Uncertainty Estimation in Vision-Aided Robot Teleoperation System

Shubham Vyas

Technische Universiteit Delft

Uncertainty Estimation in Vision-Aided Robot Teleoperation System

by

Shubham Vyas

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology.

to be defended publicly on Friday January 12, 2018 at 01:00 PM.

Student number:	4521919	
Supervisors:	Dr. ir. C.J.M Verhoeven,	TU Delft
	Dr. -Ing. T Krueger,	ESA-ESTEC
	Dr. -Ing. A Schiele,	TU Delft
Thesis committee:	Prof. Dr. ir. E.K.A Gill,	TU Delft
	Dr. ir. G.H.C.E de Croon,	TU Delft
	Dr. ir. C.J.M Verhoeven,	TU Delft
	Dr. -Ing. T Krueger,	ESA-ESTEC
	Dr. -Ing. A Schiele,	TU Delft

TUDelft Delft University of Technology

Preface

This thesis was submitted in partial fulfilment of the requirements for the MSc at the Department of Space Engineering, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. The research work was carried out at the Human Robot Interaction Laboratory at the European Space Agency's European Space Research and Technology Centre (ESA-ESTEC) under the supervision of Dr. -ING Thomas Krueger.

The thesis aims to fill a gap in the understanding of errors and uncertainties in robot-vision systems with the hope that it sets a paradigm where accuracy in a robot system is not an afterthought but rather a part of the development process. The results obtained from this thesis will be integrated into the teleoperation framework at the Human Robot Interaction laboratory and used in the future. The knowledge of errors and uncertainties could potentially increase the efficiency of teleoperated tasks by providing more information to the user and the system. This system could specifically used to improve future space telerobotic missions.

This thesis would not be possible without the help and support of my supervisors at ESA and TU Delft and the wonderful colleagues at the lab. I would like to thank Dr.-Ing Thomas Krueger for his guidance and support which were vital to the success of this thesis. His encouragement led me to explore multiple areas of interest and develop a broader understanding of the field. This thesis was made possible due to my supervisors at TU Delft: Dr. Ir. Chris Verhoeven and Dr. -Ing Andrè Schiele. I would like to thank them for giving me an opportunity to intern at the lab at ESTEC. It was from the work done during the internship, that the topic for this thesis was conceived.

During my time at the laboratory for one and a half years, I have learnt a lot, more than I could have possibly imagined. This is one of the best experiences that I will look back to. I would like to thank the current and the former members of the Human Robot Interaction laboratory for making my time at the lab memorable. I look up to each member of the lab as they showcase qualities I would like to imbibe in myself as a future engineer-researcher. Thanks to Frank and Emiel for showing me that the time spent doing the things *right* instead of *acceptable* is worth the effort as it saves a lot of time in the future. Thanks to Jan for our numerous discussions which helped me think critically about my work. I will remember the lessons you taught me during these discussions with a smile. I would also like to thank my friends at ESTEC and in Delft who made my time in the Netherlands memorable and fun. Thank you for the innumerable memories which I shall always cherish. A particular thank you goes to Yeshwanth for the numerous discussions throughout my time in the Netherlands which resulted in motivating me to do better things and be a better person. Last but not the least, I would like to thank my parents and my bother for being there to support me while I was far away from home.

Shubham Vyas
Delft, January 2018

Abstract

Teleoperation allows the use of human intelligence and decision making in remote tasks which are too dangerous for humans to perform. Technologies such as force feedback and haptic guidance have shown to increase task efficiency during teleoperation. In an unmodeled environment, sensors provide input for haptic guidance or present extra information about the environment to the user in order to make decisions and to perform the tasks. These sensors come with inherent errors and uncertainties which propagate through the teleoperation system. The absence of knowledge of these errors has been shown to cause deterioration in the task performance. These errors can further cause the application of forces on the environment by the robot without the knowledge of the user while using haptic guidance. Thereby, the strategies being used to increase task performance can have some adverse hidden effects. Thus, it crucial to have an understanding of the behaviour of the errors and uncertainties in the system. It is considered critical for making decisions about how the robot system can be controlled and used to manipulate objects remotely.

In this thesis, a novel framework for estimating the uncertainties in a vision-aided teleoperation system in real-time is introduced. The uncertainty estimate can then be used by the control system or communicated to the user. Methods to use the uncertainty estimate for haptic guidance and for user display are proposed. Furthermore, the thesis analyzes the behaviour of the uncertainties in the system and the sensitivity of the system to individual component errors. It evaluates the uncertainties in individual components of the system and implements an uncertainty model for each of them. It then provides a method to propagate these uncertainty models through the system. This results in a final uncertainty estimate in the frame of reference of interest for the task. Experiments were performed to validate the component uncertainty models, the propagation method, and the system as a whole. Additionally, an inverse of the propagation method is also conceived so as to obtain the component accuracy specification from system uncertainty requirements. This can be used in the design of future teleoperation systems.

Contents

List of Figures

List of Tables

1 Introduction

This thesis focuses on determining and estimating the errors in a vision-aided teleoperation system and their uncertainties. It aims to set a paradigm of error and uncertainty evaluation in robot-vision systems and their applications in teleoperation with force-feedback.

Robot-vision systems are used for aiding teleoperation tasks at the Human Robotics Interaction Laboratory (formerly known as Telerobotics and Haptics Laboratory) at the European Space Agency's European Space Research and Technology Centre (ESA-ESTEC) at Noordwijk in the Netherlands.

This chapter provides the Motivation for the research in Section 1.1 and along with Section 1.3 explains why the knowledge of errors and uncertainties is important for a vision-aided robot teleoperation system. An overview of the robot-vision system used for teleoperation is given in Background Section 1.2 along with an introduction to teleoperation and Haptic Feedback. The main problem addressed by this thesis is explained in Section 1.3. This is followed by a description of the goals of this thesis. Finally, the thesis outline is presented which sets the course for the remainder of this thesis.

1.1. Motivation For Research

On Tuesday 8th September, the Danish Astronaut *Andreas Mogenson*, controlled a rover on ground from the International Space Station (ISS) and successfully navigated an obstacle course and performed a sub-millimeter assembly task. To fully understand the importance of this experiment, one should realize that he was on the ISS during the experiment travelling at $27,500\ km/h$ about $400\ km$ above the surface of the earth. The signal to control the robot arm and the rover travelled $90,000\ km$ as the round trip distance due to the use of geostationary relay satellites . Even with time delays caused by such large distances, he was able to successfully perform the task. This was in part due to the force-feedback enabled teleoperation. A custom made one degree of freedom (1-DoF) joystick (also used in *Haptics - I* [3]) enabled the astronaut to feel the forces that the robot on ground felt which interacting with its environment. This experiment was called *Interact* [4–6]. *Interact* showed that a robot could be teleoperated in an unknown environment to perform tasks remotely. The robot consisted of an Ambot Mobile Platform, two KUKA 7-DoF LightWeight Robot Arms with RobotiQ Grippers and a stereo camera mounted on the Schunk Powerball lightweight robot arm. The rover can be seen in Fig. 1.1.

This experiment re-used the force-feedback joystick and tablet PC which were flown to the ISS for the *Haptics-I* [3] and *Haptics-II* [7] experiments. For the experiment, the astronaut had to locate a mechanical taskboard and navigate to it avoiding obstacles. After the taskboard was located and approached, a coarse alignment of the robot arm holding the peg and the hole on the task board was performed using an external tracking system. He then performed fine alignment of the peg with the hole using the force-feedback from the *Haptics-I* joystick to control the movement of the robot arm on the rover. After alignment, he performed the peg-in-hole task which required sub-millimeter accuracy for correct insertion. This demonstrated the ability to perform sub-millimeter assembly tasks with tight mechanical tolerances over time delays. The force-feedback allowed the astronaut to feel the misalignment and perform the appropriate corrections.

Experiments like these showcase the development of technologies that one day could enable control of robots on the surface of Moon or Mars and assist astronauts in their activities. This could enable a

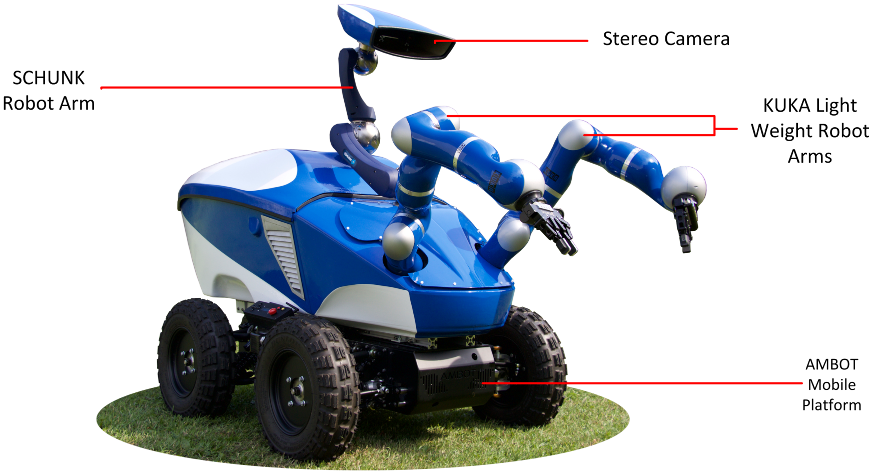


Figure 1.1: Interact Centaur Rover

reduction in Extra Vehicular Activities (EVAs) to only when required for the science tasks. Maintenance tasks (Such as Habitat Building, repair, etc) could be performed remotely using teleoperated robots with force-feedback control for higher efficiency. This would help reducing risks for the astronaut during spaceflight.
However, for the *Interact* experiment, an external tracking system [8] was used for the navigation and alignment of the peg with the hole. The tracking system served as the ground truth for the driving and alignment and can be seen in Fig. 1.2. Such an experiment would not be possible without the tracking system if no modifications are made to the robot and the rover so that it can sense its environment. For sensing the environment, various sensors such as Cameras (Monocular, Stereo, Time of Flight) or other sensors such as LIDAR can be used. This causes the transition of the teleoperation task from simple teleoperation with video to model-based teleoperation as a model of the environment is created using these sensors and this model is used during teleoperation to provide information to the user. Introduction of sensors also introduces uncertainty of the sensor measurements in the system. Thus, to use these sensors to enhance the teleoperation performance and make the teleoperation system independent of the Motion Capture system the knowledge of the uncertainties introduced by the sensor systems is of paramount importance.

Motion Capture Systems such as *Vicon* are not available on the surface of Moon or Mars and neither is it feasible to set up these systems there. Thus, the use of sensors to the model the environment is the only option. This can already be seen in the rover missions to Mars, the Mars Exploration Rover mission had 20 cameras, out of which 14 were engineering cameras for modelling the environment for navigation etc. [9]. The Mars Science Laboratory mission has 12 engineering cameras [10] which are used for rover operation. It can be safely assumed that the future rover missions, including those which might include teleoperation from an orbiter, will have cameras and other sensors to model the environment. These sensors introduce uncertain measurements to the robot system and the performance of the robot system is dependent of the knowledge of the uncertainties in its system. Thus, a study of these uncertainties is warranted and carried out in this thesis.

1.2. Background

This section gives an overview of the system that is used for the study of uncertainties in vision-aided robot teleoperation. It also provides an introduction to teleoperation with Haptic (force) feedback and the effects of uncertainty in such a system.

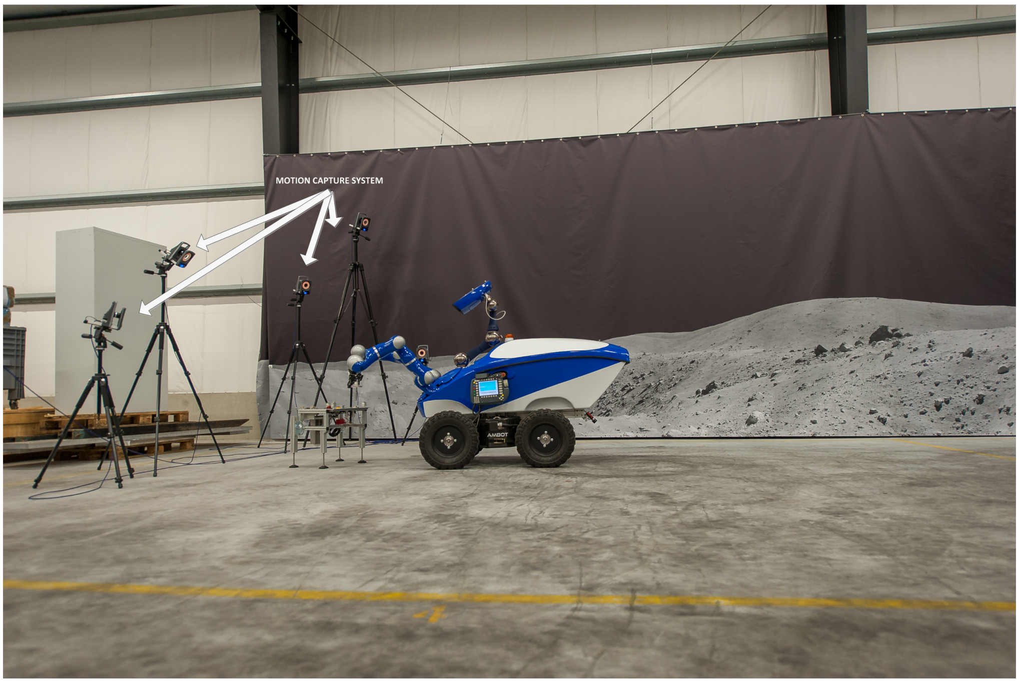


Figure 1.2: Motion Capture System Used for the Interact Experiment

1.2.1. Overview of the System

The vision-aided teleoperation system considered in this thesis consists of two 7-DoF robot arms mounted on a rover platform. A sensor (Mono + Stereo Camera) is mounted on one of the robot arms for object detection and the other robot arm has a gripper mounted on its end for object manipulation. This thesis concerns with the study of errors in the loop pertaining Object Detection using the camera till object manipulation using the gripper. This loop can be seen in Fig. 1.3.

