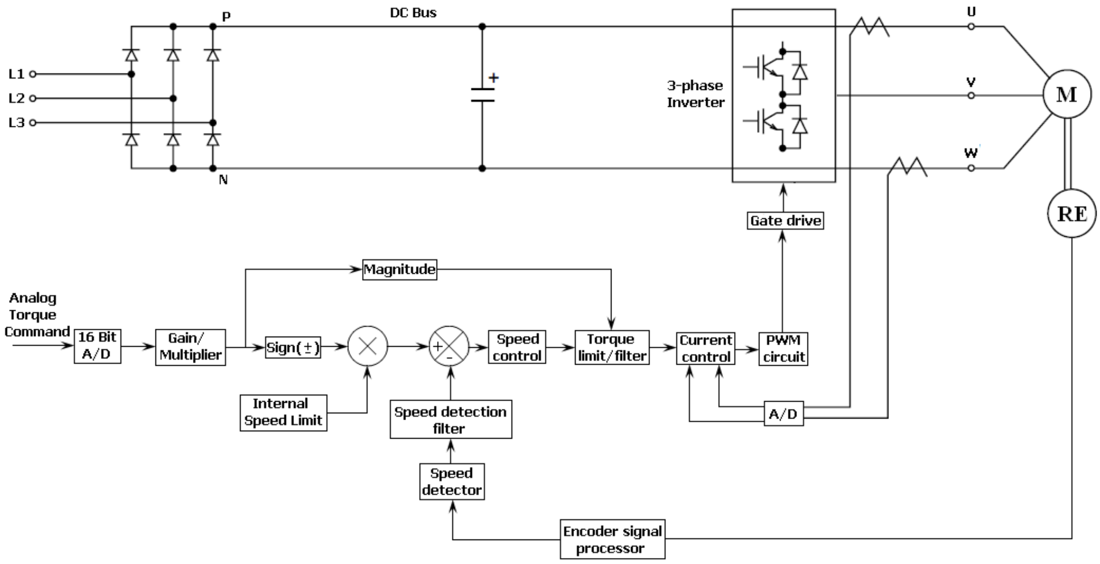



Figure D.16: Block diagram of Servo-driver in Torque Control Mode

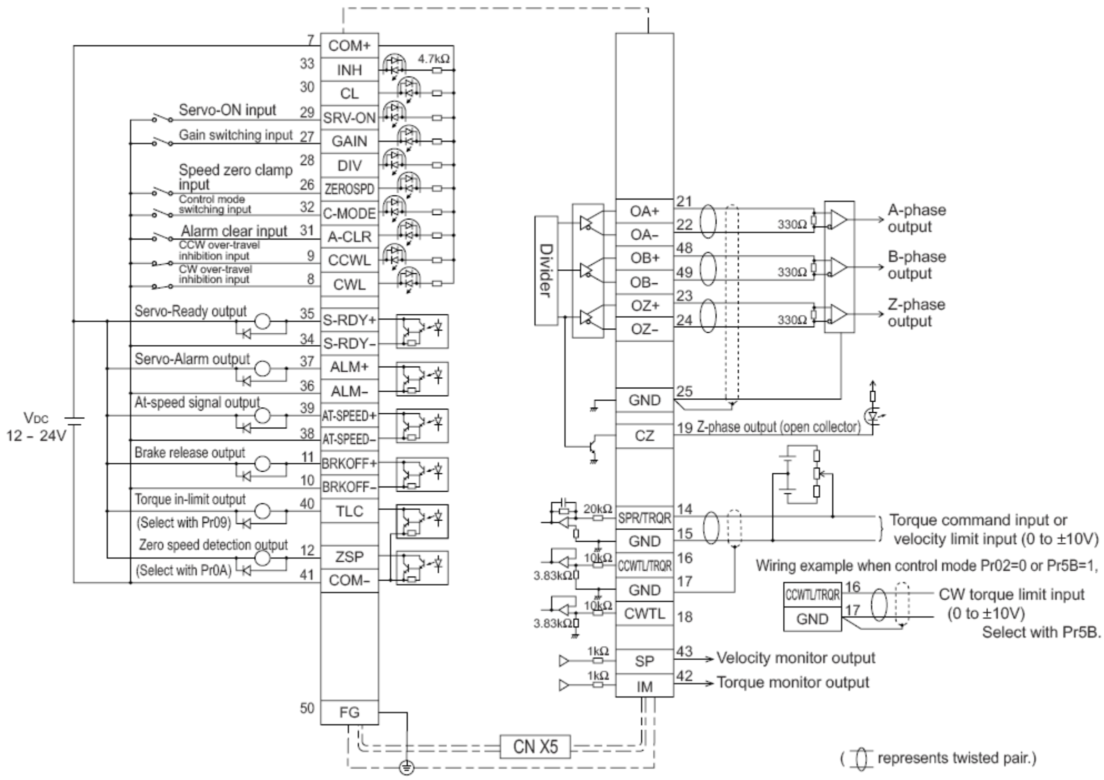



Figure D.17: Various input and output signals for Torque Control Mode

# References

[1] T. B. Sheridan. Telerobotics, Automation, and Human Supervisory Control. The MIT Press, Cambridge, Massachusetts, 1992.

[2] G.R.Hopkinson, A. Mohammadzadeh, “Radiation Effects In (CCD) Imagers& CMOS Active Pixel Sensors” J. High Speed Elect and Sys., Vol.14, Issue 2, June 04, ISSN: 0129-1564

[3] Hannaford, Blake. "A design framework for teleoperators with kinesthetic feedback." Robotics and Automation, IEEE Transactions on 5, no. 4 (1989): 426-434.

[4] Nash, Eric B., Gregory W. Edwards, Jennifer A. Thompson, and Woodrow Barfield. "A review of presence and performance in virtual environments." International Journal of human-computer Interaction 12, no. 1 (2000): 1-41.

[5] Welch, Robert B. "How can we determine if the sense of presence affects task performance?" Presence: Teleoperators and Virtual Environments 8, no. 5 (1999): 574-577.

[6] Sallnäs, Eva-Lotta, Kirsten Rassmus-Gröhn, and Calle Sjöström. "Supporting presence in collaborative environments by haptic force feedback." ACM Transactions on Computer-Human Interaction (TOCHI) 7, no. 4 (2000): 461-476.

[7] Whitney, Daniel E. "Historical perspective and state of the art in robot force control." The International Journal of Robotics Research 6, no. 1 (1987): 3-14.

[8] Hogan, Neville. "Stable execution of contact tasks using impedance control." In Robotics and Automation. Proceedings. 1987 IEEE International Conference on, vol. 4, pp. 1047-1054. IEEE, 1987.