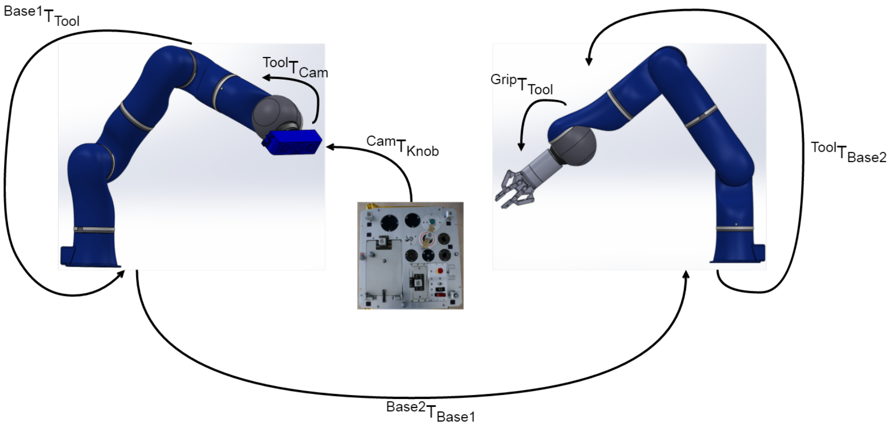


Figure 1.3: Transformation in Vision-Aided Teleoperation System

The transformations shown in Fig. 1.3 show the position of the object in subscript with respect to the frame written in superscript. The sensor on the robot arm consists of a 2D camera for video transmission to the operator as well as a 3D camera for object detection. Any other sensor can be used as well as long as the methods given in this thesis for error modelling and propagation are followed.

The sensor is used to find an object on the task-board, such as a knob. The object is detected in the sensor's frame of reference. This pose of the object is then propagated through the transformations shown to obtain its position in the Gripper's frame of reference for manipulation. These transformations can be listed as:

$CameraFrame \rightarrow ToolFrame\ 1 \rightarrow BaseFrame\ 1 \rightarrow BaseFrame\ 2 \rightarrow ToolFrame\ 2 \rightarrow GripperFrame$

This chain is represented mathematically using transformation matrices as follows:

$$^{Tb}T_{Grip} = {}^{Tb}T_{Cam} \cdot {}^{Cam}T_{Tool1} \cdot {}^{Tool1}T_{Base1} \cdot {}^{Base1}T_{Base2} \cdot {}^{Base2}T_{Tool2} \cdot {}^{Tool2}T_{Grip} \tag{1.1}$$

The equation 1.1 assumes ideal transformation matrices for each transformation operation. However, in real life, ideal transformations are extremely hard to obtain. All measurements made in a robot-vision system are approximations of the actual physical quantities. Due to this, the uncertainty in the error of the transformations could cause an error in the position of the knob in the gripper frame preventing it from being successfully manipulated. It can be seen in Fig. 1.4, that the effect of difference in estimated and true parameters can cause difference in positions, which if unaccounted for, could cause problems in the system.

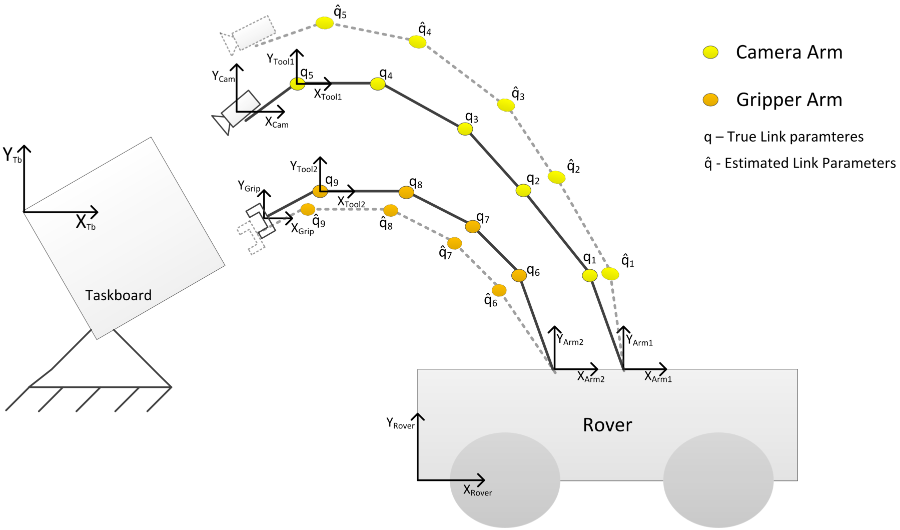


Figure 1.4: Difference Between Estimated and True Parameters in a Teleoperation System

The errors in the transformations are assumed to be distributed normally (due to the Central Limit Theorem [11]), i.e they form a Gaussian Distribution. The distribution is centered about zero (0) as all the systematic errors are assumed to be removed using proper calibration techniques. The transformation along with the uncertainty distribution is represented as:

$$\mathcal{N}(^{Tb}T_{Grip}, \Sigma) \tag{1.2}$$

where:

$\mathcal{N}$ - Gaussian/Normal Distribution.
T - Transformation Matrix which is the Mean of the Distribution.
Σ - Covariance Matrix

Since Gaussian distribution is assumed for all transformations, they are generally represented as (T, Σ) in this thesis.

The components of the vision-aided teleoperation system being used in this thesis are listed below:

- *2 7-DoF Robot Arms* - The Robot arms used for this thesis study are Kuka LWR 4+ as they are capable of operating in position control and impedance control required for the teleoperation setup at the Human Robot Interaction Laboratory.

- *Stereo Camera with Texture Projection* - An Ensenso N35 stereo camera with texture projection is used. Texture projection allows surface reconstruction for surfaces with no artifacts that can be matched for stereo depth calculation.

- *Color Camera* - A Prosilica GT2000C is used for streaming video back to the operator. A potential use for the color video stream is the communication of final Uncertainty using overlays on the video stream.
- *Taskboard* - A taskboard same as the one used for the *Interact* experiment is used in this thesis. The taskboard consists of objects that are detected using the vision system and manipulated using the grippers.
- *Motion Capture System* - An external tracking system was used in the Interact experiment. This research aims to create a system that replaces this. Hence, the external tracking system is required to compare performance of both systems and verify if the new system developed is capable of replacing the old system.

1.2.2. Teleoperation and Haptic Feedback

Robots can be used in situations where it is not possible or too dangerous for humans to go and perform manipulation tasks. Examples of such areas could be outer space, nuclear hazard areas, underwater repairs etc. Such environments are usually partially or completely unknown. Thus, it is difficult for fully autonomous robots to operate in these environments. The decision-making ability of humans is still required in such unknown environments. For this, the robots can be teleoperated i.e controlled by the humans using a master device. A master device can be anything that is used to record human actions or take inputs from a human operator. Examples include but are not limited to joysticks, touchscreen etc. A teleoperation system consists of a master device with which the operator interacts and this is used to control the slave device which is a robot. Using teleoperation, the planning and decision-making skills of the human can be used while the human is in a safe environment while remotely performing the task. To extend humans capability to teleoperate the robot, both visual and haptic feedback from the robot and its environment is required. This ensures the operator is provided with conditions similar to the robot in remote environment. Visual feedback can be provided using cameras and computer vision systems. Haptic feedback can be classified as the following:

- **Force Feedback**
  The reaction forces felt by the robot while interacting with the remote environment are transmitted to the operator using the master device. These forces can be scaled and/or mapped according to the task requirements. When a teleoperation system communicates both position and forces between the master and slave side, it is called bilateral teleoperation.
- **Haptic Guidance**
  Since the master device is able to transmit forces to the operator, this can be used to transmit virtual forces to the operation. These can be used to increase task efficiency as seen in [1]. The virtual forces could be used to provide various optimizations in the task such as obstacle avoidance, optimum power consumption etc.

1.2.3. Effects of Errors in Teleoperation

Since only for haptic guidance based teleoperation, a model of the environment is required, the effects of errors in haptic guidance are discussed in this part. Haptic guidance requires a reference using which the guidance forces can be modeled. This reference is taken by measuring and modeling the remote environment using sensors. Since all sensors have errors and uncertainties associated with them, these uncertainties are implanted in the teleoperation system. These uncertainties propagate through the system along with other sources of uncertainties (Calibration, robot joint encoders) and result in errors in the reference taken for haptic guidance.

Experiments conducted by Mol et. al in [12] and by Mol in [13] show that if orientation and translation errors are higher than certain thresholds while aligning the peg to the hole, it can prevent peg-in-hole task completion in an autonomous task using robots with impedance control. Thus, it is then required to use manual control of the robot (teleoperation) for such tasks. Haptic feedback can be used in manual control to complete such tasks. It is shown by Oosterhout et. al in [1] that Haptic Guidance, a method of haptic feedback, can assist the user and increase task performance during teleoperation. Haptic guidance is also shown to increase task performance in other applications such as in robot-assisted surgery [14–16], remote assembly [17–19] and assistance during driving a vehicle [20, 21].

It is assumed in all of the above works except [1] that the reference for haptic guidance is perfect. The reference for haptic guidance is taken from the sensor system present on the remote system that models the environment. It is generally assumed that the models used by haptic guidance system are perfect representations of the environment and do not contain any errors. But in practical applications, the model does contain errors and uncertainties as sensor systems are not ideal. It is shown in [1], by using user studies, that errors in reference can degrade task performance in a haptic guidance task. Furthermore, lack of knowledge of errors in the reference provided to the guidance system from the sensors can also result in incorrect guidance forces being transferred to the user and the system may exert large forces on the environment without the knowledge of the user as shown by Smisek et al. in [22].

From the above discussion, the effects of errors in haptic guidance can be summarized as follows:

- Decrease in Task Efficiency.
- Exertion of forces on the environment without the knowledge of the operator.

1.3. Problem Description

The first step to deal with the effects of the errors in a teleoperation system is to understand and characterize the errors. By doing this, the errors can be estimated in the system and this gives an insight to the system performance. To the best of author's knowledge, no study has been performed to evaluate the errors and uncertainty in those errors for a robot-vision system. This extends to robot-vision systems used for teleoperation as well. Thus, there is no method currently for evaluation of errors which can be followed or implemented. The knowledge of uncertainty in errors of a system have large potential uses in adjusting the system so that it fits best for the given teleoperation task, however no such uncertainty models or estimation methods are available.

Human operators use teleoperation system to perform tasks remotely and can potentially use haptic guidance for higher task efficiency. But the operator does not know how much he/she can trust the guidance forces if the system has no knowledge about the errors in the reference used for haptic guidance. This leads to the definition of the main problem statement addressed in this thesis:

> **Main Problem Statement:**
> *During teleoperation, the human operator does not have the knowledge of the errors in the system. This can lead to unknown errors that can reduce task performance and exert forces on the remote environment unknown to the operator. Currently, no system exists to determine the spatial errors and uncertainties in those errors in a vision-aided robot teleoperation system and communicate them to the user to assist in making decisions or to adjust the control method used for the task.*

This problem statement can be divided in to sub-problems that solves parts of the issue and if these solutions are combined, can give a final solution to address the main problem. They provide a step-by-step approach to answer the main problem given above. These sub-problems can be stated as:

> **Sub-Problem 1:**
> *There is no standard and uniform way defined yet on representing the uncertainty of various components of the teleoperation system. This is followed by the lack of error and uncertainty models for components of the teleoperation system such as Camera, Robot Arm etc.*

> **Sub-Problem 2:**
> *Once the individual error models of the components are obtained, there is no method currently devised to propagate the errors through the chain of transformations in the teleoperation system to give the final uncertainty estimate. Without this, the system's behaviour cannot be studied.*

Sub-Problem 3:
The is no uncertainty estimation method/framework outlined under which the error propagation can be done during teleoperation. There is no well-defined way currently as to how uncertainty estimation can be carried out along with teleoperation. Due to this, there is no way of knowing the errors in real-time during teleoperation tasks.

1.4. Thesis Goals

The goals of this thesis is to address and solve the problem statements stated in Section. 1.3. The thesis aims to create a proof-of-concept system that can demonstrate the estimation of error in a teleoperation system in real-time and make it available to the user and the control system. The goals are defined as solutions to the sub-problems given in Section. 1.3.

The first goal is related to the first sub-problem:

Goal 1:
To find a uniform way of representing uncertainty of various components and use this to model the uncertainty of the components used in the teleoperation system.

This goal aims to propose a way to uniformly describe errors in a teleoperation system. It furthermore uses this method of description to determine the uncertainty models using theory and experiments for various components of the system such as Cameras, Robot arm, etc.

The second goal is related to the second sub-problem:

Goal 2:
To propagate the uncertainties through the chain of transformations in the teleoperation system and study the behaviour of errors in the system.

This goals aims to find and apply a method of uncertainty propagation so that the object can be represented in the gripper frame along with its uncertainty. This would be a result of propagation using the transformations and their uncertainties in the kinematic chain. Once, this is achieved, the system's response of change in uncertainties/accuracies of its components can be studied.

The third goal is related to the third sub-problem:

Goal 3:
To determine, describe and demonstrate an Uncertainty Estimation Framework *that can estimate the errors in the vision-aided teleoperation system in real-time.*

This goal answers the need for an uncertainty estimation framework by creating a proof-of-concept demonstrator using the concepts developed by completing the previous goals. Using this, a common method for uncertainty estimation can be devised for robot-vision systems with varied applications including teleoperation. This framework is extensible and allows other sensors and transformations to be included in it. Furthermore, it gives a final uncertainty estimate in real-time and this can either be displayed to the operator and/or fed into the control system.

1.5. Thesis Outline

This thesis is arranged as follows, the next three chapters address the sub-problems and their respective goals given in Sections. 1.3 and 1.4. Chapter 2 describes the error models of various components of the teleoperation system along with an uniform representation method. A method to propagate these error models is given in Chapter. 3 along with the study on the behaviour of the system with changes to the error models of its components. These results are combined in Chapter. 4 to form an uncertainty estimation framework to estimate the final uncertainty in real-time during teleoperation along with experimental validation. It also provides the potential applications of the final error estimate and how the error estimate could be used to improve teleoperation task efficiency. The final chapter draws the conclusions from the work done in this thesis and presents ideas that can be implemented to further enhance the work presented in this thesis.

2

Error and Uncertainty Modelling

In this Chapter, various sources of errors are studied. The uncertainties in these errors are modelled in a way that facilitates the propagation of errors and uncertainties.

2.1. Problem

Techniques such as haptic guidance have shown to increase task performance during teleoperation tasks [1, 14–21]. Vision based sensors (such as mono/stereo cameras) can be used to generate remote environment models based on which guidance forces for a task can be calculated. Vision based sensors, like any other sensors, come with inherent inaccuracies and uncertainties. These uncertainties propagate through the teleoperation system along with other inaccuracies present in the system such as in the actuators etc. The effects of errors in certain types of shared-control teleoperation systems such as Haptic Guidance have been studied and documented [1, 12, 13, 22]. The inaccuracies in haptic guidance can result in lower task performance [1] and/or exertion of hidden forces on the environment [22].

The values of spatial errors in various parts of a teleoperation system and how they propagate have not yet been studied. The first step in finding the total error in the teleoperation system is to model errors in individual components. These can then be propagated to find the total error in the system.

To find out the total error in a vision-aided teleoperation system, the individual errors of the components used in the teleoperation system have to be modelled.

2.2. Approach: Modelling individual error sources for error propagation

In this chapter, individual error sources in the slave side of the teleoperation setup are looked at and their uncertainties are modelled. These uncertainties are modelled as covariance matrices for the 6×1 pose vector containing translations and rotations: $\begin{bmatrix} T_x & T_y & T_z & R_x & R_y & R_z \end{bmatrix}$. This covariance matrix can be represented as:

$$\Sigma = \begin{bmatrix} \boldsymbol{\sigma^2_{T_xT_x}} & \sigma^2_{T_xT_y} & \sigma^2_{T_xT_z} & \sigma^2_{T_xR_x} & \sigma^2_{T_xR_y} & \sigma^2_{T_xR_z} \\ \sigma^2_{T_yT_x} & \boldsymbol{\sigma^2_{T_yT_y}} & \sigma^2_{T_yT_z} & \sigma^2_{T_yR_x} & \sigma^2_{T_yR_y} & \sigma^2_{T_yR_z} \\ \sigma^2_{T_zT_x} & \sigma^2_{T_zT_y} & \boldsymbol{\sigma^2_{T_zT_z}} & \sigma^2_{T_zR_x} & \sigma^2_{T_zR_y} & \sigma^2_{T_zR_z} \\ \sigma^2_{R_xT_x} & \sigma^2_{R_xT_y} & \sigma^2_{R_xT_z} & \boldsymbol{\sigma^2_{R_xR_x}} & \sigma^2_{R_xR_y} & \sigma^2_{R_xR_z} \\ \sigma^2_{R_yT_x} & \sigma^2_{R_yT_y} & \sigma^2_{R_yT_z} & \sigma^2_{R_yR_x} & \boldsymbol{\sigma^2_{R_yR_y}} & \sigma^2_{R_yR_z} \\ \sigma^2_{R_zT_x} & \sigma^2_{R_zT_y} & \sigma^2_{R_zT_z} & \sigma^2_{R_zR_x} & \sigma^2_{R_zR_y} & \boldsymbol{\sigma^2_{R_zR_z}} \end{bmatrix} \tag{2.1}$$

where σ^2 represents the covariance and T and R represent the translation and rotation in $x, y\ and$ z directions respectively. The diagonal entries of the matrix are the squares of standard deviations (variances) in the respective elements of pose.

Modelling the individual uncertainties in the system as covariance matrices enable propagation of the uncertainties through the system. The propagation model will be explained in later chapters.

Example:
If the uncertainties in each degree of freedom is given by the variance vector which is the square of standard deviations in each DoF:

$$\begin{bmatrix} 0.3mm & 0.1mm & 0.5mm & 0.1^0 & 0.3^0 & 0.05^0 \end{bmatrix} \tag{2.2}$$

And if it is assumed that there is no correlation between the errors in different degrees of freedom, then the covariance matrix is generated as:

$$\Sigma = \begin{bmatrix} 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \tag{2.3}$$

In the following sections of this chapter, individual parts of the vision-aided teleoperation system are analyzed and an error model in the form of a covariance matrix is devised for each of the components. Some of these error models (matrices) are dynamic and change during the operation of the system while others are static.

2.3. Stereo Vision System

2.3.1. Stereo Camera

The teleoperation setup taken as reference for this thesis consists of a projected texture stereo vision sensor: Ensenso N35 camera. The N35 camera outputs a point cloud of the scene in its view field. The errors in the measurements made by this camera (Point Cloud) are studied and modelled in this section.
For this camera, the manufacturer *Ensenso* provides the data in the form of look-up table for Z-Accuracy, view field X, view field Y, optics blur, and pixel size with respect to the distance [23]. From these values, a method is proposed to determine the covariance matrix to represent the uncertainty. This method along with its assumptions is described as follows:
Initially the point cloud from the camera is acquired using a frame grabber. Then the scene point cloud is scanned to find the object of interest. Once the object of interest is found, the pose of the object in the scene with respect to the camera frame is determined. From this pose, the region of interest in the scene point cloud is known and the uncertainty in this region can be determined. This process can be seen in the flowchart Figure. 2.1.

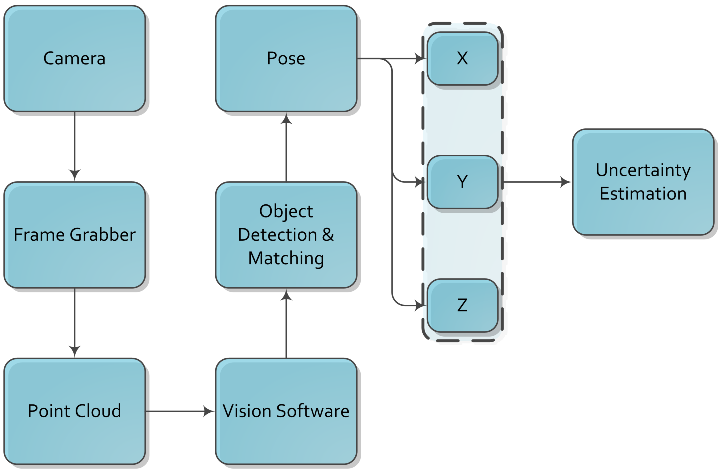


Figure 2.1: 3D Vision Uncertainty Estimation Flow Chart

Since there are no correlations between the pose elements at this stage, the covariance matrix is a diagonal matrix. Furthermore, the variance is only in the translation part as the camera performs only translation measurements. Hence, only $\sigma^2_{T_xT_x}, \sigma^2_{T_yT_y}\ and\ \sigma^2_{T_zT_z}$ have to be determined.
From the pose of the object detected, the Z distance (d) is extracted. At this distance, the accuracy parameters are extracted from the look-up table. The variances for $x, y\ and\ z$ translations are calculated as follows:

- ***For X and Y:***
  The contributing factors for uncertainty in X and Y directions are the optics blur, the pixel size, and the stereo reprojection error. Optics Blur (Ob) in pixels is obtained from the data given by the manufacturer in the form of a look-up table. This can be denoted as a function of distance as: $f_1(d)$:
  $$Ob_x = f_1(d)\ px \tag{2.4}$$
  Pixel size (Px) in millimeters is also determined from the look-up table and is denoted as function $f_2(d)$:
  $$Px_x = f_2(d)\ mm \tag{2.5}$$
  The variance in x $and\ y$ directions are calculated as the product of the optics blur, pixel size and the stereo reprojection error (e_{rpoj}). The reprojection error is obtained while performing camera calibration and is generally between 0.1 and 0.3 as was informed by the camera support. For either X or Y, this can be expressed mathematically as:
  $$\sigma^2_{T_xT_x} = Ob_x \cdot Px_x \cdot e_{rpoj}\ mm \tag{2.6}$$

- ***For Z:***
  The Z-accuracy (e_z) is obtained as a function of distance from the look-up table. This is assumed to be the bounds for maximum error in the translation along z. The uncertainty associated with this error is assumed to be distributed uniformly between these bounds. The next step is to determine the variance along z which can be calculated by finding the second central moment of the uniform distribution. The second central moment of the distribution is also the standard deviation of the distribution. This value of standard deviation is used to find variance in the z direction. This step is in accordance to the Guidance to Uncertainty in Measurement by ISO [24, 25]. The process explained above can be visualized as shown by the steps 1-3 in Figure. 2.2 and equations 2.7, 2.8, and 2.9.
  $$e_z = f_3(d) \tag{2.7}$$
  The nth moment of a distribution about a value c is defined as:
  $$\mu_n = \int_{-\infty}^{\infty} (x-c)^n\, f(x)\, dx \tag{2.8}$$
  Evaluating this integral for a uniform distribution about the mean with half width e_z gives the variance as:
  $$\sigma^2_{T_zT_z} = \frac{e_z^2}{3} \tag{2.9}$$

2.3.2. Vision software: Matching algorithm

In the current teleoperation setup, *Halcon 12* machine vision software package is used for matching point clouds in order to obtain pose information about the objects in the scene/environment. The result of point cloud matching in the Halcon software gives the following information:

- Pose: 6×1 Pose matrix of the Object Found.
- Score: Fraction of points of the model successfully matched in the scene.

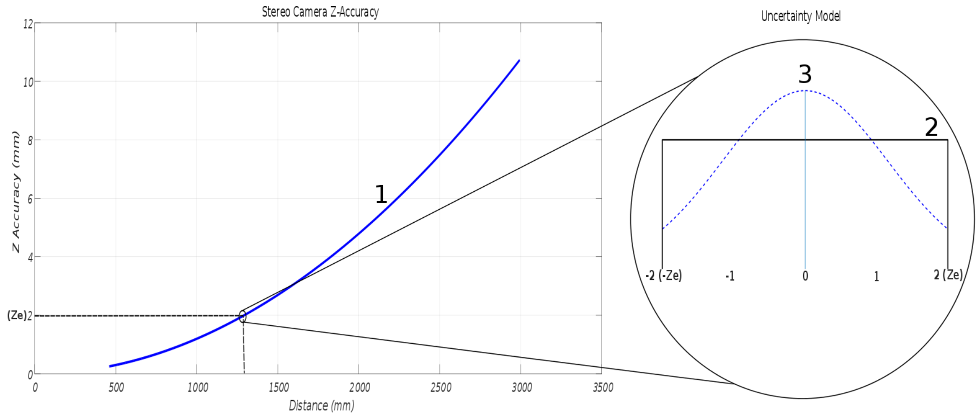


Figure 2.2: Stereo Camera Accuracy Model

A gain matrix based on the score obtained during the matching is proposed as the uncertainty model in this step. The gain matrix is a diagonal matrix with the gain factor as the elements. This gain matrix is multiplied with the previous uncertainty covariance matrix (from stereo camera) to give a resultant measure of uncertainty.

$$Gain\ Matrix\ (G) = Gain \cdot I \tag{2.10}$$

where I: Identity Matrix.
The new covariance can be found by multiplying the covariance matrix with the gain matrix:

$$\Sigma_{new} = \Sigma_{old} \cdot G \tag{2.11}$$

The gain factor is calculated based on the score and has an inverse relationship to the score i.e. if the score is high, the gain in uncertainty is low. Furthermore, the relationship between the gain factor and the point cloud matching score is proposed to be exponential rather than linear. The relationship between gain and matching score can be seen in Fig. 2.3.

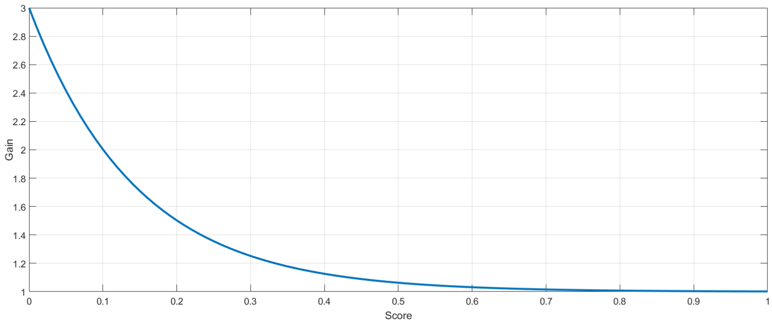


Figure 2.3: Gain Factor and Score

The relationship is exponential is due to the following reasons:

- The gain of any score above 0.5 is assumed to be 1 (or near 1). This is because while obtaining a point cloud for matching, almost half of the object is occluded by itself. Hence, a score of 0.5 is considered as a good match. Thus, if $Score > 0.5$ then $Gain = 1$.
- The behaviour of the gain matrix can be finely tuned to suit application at lower scores by changing the parameters without affecting the gain at higher scores.