[9] Hannaford, Blake, and Robert Anderson. "Experimental and simulation studies of hard contact in force reflecting teleoperation." In Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on, pp. 584-589. IEEE, 1988.

[10] Everett, Steven Edward, and Rajiv V. Dubey. "Human-machine cooperative telerobotics using uncertain sensor or model data." In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, vol. 2, pp. 1615-1622. IEEE, 1998.

[11] Weber, Carolina, Verena Nitsch, Ulrich Unterhinninghofen, Berthold Farber, and Martin Buss. "Position and force augmentation in a telepresence system and their effects on perceived realism." In EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint, pp. 226-231. IEEE, 2009.

[12] Dubey R, Everett S, Pernalete N, Manocha K. Teleoperation assistance through variable velocity mapping. IEEE Trans Robot Autom 2001;17:761–6.

[13] Manocha, Karan A., Norali Pernalete, and Rajiv V. Dubey. "Variable position mapping based assistance in teleoperation for nuclear cleanup." In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, vol. 1, pp. 374-379. IEEE, 2001.

[14] Everett, Steven E., and Rajiv V. Dubey. "Model-based variable position mapping for telerobotic assistance in a cylindrical environment." In Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, vol. 3, pp. 2197-2202. IEEE, 1999.

[15] Bettini, Alessandro, Panadda Marayong, Samuel Lang, Allison M. Okamura, and Gregory D. Hager. "Vision-assisted control for manipulation using virtual fixtures." Robotics, IEEE Transactions on 20, no. 6 (2004): 953-966.