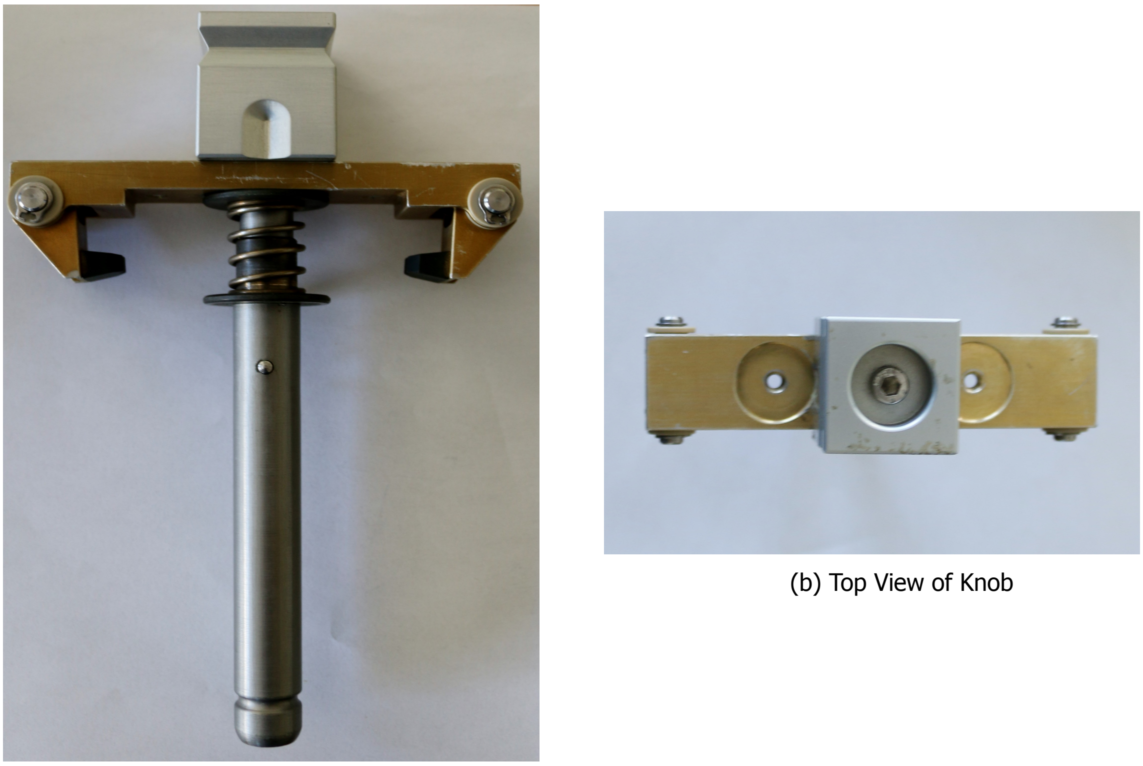
Keeping these reasons in mind, the gain function based on score is selected to be:

$$G = 2 \cdot (0.001^{S}) + 1 \tag{2.12}$$

The point of the highest gain is at 0 score. Theoretically, it is possible to set infinite gain at zero score by changing the function. However, infinite gains are not preferable during implementation of the system. Thus a high gain of 3 is selected along with a flag for 0 score to convey the information that the object is not found in scene.

2.3.3. Experimental Validation

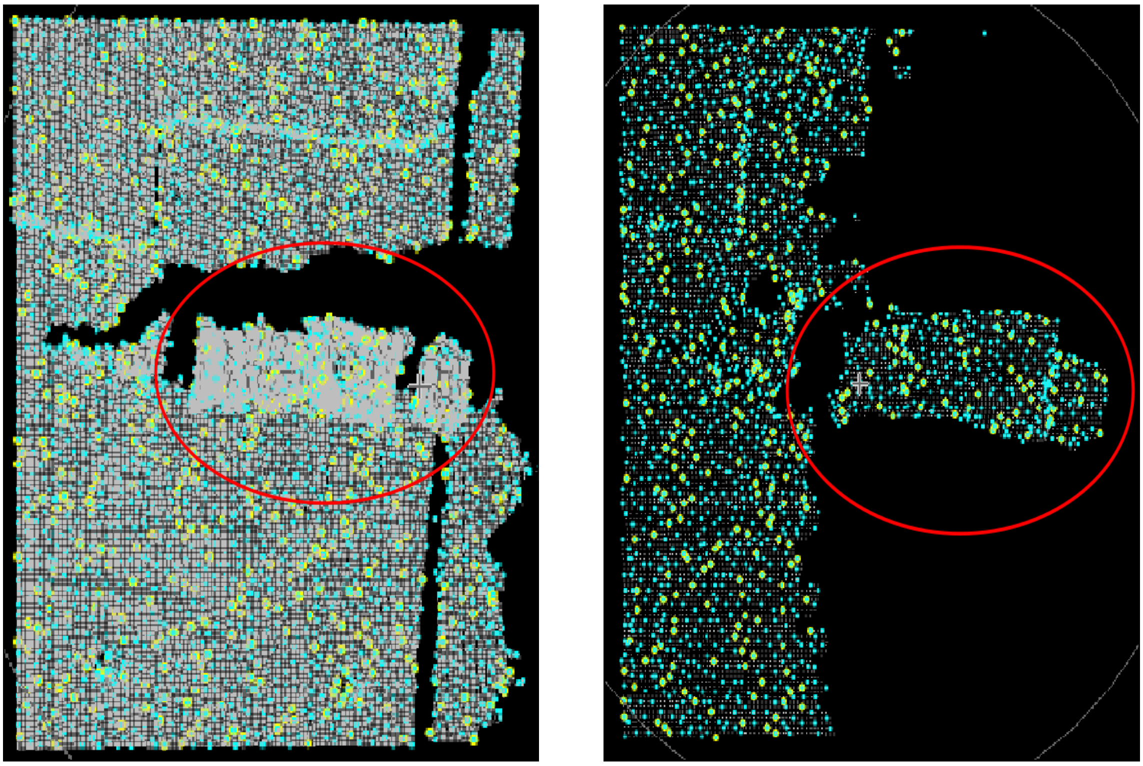
For the validation of the vision system uncertainty estimation, point clouds of an object (knob with handle shown in Fig. 2.4) were recorded at 7 different positions with the Ensenso camera. The relative distance between the camera and the object was varied from $0.655\,m$ to $1.51\,m$. In Fig. 2.5, the point clouds of the knob at the minimum distance ($0.655\,m$) and maximum distance ($1.51\,m$) are shown. From this, it can be seen that the point clouds are more dense and hence more detailed at lower distances than at higher distances as expected. Beyond $1.51\,m$, the knob is found to be undetectable due to the sparse point cloud. A high density point cloud is more accurate for matching when compared to a low density point cloud. Hence the point cloud taken as a reference for matching was the point cloud at $0.65\,m$. The change in density of point clouds is one of the reasons for reduction in accuracy of the vision system as the object is taken further away from the camera.
In the point clouds, the knob was detecting using surface based matching in *Halcon 12*. At each distance, 1000 matches of the static knob were recorded and the outliers were removed using an iterative implementation of the Grubb's Test for Outliers [26].


(a) Side View of Knob

(b) Top View of Knob

Figure 2.4: Knob Used For Validation

From the 1000 matches at each distance ($0.65\,m$, $0.79\,m$, $0.93\,m$,$1.07\,m$, $1.19\,m$, $1.35\,m$, and $1.51\,m$), the mean and standard deviation of the poses were calculated. The standard deviation of the pose is taken as the uncertainty in the pose and this can be compared to the predicted theoretical uncertainty given previously in this section. Due to the nature of the experiment, the uncertainty in pose obtained experimentally is from *both the stereo error and the matching algorithm error*. During the experiments, the view of the knob was controlled so that the matching obtained a high score (> 0.5). This was done to minimize the effect of the matching algorithm on the uncertainty.



(a) Knob Point Cloud at $0.65\,m$

(b) Knob Point Cloud at $1.5\,m$

Figure 2.5: Knob Point Cloud at Various Distances. The red circle indicates the top of the knob with handle.

The histogram results from one of the runs in the experiment (Z Direction, $0.93\,m$) can be seen in Fig. 2.6.

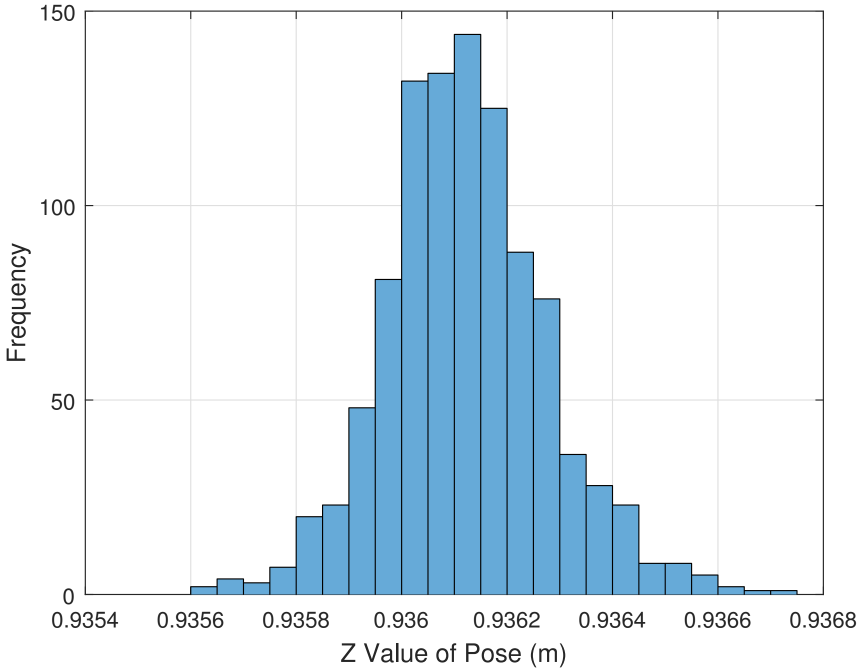


Figure 2.6: Z Component of Pose at $0.93\,m$ Histogram

The behaviour of uncertainty in x, y, z directions with respect to distance from the camera can be obtained by calculating the standard deviations of the measurements at different distance from camera. The theoretical predictions of the uncertainty can be generated using the methods given in Section 2.3. The experimental and predicted theoretical uncertainties can be then compared and this comparison is shows in the following Figures. 2.7, 2.8, and 2.9 for x, y and z directions respectively.