[16] Rosenberg, Louis B. "Virtual fixtures: Perceptual tools for telerobotic manipulation." In Virtual Reality Annual International Symposium, 1993., 1993 IEEE, pp. 76-82. IEEE, 1993.

[17] Park, Shinsuk, Robert D. Howe, and David F. Torchiana. "Virtual fixtures for robotic cardiac surgery." In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2001, pp. 1419-1420. Springer Berlin Heidelberg, 2001.

[18] Abbott, Jake J., Panadda Marayong, and Allison M. Okamura. "Haptic virtual fixtures for robot-assisted manipulation." In Robotics research, pp. 49-64. Springer Berlin Heidelberg, 2007.

[19] Pezzementi, Zachary, Allison M. Okamura, and Gregory D. Hager. "Dynamic guidance with pseudoadmittance virtual fixtures." In Robotics and Automation, 2007 IEEE International Conference on, pp. 1761-1767. IEEE, 2007.

[20] Warren, Charles W. "Global path planning using artificial potential fields." In Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on, pp. 316-321. IEEE, 1989.

[21] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." The international journal of robotics research 5, no. 1 (1986): 90-98.

[22] Volpe, Richard, and Pradeep Khosla. "Manipulator control with superquadric artificial potential functions: Theory and experiments." Systems, Man and Cybernetics, IEEE Transactions on 20, no. 6 (1990): 1423-1436.

[23] Koren, Yoram, and Johann Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation." In Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, pp. 1398-1404. IEEE, 1991.

[24] Sanchez, Emilio, A. Rubio, and Alejo Avello. "An intuitive force feed-back to avoid singularity proximity and workspace boundaries in bilateral controlled systems based on virtual springs." In Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 2, pp. 1302-1307. IEEE, 2002.

[25] Mobasser, Farid, and Keyvan Hashtrudi-Zaad. "Predictive teleoperation using laser rangefinder." In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on*, pp. 1279-1282. IEEE, 2006.

[26] Huijun, Li, and Song Aiguo. "Virtual-environment modeling and correction for force-reflecting teleoperation with time delay." *Industrial Electronics, IEEE Transactions on* 54, no. 2 (2007): 1227-1233.

[27] Mitra, Probal, and Günter Niemeyer. "Model-mediated telemanipulation." *The International Journal of Robotics Research* 27, no. 2 (2008): 253-262.

[28] Tzafestas, Costas, Spyros Velanas, and George Fakiridis. "Adaptive impedance control in haptic teleoperation to improve transparency under time-delay." In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 212-219. IEEE, 2008.

[29] Yoon, Woo-Keun, Toshihiko Goshozono, Hiroshi Kawabe, Masahiro Kinami, Yuichi Tsumaki, Masaru Uchiyama, and Mitsushige Oda. "Model-based space robot teleoperation of ETS-VII manipulator." *Robotics and Automation, IEEE Transactions on* 20, no. 3 (2004): 602-612.

[30] Unger, Bertram J., A. Nicolaidis, Peter J. Berkelman, A. Thompson, S. Lederman, Roberta L. Klatzky, and Ralph L. Hollis. "Virtual peg-in-hole performance using a 6-dof magnetic levitation haptic device: Comparison with real forces and with visual guidance alone." In Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on, pp. 263-270. IEEE, 2002.

[31] Johnson, David E., and Peter Willemsen. "Six degree-of-freedom haptic rendering of complex polygonal models." In Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on, pp. 229-235. IEEE, 2003.

[32] Cameron, Stephen, and R. Culley. "Determining the minimum translational distance between two convex polyhedra." In Robotics and Automation. Proceedings. 1986 IEEE International Conference on, vol. 3, pp. 591-596. IEEE, 1986.

[33] J.C. Latombe. Robot Motion Planning. Kluwer Academic Publishers, 1991.

[34] M.C. Lin. Efficient Collision Detection for Animation and Robotics. PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, December 1993.

[35] Mirtich, Brian, and John Canny. "Impulse-based simulation of rigid bodies." In Proceedings of the 1995 symposium on Interactive 3D graphics, pp. 181-ff. ACM, 1995.

[36] Zilles, Craig B., and J. Kenneth Salisbury. "A constraint-based god-object method for haptic display." In Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, vol. 3, pp. 146-151. IEEE, 1995.