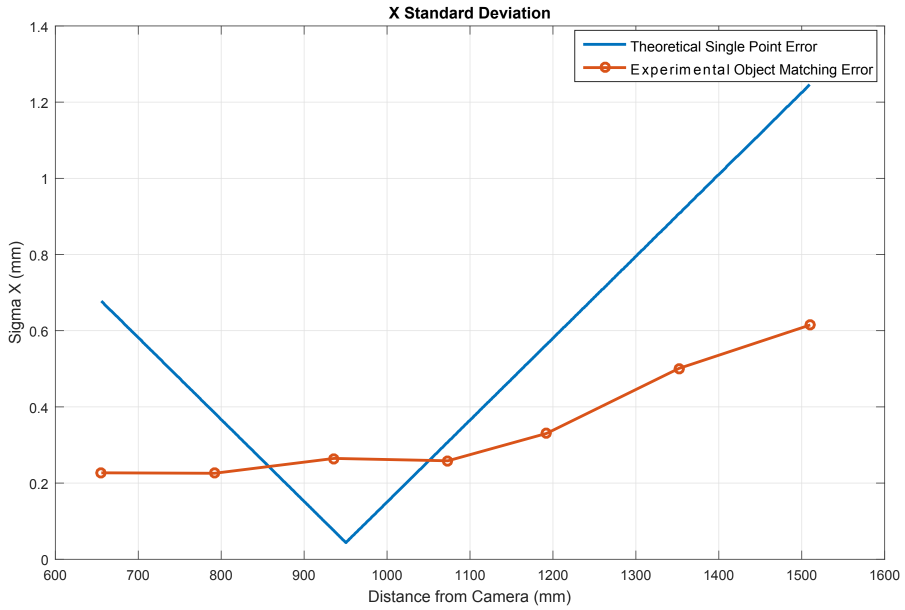


Figure 2.7: Experimental and Predicted Standard Deviations Along X Axis

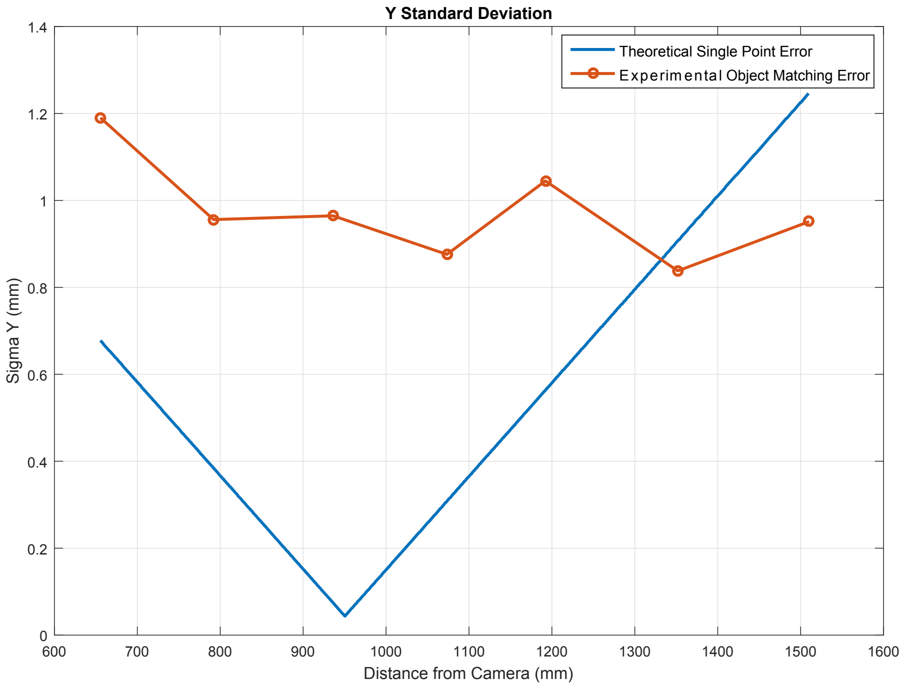


Figure 2.8: Experimental and Predicted Standard Deviations Along Y Axis

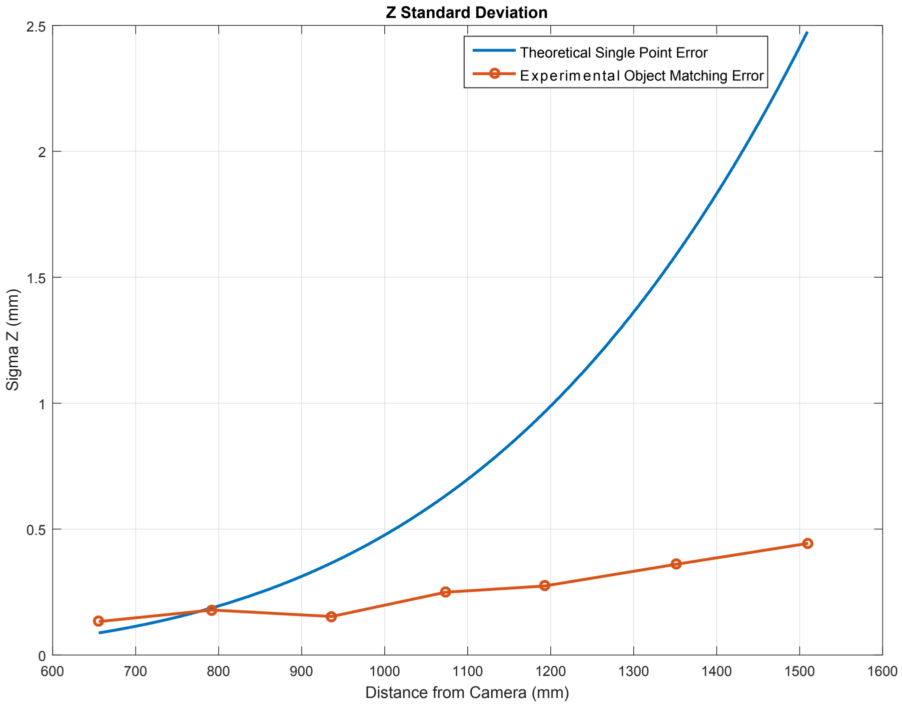


Figure 2.9: Experimental and Predicted Standard Deviations Along Z Axis

It can be seen from the above figures that the experimental and the predicted uncertainties are in the vicinity of each other, however, the behaviour of the experimental and predicted uncertainty with distance is different. It should be noted that the predicted theoretical uncertainty is for a single point matched and estimated by the stereo system, while the experimental uncertainty is obtained after matching objects in the point cloud. The predicted or theoretical errors in x and y reduce till the distance is same as the focus distance of the camera optics at which the optics perform the best. The experiments show a different behaviour suggesting a linear increase of the uncertainty in the x direction and almost stable uncertainty in the y direction (Change of less than $0.4\,mm$). Similarly, in the z direction, theoretical stereo error increases showing a quadratic behaviour, but in experiments, the point cloud matching for object detection linearizes the uncertainty change with distance.

From these validation experiments, it can be concluded that the theoretical error predictions are not far-off from the real uncertainties seen in the experiments. Nevertheless, theory and experiments show a different behaviour of uncertainty with distance. Experiments show a linear relationship of errors with distance after point cloud matching for object detection. Thus, two error models: theoretical and experimental, are available for the stereo vision system. Both can be used for uncertainty estimation in the teleoperation system as they are in vicinity of each other. During the final system validation, the uncertainty model obtained from experiments is used.

2.4. Hand-eye Calibration

Hand-eye calibration gives the transformation between the robot tool frame and the origin frame used by the camera. This process needs to be carried out once and need not be repeated unless the camera, robot, or the mounting hardware is changed. Thus the hand-eye calibration process is carried out once and the poses and covariance matrices are hard-coded for the task application.

Halcon 12 is also used to perform hand-eye calibration of the robot and the camera mounted on the robot. Halcon uses the method of dual-quaternions along with non-linear optimization [27, 28] to solve the $AX = YB$ hand-eye calibration problem. The result of this calibration gives the pose of the camera

in the robot tool frame and the pose of the calibration object in the robot base frame. Furthermore, it also provides the covariance matrices for both these poses.
The hand-eye calibration poses can be seen in Figure. 2.10. The poses in red are the ones determined by solving the hand-eye calibration equation ($X = {}^{cam}H_{tool}$, $Y = {}^{base}H_{cal}$) by using the transformations given in blue and green are the inputs to the calibration process ($A = {}^{cam}H_{cal}$, $B = {}^{base}H_{tool}$). During the calibration process, the robot is moved to multiple configurations and the input poses are recorded. From these, the calibrated poses are calculated along with their covariances.

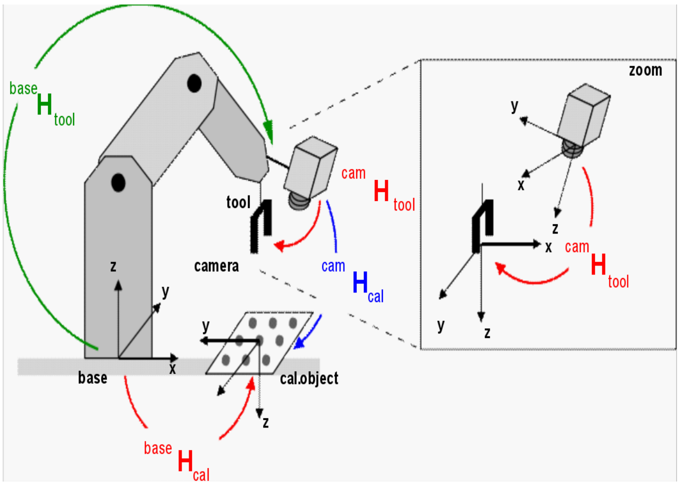


Figure 2.10: Hand Eye Calibration (Source: Halcon Documentation)

The hand-eye calibration was carried out using the procedures given by Halcon. Poses of Object in Camera frame and Tool in Base frame of the robot were recorded at 18 different poses while the object was kept stationary and the camera was moved using the robot arm. These poses were used to calculate the tool pose in camera frame along with its covariance matrix using the hand-eye calibration method given in [27, 28]. The results of the calibration are given below:
The tool pose in camera frame is given as:

$$
{}^{Cam}T_{Tool} = \begin{bmatrix} -0.0469 & 0.9989 & 0 & 0.0398 \\ -0.9989 & -0.0469 & 0 & -0.0135 \\ 0 & 0 & 1.0000 & -0.0647 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{2.13}
$$

The covariance matrix for the above given transformation was calculated and the result is given below. The transnational variances are given in *millimeters* and the rotational variances are given in *degrees*.

$$
\Sigma = \begin{bmatrix} 0.1930 & -0.0000 & 0.0000 & 0.0016 & -0.0009 & -0.0029 \\ -0.0000 & 0.3897 & -0.0000 & 0.0293 & -0.0000 & 0.0047 \\ 0.0000 & -0.0000 & 0.0139 & -0.0024 & -0.0005 & -0.0003 \\ 0.0016 & 0.0293 & -0.0024 & 2.2843 & -0.0003 & 0.2861 \\ -0.0009 & -0.0000 & -0.0005 & -0.0003 & 0.1856 & -0.0248 \\ -0.0029 & 0.0047 & -0.0003 & 0.2861 & -0.0248 & 0.2740 \end{bmatrix} \tag{2.14}
$$

2.5. Robot and Motion Capture System Error Models

Uncertainty models of all the components of the teleoperation system cannot be theoretically characterized. Such is the case for the Kuka LWR 4+ robot used in the current setup. Furthermore, the Motion Capture System (MCS) - *Vicon* used in the *INTERACT* experiment [4–6, 13] also does not have an error model associated with it. Since the uncertainty estimation system being developed in this thesis is aimed at replacing the MCS, its accuracy must be understood to enable comparison.

In the following experiments, the accuracy (error distribution) of the robot and the MCS is found using an *Optical Coordinate Measuring Machine (CMM)* as ground truth. The accuracy of Nikon K600 optical CMM in X, Y, and Z directions is given as $15, 15$, and $60\mu m$ (2σ) respectively by the manufacturer.

2.5.1. Experimental Method

Multiple experiments were carried out on the Vicon MCS and the Kuka LBR4+ robot. These experiments were aimed at finding the accuracy of the robot in different volumes in its workspace and also determining the accuracy of the MCS. These systems were evaluated based on the stability of their measurements over time and the accuracy of their measurements in a given volume. For all the experiments, markers were used for the Optical CMM and the Vicon camera system. These markers were different for different systems:

- Passive Infrared Markers for Vicon MCS.
- Active infrared LED markers for the Nikon K600 optical CMM.

As the markers of one system are not recognized by the other system, it can be inferred that the markers do not cause interference between the system or reduce the measurement accuracy. To measure the accuracy in a given volume, the markers were placed on a peg tool mounted on the robot arm. The markers along with the peg mounted on the robot arm can be seen in Fig. 2.11.

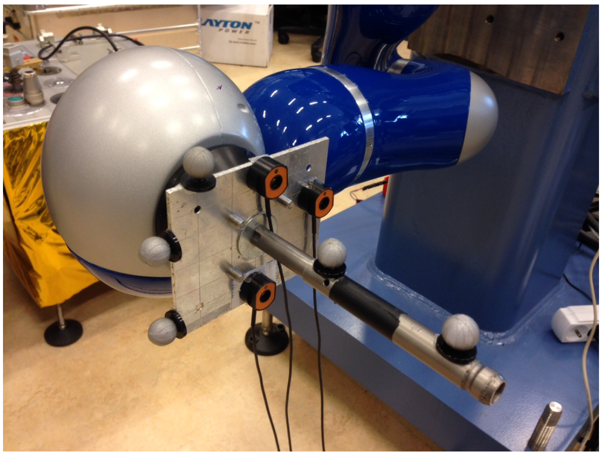


Figure 2.11: Markers on Peg Tool Mounted on Robot

It is shown in [29] that the root mean square error of the vicon camera system reduces as the number of cameras that can view the markers is increased. However, this effect reaches a saturation value when using more than 5 cameras that can see the marker. Hence, for all the experiments, 6 vicon cameras were used and all the markers were visible to all the cameras during all the experiments. This is done to exclude the number of cameras and their viewing configuration as a influencing factor from these experiments.

The readings from the 3 systems: Kuka, Vicon, and Nikon were captured on three different computers and analyzed using Matlab. The frequency of measurements are as follows: Robot arm: $60Hz$, Vicon: $200Hz$, and Optical CMM: $5Hz$. Higher frequency data were sub-sampled to the lowest frequency $(5Hz)$ by using nearest neighbor interpolation. Both the Vicon and the Nikon K600 were calibrated according to the methods specified by the manufacturers before the experiments. The room temperature was kept constant at $24 \pm 1^{\circ}C$.

Two different types of experiments were carried out. These are described as following:

A. Stability of Measurements

To examine the stability of the readings given by all the three systems, they measure a static object over time. The recommended warm-up time from the manufacturer was not followed in order to see the trend in the measurements starting from the switch-on time of the instruments. Along with the measurements recorded by the devices, the temperatures of the devices were recorded using a FLIR i3 infrared thermometer [30]. The robot arm was kept static as it measured its position and the passive and active markers were kept on a rigid static object while the measurements were made.

B. Accuracy Evaluation

The peg with the markers (seen in Fig. 2.11) was moved using the robot to various positions in a $20 \times 20 \times 20cm^3$ volume. The selected poses were the corners of the volume and the paths taken were the edges and the diagonals of the cubic volume. While moving to these poses, the peg tool's movement was measured using the robot arm, the MCS, and the Optical CMM as shown in Fig. 2.13. The recording of these measurements was not initially synchronized as the recording was started within 30 seconds on each of the computers. This was accounted for in the data conditioning later.

The robot was operated in impedance control mode with high linear and angular stiffness values: 3500 Nm and 250 Nm/rad respectively. Position verification was implemented to check if the robot was in the vicinity of the desired pose i.e < 1cm and $< 1^{\circ}$. When this condition was achieved, the robot was kept static for 15 seconds and then the command was issued to move to the next pose. This process was carried out for all the 9 poses in one run. Multiple runs were carried out per measurement volume with the total number of runs being 10. Table. 2.1 shows the volumes, runs and the octant [31] of the workspace of robot the runs were carried out in. The robot was mounted in the $+Y$ direction and this reference is used to describe the octants. The mounting of the robot with respect to the octants can be seen in Fig. 2.12. The octants III, IV, VII, and VIII are behind the robot and hence not applicable. The volumes were placed on either side of the robot in octants V and VI to ensure that the effects caused by the gravity vector could be taken into account.

Volume Number	*No. of Runs*	*Octant*
1	2	V
2	2	V
3	5	VI

Table 2.1: Volumes and Octant

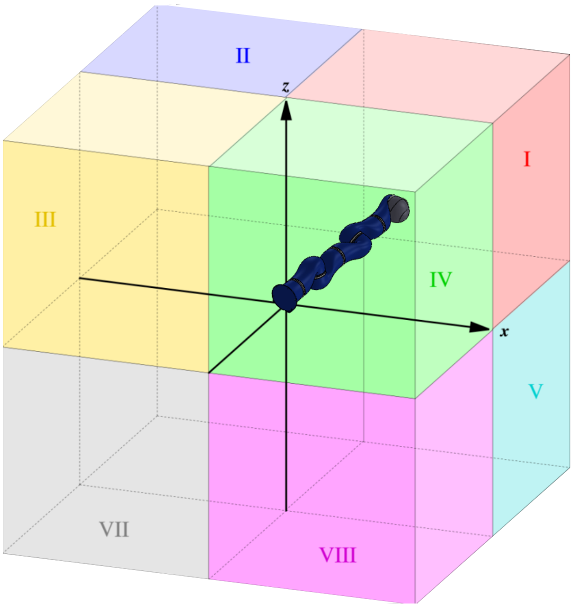


Figure 2.12: Robot Mounting and Octants

The pose of the peg mounted on the robot was measured by all the 3 systems throughout the whole experiment. The placement of the robot arm, vicon cameras and the K600 optical CMM can be seen in Fig. 2.13.

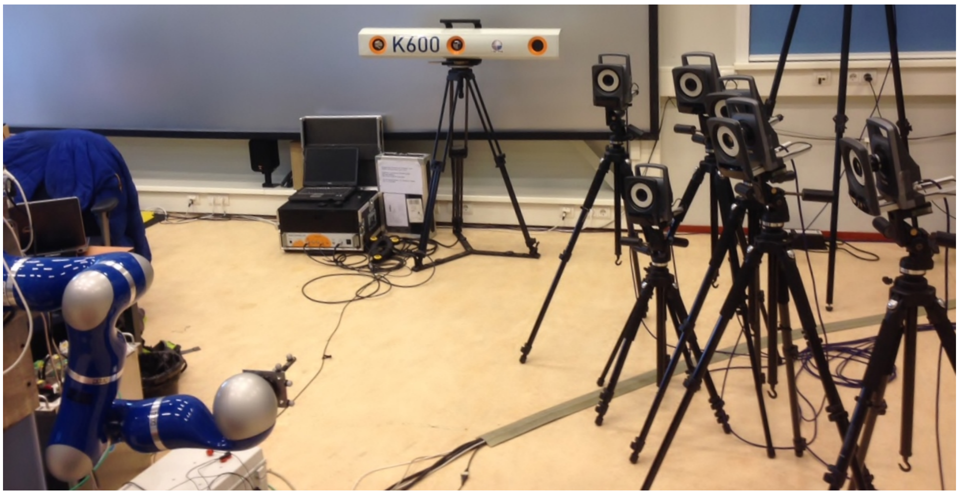


Figure 2.13: Experimental Setup

The experiment was repeated multiple times for each volume location in the workspace of the robot. The first two runs in *Volume 1* were carried out immediately after each other. These were done after the robot and Vicon were switched on for several hours (> 4 Hours). The same method is followed for the five runs in *Volume 3*. The two runs in *Volume 2* were carried out with few hours in-between the experiments. The first one was carried out after $60min$ of switching on the instruments and the second was done 2 hours after the first experiment.

Data Conditioning

After the raw data was obtained, the post-processing steps applied on the data are explained in the steps given:

- Offset the robot tool frame pose measurements by $2.858cm$ so that it matches the peg plane on which the markers are mounted.
- Move origin to the first data point.
- Calculate distances from the origin (1st position measured).
- Interpolate all data to lowest frequency ($5Hz$) using nearest neighbor interpolation.
- Perform synchronization of data by observing the graph and shifting the data to synchronize the start of movement.
- Find slope of distance at every timestep and check if slope was below a minimum threshold for more than 10 seconds. These are the stationary poses at the corners of the cubic volume.
- Find mean distance and mean error (ME) at the stationary pose with respect to the initial pose. ME is defined as the average difference between the distances measured by robot/MCS and the Optical CMM during 15 seconds at each pose.

2.5.2. Kuka LBR 4+ Robot

A. Stability of Measurements

The robot arm was kept stationary (using internal brakes) for 66 minutes and the drift was measured as the distance with respect to the initial position measurement. The measurements were sub-sampled using nearest neighbor interpolation to a frequency of $1Hz$. The plot of the drift can be seen in Fig. 2.14. From the figure, it can be seen that the robot arm reports a drift in its measured pose by about $0.0045mm$ in about 50 minutes and settles. This can be modeled as a first order system with a rise time of 1350 seconds for 10% and 90% of maximum value.

The average temperatures of the joints of the robot arm during the experiment can be seen in Table. 2.2.

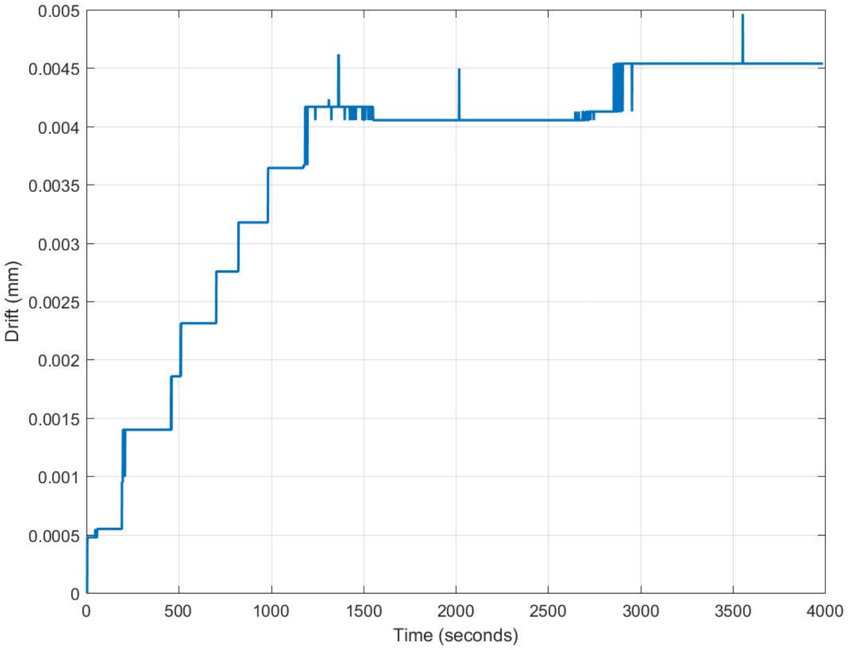


Figure 2.14: Kuka LBR4+ Drift with Time

Time (Minutes)	*Temperature (°C)*
0	24.5
20	34
45	46
60	48

Table 2.2: Kuka LBR4+ Temperatures

B. Accuracy Evaluation

Following the procedures given in Section. 2.5.1, the Mean Error (ME) was obtained for each pose during all the runs in the octants V and VI. A sample from the results can be seen in Fig. 2.15. These are from the runs performed in *Volume 1* and *2*.

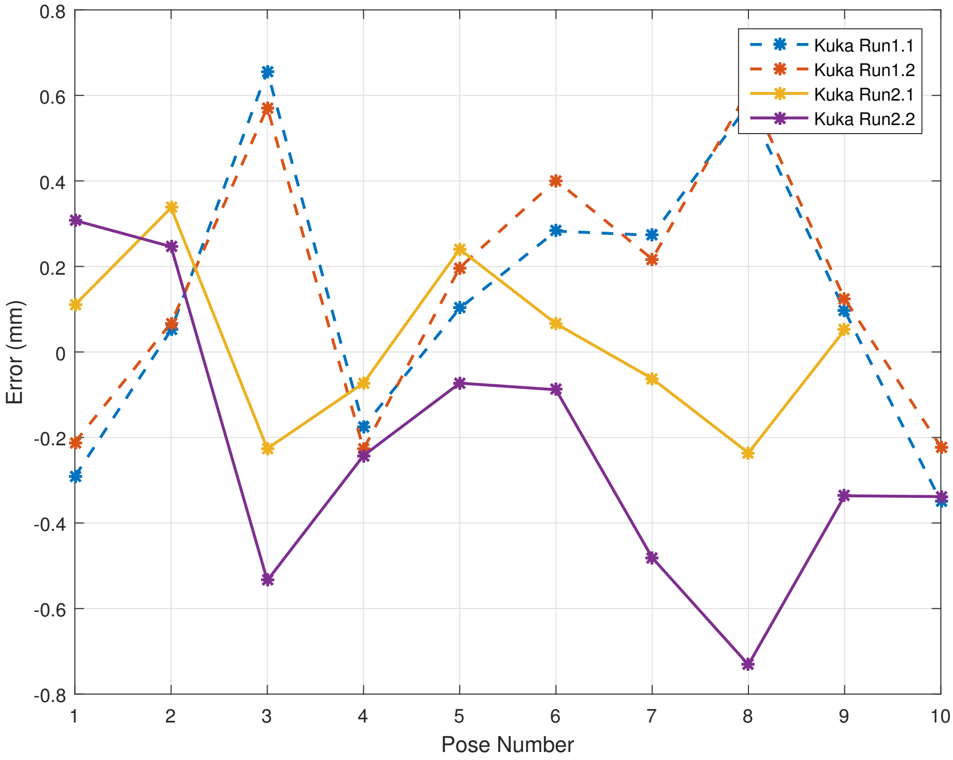


Figure 2.15: Accuracy Results from 2 Runs in each *Volume 1* and *2*

Errors from all the runs in each volume were combined together. These results were verified to come from a normal distribution using the Lilliefor's Test [32]. Subsequently, a Two-Sample Kolmogorov-Smirnov test (5% Significance) was used to confirm that the errors from different runs come from the same distribution.
To verify if the robot's accuracy distribution was affected by the configuration of the robot or the gravity vector in impedance control mode, error distributions from both octants were compared. The measurements in octants V and VI were done with the gravity vector changing its direction along with the configuration of the robot. The resulting histogram of errors can be seen in Fig. 2.16. From the histograms, a significant overlap can be seen in accuracy data from octants V and VI. Furthermore, the accuracy results from different volumes were subjected to the Two-Sample Kolmogorov-Smirnov test and they pass the test validating the notion that errors in both octants/configurations are from the same distribution. Hence, the errors from both octants can be combined to give the final error distribution of the robot.

The final error distribution of the Kuka LBR 4+ robot arm is obtained by combining all the results from the runs done in the experiments. A Gaussian distribution is then fit onto this result. This can be seen in Fig. 2.17. The mean and standard deviation of this distribution is calculated as:

$$\mu_{Kuka} = 0.0179\ mm \tag{2.15}$$

$$\sigma_{Kuka} = 0.2728\ mm \tag{2.16}$$

The mean of the results is lower than the precision at which the measurements can be made with the Optical CMM, hence the mean can be assumed to be *Zero* (0).

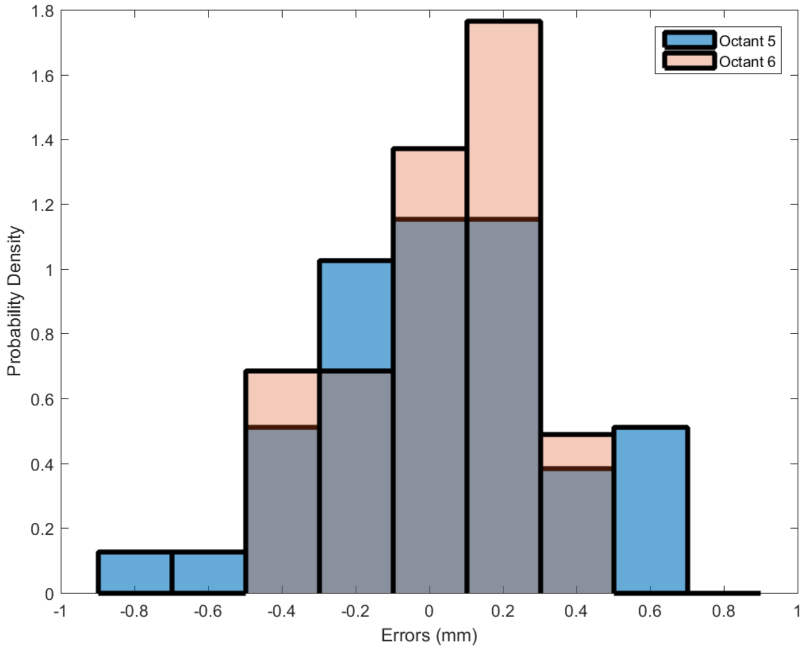


Figure 2.16: Histograms of Robot Errors in *Octants V,VI*

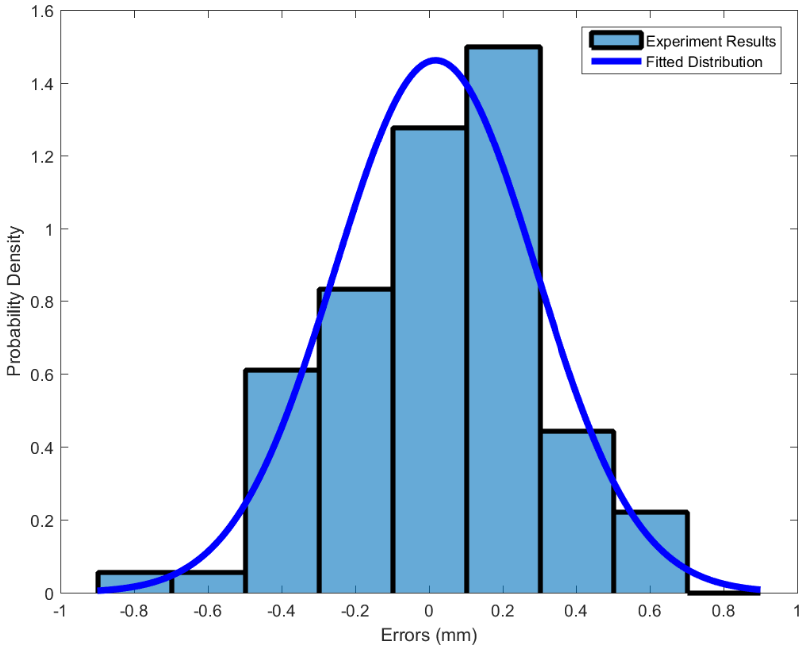


Figure 2.17: Kuka Robot Arm Error Distribution

Since these are the total errors in distance measured by the robot, the errors in individual axis are taken as:

$$\sigma^2_{T_xT_x} = \sigma^2_{T_yT_y} = \sigma^2_{T_zT_z} = \frac{\sigma^2_{kuka}}{3} = 0.0248mm \tag{2.17}$$

It can be seen that the translation distance errors are very small in magnitude. From this, it is predicted that the rotation errors are also small and hence they are assumed to be insignificant in magnitude. It is difficult to measure the rotational errors due to difficulties in alignment of axis of different measurement systems. The alignment of the axes of different measurement systems introduce larger errors than the ones being measures. Due to this and the fact that errors seen from the experiments are of low magnitude, the rotational errors from kuka pose feedback are assumed to be zero or near zero and hence do not contribute to its covariance matrix.