[37] Adachi, Yoshitaka, Takahiro Kumano, and Kouichi Ogino. "Intermediate representation for stiff virtual objects." In Virtual Reality Annual International Symposium, 1995. Proceedings., pp. 203-210. IEEE, 1995.

[38] Avila, Ricardo S., and Lisa M. Sobierajski. "A haptic interaction method for volume visualization." In Visualization'96. Proceedings., pp. 197-204. IEEE, 1996.

[39] Srinivasan, Mandayam A., and Cagatay Basdogan. "Haptics in virtual environments: Taxonomy, research status, and challenges." Computers & Graphics 21, no. 4 (1997): 393-404.

[40] Spong, Mark W., and Mathukumalli Vidyasagar. Robot dynamics and control. John Wiley & Sons, 2008.

[41] Brown, Christopher M. "Some mathematical and representational aspects of solid modeling." Pattern Analysis and Machine Intelligence, IEEE Transactions on 4 (1981): 444-453.

[42] Flynn, Patrick J., and Anil K. Jain. "BONSAI: 3D object recognition using constrained search." IEEE Transactions on Pattern Analysis and Machine Intelligence 13, no. 10 (1991): 1066-1075.

[43] Hearn, Donald, and M. Pauline Baker. Computer graphics, C version. Vol. 2. Upper Saddle River: Prentice Hall, 1997.

[44] Shene, Ching-Kuang, and John K. Johnstone. "On the planar intersection of natural quadrics." In Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, pp. 233-242. ACM, 1991.

[45] P. J. Besl, The free-form surface matching problem, in Machine Vision for Three-Dimensional Scences (H. Freeman, Ed.), pp. 25–71. Academic Press, San Diego, 1990.

[46] Dierckx, Paul. Curve and surface fitting with splines. Oxford University Press, Inc., 1993.

[47]Farin, Gerald E. Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann, 2002.

[48]Agin, Gerald J., and Thomas O. Binford. "Computer description of curved objects." Computers, IEEE Transactions on 100, no. 4 (1976): 439-449.

[49] Nevatia, Ramakant, and Thomas O. Binford. "Description and recognition of curved objects." Artificial Intelligence 8, no. 1 (1977): 77-98.

[50] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, Computer Graphics: Principles and Practice (2nd Ed.), Addison–Wesley 1990 ISBN-10: 0201848406.

[51] Ning, Paul, and Jules Bloomenthal. "An evaluation of implicit surface tilers."Computer Graphics and Applications, IEEE 13, no. 6 (1993): 33-41.

[52] Krishnamurthy, Venkat, and Marc Levoy. "Fitting smooth surfaces to dense polygon meshes." In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 313-324. ACM, 1996.

[53] Requicha, Aristides AG, and Jarek R. Rossignac. "Solid modeling and beyond." IEEE Computer Graphics and Applications 12, no. 5 (1992): 31-44.

[54] Hoffmann, Christoph M. Geometric and solid modeling. Morgan Kaufmann, 1989

[55] Colgate, J. Edward, and J. Michael Brown. "Factors affecting the z-width of a haptic display." In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, pp. 3205-3210. IEEE, 1994

[56] Quinlan, Sean. "Efficient distance computation between non-convex objects." In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, pp. 3324-3329. IEEE, 1994.

[57] Beckmann, N., Kriegel, H., Schneider, R., and Seeger, B. 1990. The r*-tree: An efficient and robust access method for points and rectangles. Proc. SIGMOD Conf. on Management of Data, 322–331

[58] Gottschalk, Stefan, Ming C. Lin, and Dinesh Manocha. "OBBTree: A hierarchical structure for rapid interference detection." In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 171-180. ACM, 1996.

[59]Kmiotek, P., and Y. Ruichek. "Object’s oriented bounding box based representation using laser range finder ensory data." In Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on, pp. 180-184. IEEE, 2008.

[60] Klatzky, Roberta L., and Susan J. Lederman. "Touch." Handbook of psychology (2003).

[61] Basdogan, C., Ho, C.-H., and Srinivasan, M. A. “A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments”. Proceedings of the ASME Dynamic Systems and Control Division, DSC-Vol.61 pp. 77–84,1997

[62] Kirby, Brian J. Micro-and nanoscale fluid mechanics: transport in microfluidic devices. Cambridge University Press, 2010.