2.5.3. Vicon Motion Capture System

A. Stability of Measurements

The passive markers for the vicon system were attached to a rigid stationary body. These markers were used to create a local frame using the vicon tracker software [33]. The pose of this frame was measured using the system for over 2 hours. The drift was measured as the distance with respect to the initial position measurement. This data was not sub-sampled. Fig. 2.18 shows total drift measured and Fig. 2.19 shows the drift measured in individual axis of measurement.

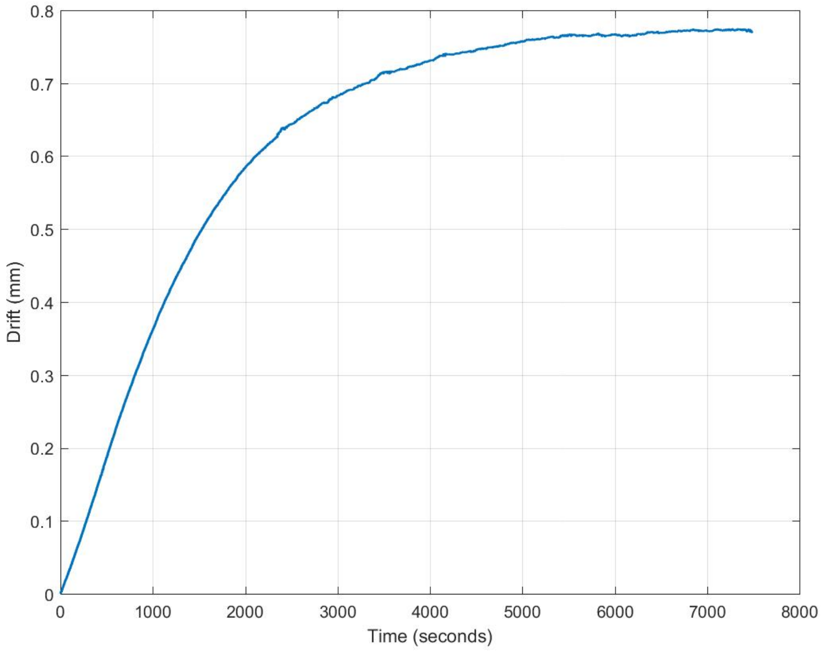


Figure 2.18: Total Drift of Vicon System

From the total drift seen in Fig. 2.18, it can be seen that this system can be modeled as a first order system with a rise time of 3200 seconds for 10% and 90% of maximum value. The average temperatures of the vicon cameras during the experiment can be seen in Table. 2.3.

B. Accuracy Evaluation

Following the experimental method given in sub-section 2.5.1, passive infrared markers were placed on the peg-tool attached to the robot end effector and 6 Vicon MCS cameras were used to track the movements of a reference frame made from the markers. This can be seen in Fig. 2.13. The Nikon K600 optical CMM was taken as the ground truth here as well.

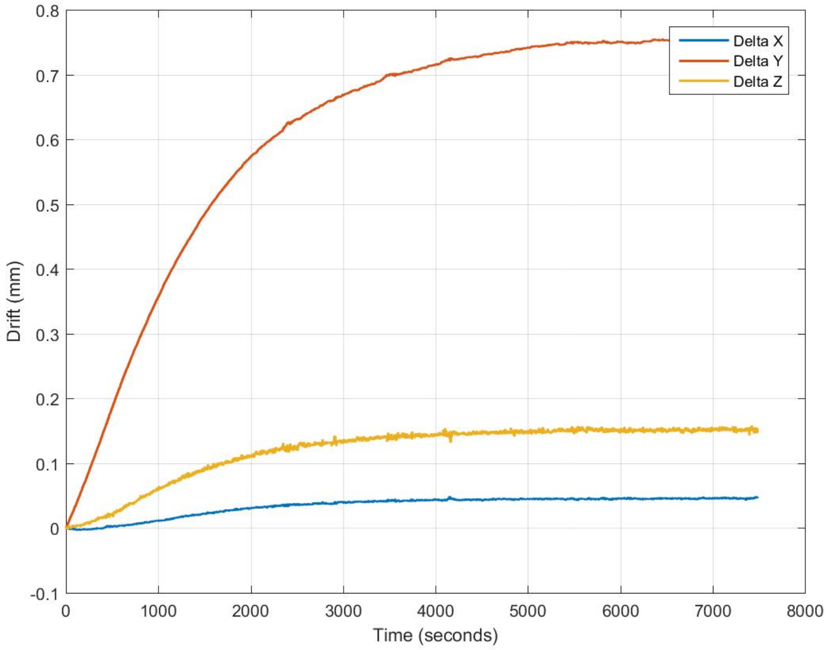


Figure 2.19: Drift in individual axis of Vicon System

Time (Minutes)	*Temperature (°C)*
0	24.5
20	41
45	58
60	59.5
120	60

Table 2.3: Vicon Camera Temperatures

The errors from different runs were subjected to Lilliefor's test for normality [32] and the result was that they all come from a normal distribution. Furthermore, a Two-Sample Kolmogorov-Smirnov test (5% Significance) was used to confirm that the errors from different runs come from the same distribution. Knowing the results from these tests, the data from all the runs were combined to give the error distribution for the *Vicon* Motion Capture System. From this a Gaussian distribution was fit to the data using the following calculated parameters:

$$\mu_{MCS} = 0.0863 \tag{2.18}$$

$$\sigma_{MCS} = 0.2323 \tag{2.19}$$

A histogram of the experimental data and the fitted Gaussian distribution can be seen in Fig. 2.20.

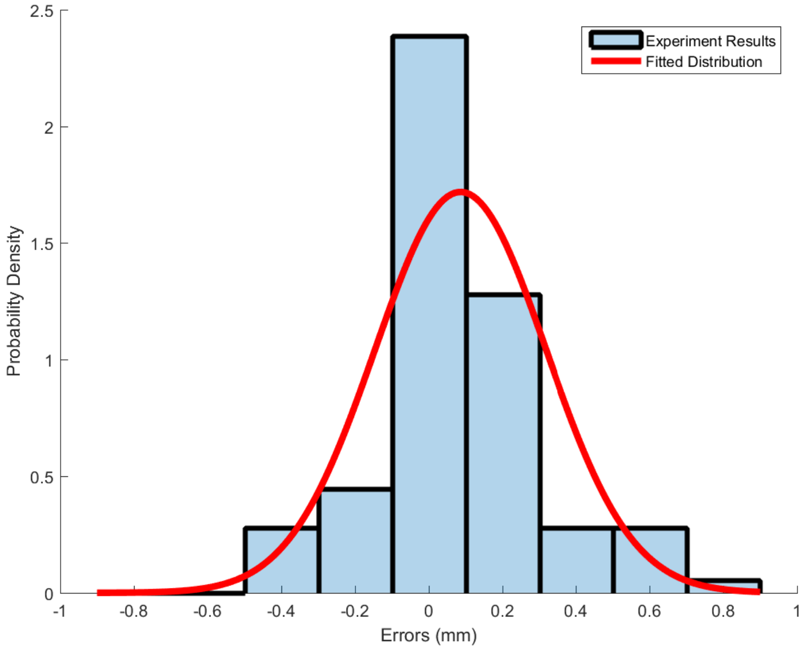


Figure 2.20: Vicon MCS Error Distribution

2.5.4. Comparison Between Kuka Robot Arm and Vicon Motion Capture System

Since the accuracy/error distributions were available for both the robot and the MCS, a comparison could be made between them. A simple comparison was made by plotting the histograms of both errors simultaneously to understand the error distributions. This can be seen in Fig. 2.21.
Along with this, the normal distributions of the errors can be described using mean and standard deviation. The standard deviations of the robot arm and MCS are 0.2759 mm and 0.2334 mm respectively. Using these standard deviations, the probability density functions of the Gaussian distribution with these parameters were plotted over the experimental results in Fig. 2.21. The effect of the standard deviations can be seen in Fig. 2.21 as the robot arm distribution is more spread when compared to MCS. However, it can be seen that the robot arm and the MCS have comparable accuracy.

A table of the mean, maximum and minimum of the absolute errors of the robot arm and vicon are given in Table. 2.4.

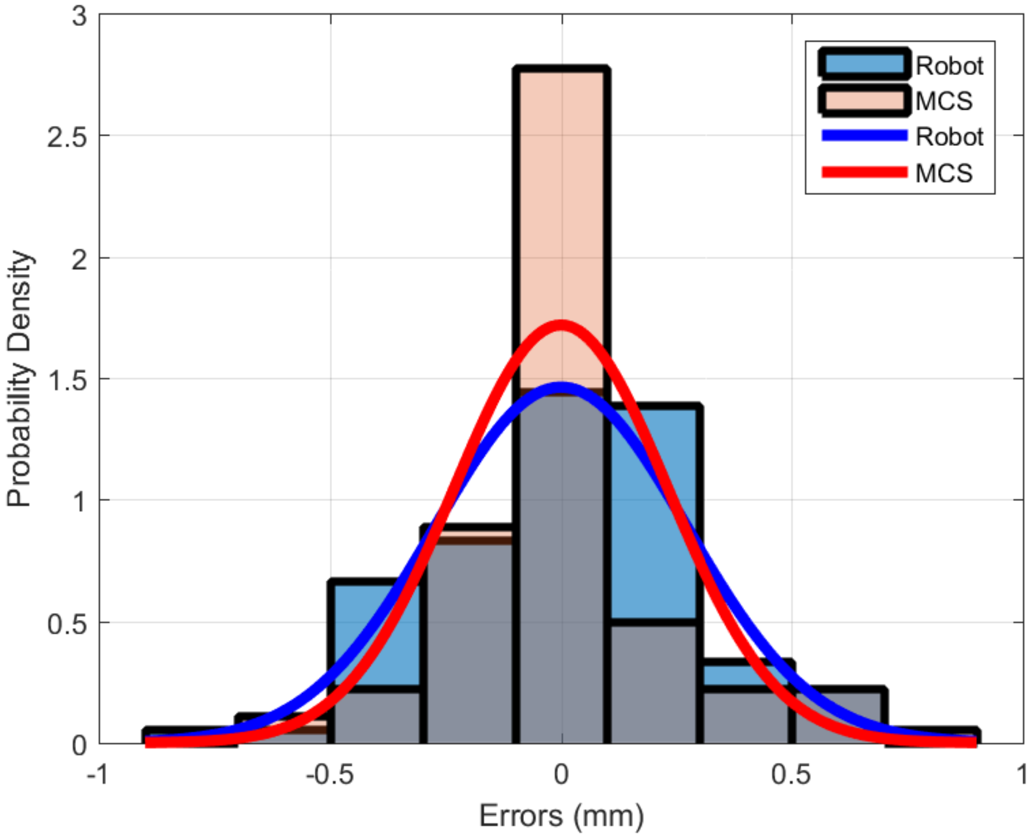


Figure 2.21: Histograms of Robot and MCS Errors

Table 2.4: Errors in Both Systems

	Min (mm)	*MAE (mm)*	*Max (mm)*	*RMSE (mm)*
Robot Arm	0.0078	0.2252	0.7312	0.2750
MCS	0.0011	0.1730	0.8574	0.2479

2.6. Robot-Robot Base Calibration

In the current teleoperation setup, the camera and the gripper are mounted on two different robot arms. Due to this setup, pose of the object detected by the camera mounted on one robot arm has to be transformed into the relative pose between the object and the gripper mounted on another arm. To perform this transformation, the spatial relationship between the base frames of the robot is required. One way to find the relationship between the two robot bases would be to find a spatial relationship (transformation) with a fame common to to both the bases. For the robot used for this thesis: Kuka LBR 4+, the manufacturer Kuka provides a method to calibrate an external frame as the robot base frame. This gives the spatial relationship between the internal base frame of the robot and the external frame. The same external frame can be used as the base frame for both robots, thus giving a relationship between the frames as shown in the equation below:

$$^{Base1}T_{Base2} = {}^{Base1}T_{Ext} \cdot ({}^{Base2}T_{Ext})^{-1} \tag{2.20}$$

This relationship can be used to find the spatial transform between the base frames and also to find the uncertainty in this transform. This can be done using the method of error propagation (from Chapter 3) if the uncertainty in individual transformations is known.

To perform external base calibration, a calibrated tool is required. For this purpose, a conical tool was calibrated in accordance with the procedures given in documentation given by Kuka. The calibration is called *XYZ 4-Point* calibration in which the tool is moved to the reference point from 4 different directions and from these the transnational parameters are obtained. A tool in the shape of a conical frustum was used for the calibration. The diameter of the tool tip was measured to be $6.5\,mm$. On the tool tip, a point was marked with a pen. The calibration was carried out by aligning the marked point with a reference point from 4 different directions. The result of the calibration provides the relative distances in X, Y, Z directions and the error in calibration. The calibration was carried out multiple times and the results can be seen in Table. 2.5. From these, results from the calibration run with the lowest error (Run 7) was chosen for base calibration.

Run	*X (mm)*	*Y (mm)*	*Z(mm)*	*Error (mm)*
1	0.501	1.195	113.018	1.039
2	-0.004	0.92	112.964	0.894
3	0.135	1.03	113.183	0.314
4	0.267	1.004	113.2	0.62
5	-0.002	0.93	113.258	0.9958
6	0.116	0.715	113.566	0.356
7	**0.043**	**0.202**	**113.339**	**0.299**
8	0.488	0.387	113.39	0.692

Table 2.5: Tool Calibration Runs: Cone Tip Dia: 6.5mm

For the external base calibration, the robot documentation gives a *3-Point Method* in which the calibrated tool is moved to 3 points in the workspace: Origin, 1 point on X-Axis, and 1 point on XY plane. This gives the transformation between the external frame and the internal base frame of the robot. This calibration process was carried out 10 times and the results of the calibration can be seen in Table. 2.6. From these runs, the standard deviation of the distance in the transformation was found to be $0.3652\,mm$ which is in the vicinity to the standard distribution of error in the robot kinematics: $0.2728\,mm$. From this, it can be hypothesized that the error in the external base calibration depends on on the tool calibration error and the robot kinematics error. In an ideal scenario, the base calibration error would be equal to the robot kinematics error.