[63] Tornero, J., J. Hamlin, and R. Kelley. "Spherical-object representation and fast distance computation for robotic applications." In Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, pp. 1602-1608. IEEE, 1991.

[64] O'Rourke, Joseph, and Norman Badler. "Decomposition of three-dimensional objects into spheres." Pattern Analysis and Machine Intelligence, IEEE Transactions on 3 (1979): 295-305.

[65] Overmars, Mark H. "Point location in fat subdivisions." Information Processing Letters 44, no. 5 (1992): 261-265.

[66] Kaufman, Arie, Daniel Cohen, and Roni Yagel. "Volume graphics." Computer 26, no. 7 (1993): 51-64.

[67] Cohen-Or, Daniel, and Arie Kaufman. "Fundamentals of surface voxelization."Graphical models and image processing 57, no. 6 (1995): 453-461.

[68] Kaufman, Arie, and Eyal Shimony. "3D scan-conversion algorithms for voxel-based graphics." In Proceedings of the 1986 workshop on Interactive 3D graphics, pp. 45-75. ACM, 1987.

[69] Mukherjee, J. K. "AI based tele-operation support through voxel based workspace modeling and automated 3D robot path determination." In TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region, vol. 1, pp. 305-309. IEEE, 2003.

[70] Reddy, D. R., and S. Rubin. Representation of Three-Dimensional Objects. No. Cmu-Cs-78-113. Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, 1978.

[71] Gargantini, Irene. "An effective way to represent uadtrees." Communications of the ACM 25, no. 12 (1982): 905-910.

[72] Wah, Wai Hon. "Introduction to STL format." Polytechnical University of Hong Kong (1999)

[73] Cohen-Or, Daniel, and Arie Kaufman. "3D line voxelization and connectivity control." Computer Graphics and Applications, IEEE 17, no. 6 (1997): 80-87.

[74] Huang, Jian, Roni Yagel, Vassily Filippov, and Yair Kurzion. "An accurate method for voxelizing polygon meshes." In Volume Visualization, 1998. IEEE Symposium on, pp. 119-126. IEEE, 1998.

[75] Helman, D. "Efficient image processing algorithms on the scan line array processor." IEEE Trans. Pattern Anal. Mach. Intell. 17, no. 1 (1995): 47-56.

[76] Kameyama, K-I., and Koichi Ohtomi. "A direct 3-D shape modeling system." In Virtual Reality Annual International Symposium, 1993 IEEE, pp. 519-524. IEEE, 1993.

[77] Kai, Chua Chee, Gan GK Jacob, and Tong Mei. "Interface between CAD and rapid prototyping systems. Part 1: a study of existing interfaces." The International Journal of Advanced Manufacturing Technology 13, no. 8 (1997): 566-570.

[78] Levine, Michael W., and Jeremy M. Shefner. "Fundamentals of sensation and perception." (1991).

[79] Soille, Pierre. Morphological image analysis: principles and applications. Springer-Verlag New York, Inc., 2003.

[80] Mukherjee, J. K. "A vicinity activation model for intelligent interface in tele-operation." In Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, pp. 1-6. IEEE, 2012.

[81] Lawrence, Dale A. "Stability and transparency in bilateral teleoperation."Robotics and Automation, IEEE Transactions on 9, no. 5 (1993): 624-637.

[82] Hokayem, Peter F., and Mark W. Spong. "Bilateral teleoperation: An historical survey." Automatica 42, no. 12 (2006): 2035-2057.

[83] Hu, Z., S. E. Salcudean, and P. D. Loewen. "Robust controller design for teleoperation systems." In Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on, vol. 3, pp. 2127-2132. IEEE, 1995.

[84] Hogan, N. "Multivariable mechanics of the neuromuscular system." In IEEE Annual Conference of the Engineering in Medicine and Biology Society, pp. 594-598. 1986.

[85] Shimoga, Karun B. "Finger force and touch feedback issues in dexterous telemanipulation." In Intelligent Robotic Systems for Space Exploration, 1992. Proceedings. Fourth Annual Conference on, pp. 159-178. IEEE, 1992.

[86] Burdea, Grigore C., and Grigore C. Burdea. Force and touch feedback for virtual reality. New York: Wiley, 1996.

[87] Rosenberg, Louis B., and Bernard D. Adelstein. "Perceptual decomposition of virtual haptic surfaces." In Virtual Reality, 1993. Proceedings., IEEE 1993 Symposium on Research Frontiers in, pp. 46-53. IEEE, 1993.

[88] S. Yamakawa, H. Fujimoto, S. Manabe, Y. Kobayashi, The necessary conditions of the scaling ratio in master-slave systems based on human difference limen of force sense, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 35 (2) (2005) 275–282.

[89] Blauert, Jens, and Robert A. Butler. "Spatial hearing: The psychophysics of human sound localization by jens blauert." The Journal of the Acoustical Society of America 77, no. 1 (1985): 334-335.

[90] Carlile, Simon. Virtual auditory space: Generation and applications. Austin TX, USA: RG Landes, 1996

[91] Loomis, Jack M., Chick Hebert, and Joseph G. Cicinelli. "Active localization of virtual sounds." The Journal of the Acoustical Society of America 88, no. 4 (1990): 1757-1764.

[92] Begault, Durand R. 3-D sound for virtual reality and multimedia. Vol. 955. Boston etc: AP professional, 1994.

[93] Doerr, K-U., Holger Rademacher, Silke Huesgen, and Wolfgang Kubbat. "Evaluation of a low-cost 3D sound system for immersive virtual reality training systems." Visualization and Computer Graphics, IEEE Transactions on 13, no. 2 (2007): 204-212.

[94] Spence, Charles, Francesco Pavani, and Jon Driver. "Crossmodal links between vision and touch in covert endogenous spatial attention." Journal of Experimental Psychology: Human Perception and Performance 26, no. 4 (2000): 1298.

[95] Rock, Irvin, and Jack Victor. "Vision and touch: An experimentally created conflict between the two senses." Science 143, no. 3606 (1964): 594-596.

[96] Heller, Morton A., Jeffrey A. Calcaterra, Shavonda L. Green, and Latonya Brown. "Intersensory conflict between vision and touch: The response modality dominates when precise, attention-riveting judgments are required." Perception & psychophysics 61, no. 7 (1999): 1384-1398.

[97] Ernst, M., and M. Banks. "Does vision always dominate haptics." In Touch in Virtual Environments Conference. 2001.

[98] O'Sullivan, Carol, Ralph Radach, and Steven Collins. A model of collision perception for real-time animation. Springer Vienna, 1999.

[99] O'Sullivan, Carol, and John Dingliana. "Collisions and perception." ACM Transactions on Graphics (TOG) 20, no. 3 (2001): 151-168.

[100] O'Sullivan, Carol, John Dingliana, Thanh Giang, and Mary K. Kaiser. "Evaluating the visual fidelity of physically based animations." ACM Transactions on Graphics (TOG) 22, no. 3 (2003): 527-536.

[101] J. Tomero, J. Hamlin, and R.B.Kelly, “Spherical-Object Representation and Fast Distance Computation for Robotic Applications”, Proceedings of the 1991 IEEE International Conference on Robotics and Automation, April 1991. pp. 1602-1608.

[102] Bargen, Bradley, and Terence Peter Donnelly. “Inside Directx (Microsoft Programming Series)”. (1998)

[103] G.R. Hopkinson et al “Radiation effects on CCDs at low operating temperatures.” IEEE Transactions on Nuclear Science 52:6, 2664-2671. Online publication: 1-Dec-2005.

[104] C. R. Dyer. Volumetric scene reconstruction from multiple views. Foundations of Image Analysis, pages 46 9 – 48 9, 2001.

[105] Peter J Kovach (2000) Inside Direct3D. Microsoft Press.ISBN-13: 978-0735606135

[106] Delaye, Vincent, and Pierre Labeye. "High-resolution eye-safe time-of-flight laser range finding." AeroSense 2000. International Society for Optics and Photonics, 2000.

[107] Amann, Markus-Christian, et al. "Laser ranging: a critical review of usual techniques for distance measurement." Optical Engineering 40.1 (2001): 10-19.