Thus, the external base calibration error is calculated experimentally for a single case and an ideal error is hypothesized. These are the uncertainties in transformations given in equation 2.20 and can be propagated through the transformations to obtain the total error in Robot-Robot base calibration.

Run	*X (mm)*	*Y (mm)*	*Z(mm)*	*RotZ*	*RotY*	*RotX*
1	-54.5230	167.2360	972.0440	-90.8370	-0.4370	-81.4010
2	-54.7910	168.9060	972.2320	-90.6960	-0.6420	-81.6140
3	-54.8600	168.2400	972.2470	-90.6480	-0.9460	-81.5250
4	-54.8930	168.1190	973.2040	-90.6130	-0.6350	-81.6060
5	-54.4150	168.9420	972.2270	-90.7280	-0.7430	-81.6320
6	-54.6090	168.5720	972.8510	-90.6280	-0.7350	-81.5560
7	-54.6610	168.1890	972.5970	-90.7200	-0.5690	-81.4410
8	-54.8410	168.9600	972.5310	-90.6610	-0.7070	-81.7610
9	-54.7820	168.7640	972.3460	-90.6640	-0.7180	-81.6370
10	-54.3560	168.6970	972.6640	-90.7400	-0.4060	-81.6920
σ	0.1918	0.5353	0.3492	0.0662	0.1754	0.1094

Table 2.6: Base Calibration Runs with Standard Deviations

2.7. Conclusion

In this chapter, the uncertainties in all the transformations in the teleoperation system were studied and modelled. Apart from for the uncertainty in the stereo vision system, the errors in the system are static and do not change during operation. The dynamic behaviour of the stereo vision system is modelled using the distance of the object detecting and the score of the match. This is validated experimentally. The uncertainty models for other transformations are static and are determined by carrying out various experiments.

Thus, all the individual uncertainty models in the system are represented using the covariance matrix and these are prepared to be propagated through the system.

3

Uncertainty Propagation

In this Chapter, the method of error propagation is devised and validated and its application to the vision-aided teleoperation system is discussed.

3.1. Problem

The pose of the object being manipulated is sent from the sensor (3D/2D camera) mounted on the end effector of a robot to the gripper attached to another robot arm for manipulation. During this transformation of the pose, there are various transformations in chain. These transformations are not an exact representation. In the previous chapter, the errors and uncertainties in these transformations were studied. Along with this, a method to represent the uncertainties using covariance matrices of poses was provided.

Once the errors and uncertainties in each transformation are known, they are compounded (or *"added-up"*) to give the final uncertainty of the object in the gripper frame. Due to the nonlinear nature of spatial transforms, the final uncertainty covariance matrix is not the result of an addition of all the covariance matrices as they are in different reference frames.

To find the final uncertainty in the object pose, a robust method of error propagation is needed which compounds the errors in individual transformations to give a final estimate.

Furthermore, a method to invert this process is desired. This can be used to determine individual component specifications (Robot, Camera Specifications) from the final accuracy requirements of the complete system.

3.2. Approach

In the chain of transformations in a robot-vision system, each spatial transform is treated as a measurement. Some measurements, such as the Tool-in-Base transform and object-in-camera transform are dynamic measurements which change as the robot is moved in the workspace. Other measurements such as Hand-eye calibration of the camera and Robot-Robot base calibration are static measurements which are needed to be carried out once and are then constant during operation.

Following the *Guide to expression of uncertainty in measurement* by ISO [24], these measurements (spatial transformations) should be reported with a quantitative indication of the quality of the measurement so that the users can judge its reliability. In the case of the robot vision system, a measurement of spatial transformations is given by its mean, which is represented by a 4×4 Homogeneous Transformation matrix and an uncertainty term which is the 6×6 Covariance Matrix as described in the previous chapter.

Thus, every transform in the robot-vision system is described by a set of a transformation matrix and a covariance matrix:

$$(T, \Sigma) \tag{3.1}$$

Thus, while composing transformation matrices the covariance matrices are also compounded using an error propagation scheme. For example, if two spatial transformations can be represented as:

$$(T_1, \Sigma_1), (T_2, \Sigma_2) \tag{3.2}$$

The composite transformation consists of a transformation matrix and the resulting covariance matrix which can be expressed as a compound function as shown:

$$(T_3, \Sigma_3) = compound((T_1, \Sigma_1), (T_2, \Sigma_2)) \tag{3.3}$$

While the transformation matrix is calculated by multiplying the mean transformations, the final covariance is computed using the compound function which is obtained from error propagation scheme described in the following section.

3.3. Error Propagation

An error propagation scheme is proposed in this section. This method given here is adapted from the method originally proposed by Su and Lee in [34] which focuses on robotic assembly tasks and does not include the error models of sensors and actuators. The authors focus on the uncertainty in spatial transformations and how they can be propagated to find the probability of achieving the goal in the given task. In the current work, the focus is kept on propagating error models obtained from the sensors and actuators. These form a part of an online error estimation framework. This provides an extension to the work done by Su and Lee in [34] by running the error propagation online during the teleoperation tasks and updating the inputs used in propagation based on error models of the components used in the teleoperation setup (Robot Arm, Stereo Camera, etc).
To begin with the error propagation, the concept of differential transforms [35] is introduced. The differential transform is used to represent a small error in a spatial transform. This is used in the process of deriving the error propagation method. A pose vector is represented as $\begin{bmatrix} x & y & z & \theta_x & \theta_y & \theta_z \end{bmatrix}$, a small error in this pose can be represented as $\begin{bmatrix} d_x & d_y & d_z & \delta_x & \delta_y & \delta_z \end{bmatrix}$ where d and δ are translation and rotation errors respectively. Assuming the errors are small, the differential transform for errors in the pose can be written as (from [35]):

$$\Delta_T = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.4}$$

This can be used to model the error in a nominal transform to get the real transform as (given in [34]):

$$T_{real} = T_{nom}(I + \Delta_T) \tag{3.5}$$

Where I is the Identity matrix and Δ_T is the differential transform representing the errors in the pose. It is decided to post-multiply the error term in this case so that the errors are expressed in the object frame rather than the base frame. This difference can be seen in Fig. 3.1. This is of special interest in this thesis as the errors are needed the frame of the object of interest, not in the base frame.

The differential transform in object frame can be manipulated to find the error or differential transform in the base frame using the relation (from [35]):

$$\Delta_{T_{object}} = T_{nom}^{-1} \Delta_{T_{base}} T_{nom} \tag{3.6}$$

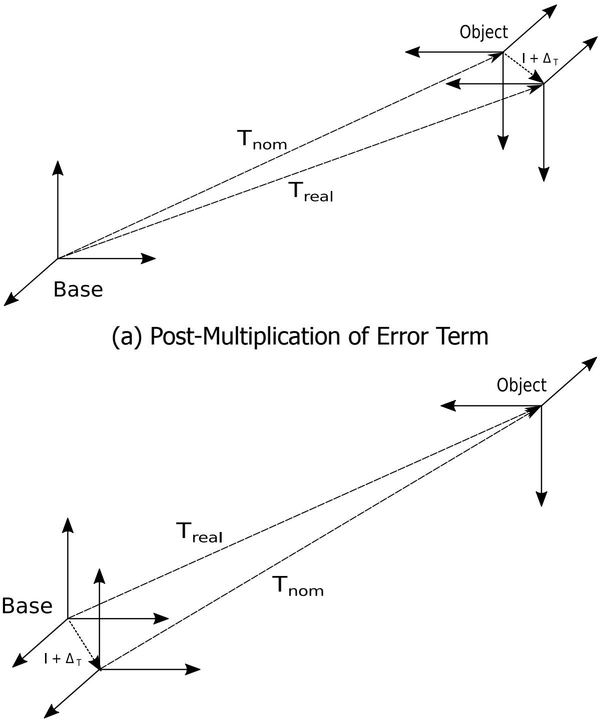
Each transformation matrix can be seen as a non-linear function transforming input position and orientation into an output position and orientation. Let the input be $x \epsilon \mathbb{R}^6$ and the transformation function be $f : \mathbb{R}^6 \rightarrow \mathbb{R}^6$. Using the first order Taylor series approximation, the transformation function can be estimated as:

$$y = f(x) \approx f(\mu_x) + J(x - \mu_x) \tag{3.7}$$

where $y \epsilon \mathbb{R}^6$ is the output, μ_x is the mean of the input x, and $J \epsilon \mathbb{R}^{6\times 6}$ is the jacobian matrix $\partial f / \partial x$ evaluated at μ_x .
From this it can be seen that the mean of the output vector is equal to the first term of the approximation and the covariance of the output is related to the second term [36]:

$$\mu_y \approx f(\mu_x) \tag{3.8}$$

$$\Sigma_y \approx J \Sigma_x J^T \tag{3.9}$$



(a) Post-Multiplication of Error Term

(b) Pre-Multiplication of Error Term

Figure 3.1: Effect of Pre-Multiplication and Post-Multiplication of Errors

To use these results in propagation of covariance matrices, the concept of *Spatial Jacobian* is introduced. This is the jacobian of the T_{nom} matrix [34, 35]. The *Spatial Jacobian* is different from the more generally used *Kinematic Jacobian* in robotics. The *Kinematic Jacobian* relates properties in joint space such as Angular Velocity and Torques to the properties in task space such as Spatial Velocity and Forces. However, the *Spatial Jacobian* is used to transform the errors from one coordinate from to another. From [35] and [34] it can be seen that Equation. 3.6 can also be represented using the following relationship:

$$\epsilon = J\epsilon' \tag{3.10}$$

where:

ϵ - 6×1 Mean Error Vector in Object Frame
J - Spatial Jacobian of T_{nom}
ϵ' - 6×1 Mean Error Vector in Base Frame

The 6×6 spatial jacobian matrix can be constructed from the T_{nom} matrix as:

$$J = \begin{bmatrix} R_n^T & R_n^T D_n^T \\ 0 & R_n^T \end{bmatrix} \tag{3.11}$$

where:

R_n^T - Transpose of the Rotation Matrix part of T_{nom}

D_n - Skew-Symmetric Matrix of the T_{nom} pose vector $= \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$

From Equation. 3.9, the covariance of the mean error vectors ϵ and ϵ' in Equation. 3.10 can be written as:

$$\Sigma_\epsilon = J\Sigma_{\epsilon'}J^T \tag{3.12}$$

A simple chain of two transformations can be represented by (T_1, Σ_1) and (T_2, Σ_2) as shown in Fig. 3.2. Using differential transforms instead of covariance matrices, the transformations can be represented using the following equations:

$$T_1 = T_{1n}(I + \Delta_1) \tag{3.13}$$

$$T_2 = T_{2n}(I + \Delta_2) \tag{3.14}$$

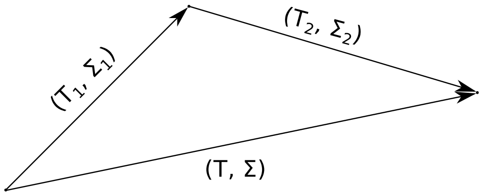


Figure 3.2: Simple Transform with Uncertainties

From the above equations, it can be seen that the nominal transform can be taken as the mean of the measurement while $(I + \Delta)$ is the uncertainty term. To find the resultant transform (T, Σ), the differential transform equations are multiplied as shown below:

$$T = T_1T_2 = (T_{1n}(I + \Delta_1))(T_{2n}(I + \Delta_2)) \tag{3.15}$$

$$T = T_1T_2 = T_{1n}T_{2n}(I + \Delta_1 + \Delta_2 + \Delta_1\Delta_2) \tag{3.16}$$

By expressing Δ_1 in reference frame of T_2 using Equation. 3.6 the following relation is obtained:

$$\Delta_1 = T_{2n}^{-1}\Delta_1T_{2n} \tag{3.17}$$

Using Equation. 3.17 and substituting it in Equation. 3.16 while ignoring higher order error terms (first order approximation):

$$T = T_1T_2 = T_{1n}T_{2n}(I + T_{2n}^{-1}\Delta_1T_{2n} + \Delta_2) \tag{3.18}$$

From the above equation, it can be inferred that the differential matrix for the combined transform is: $T_{2n}^{-1}\Delta_1T_{2n} + \Delta_2$. One of the insights from this is that the *error of the combined transformation is the addition of individual errors after converting them to the same frame of reference*. This is seen here as the error in frame one (Δ_1) is converted to an error in the second frame $(T_{2n}^{-1}\Delta_1T_{2n})$ and added to the error in second transformation (Δ_2).

From the equations 3.9, 3.10 and 3.12 the covariance of T, i.e Σ, in equation 3.18 can be expressed using the following relationship:

$$\Sigma = J_2\Sigma_1J_2^T + \Sigma_2 \tag{3.19}$$

Using the equation 3.19, the *compound* function for error propagation mentioned in the previous section can be described as:

$$(T_3, \Sigma_3) = compound((T_1, \Sigma_1), (T_2, \Sigma_2)) \tag{3.20}$$

where:

$$T_3 = T_1 \cdot T_2 \tag{3.21}$$

$$\Sigma_3 = J_2\Sigma_1J_2^T + \Sigma_2 \tag{3.22}$$

3.3.1. Validation using Monte-Carlo Analysis

To validate the method given in equation 3.20, a Monte-Carlo analysis was carried out on a planar case. This is similar to the validation method used by Barfoot and Furgale in [37]. Su and Lee do not perform any validation of their proposed work in [34]. Since their method is adopted in this chapter, a validation of the work is performed using Monte-Carlo analysis.

For the Monte-Carlo analysis, a synthetic transformation is constructed which transforms a point in x direction by $1\,m$. This transformation is carried out 100 times to ideally move $100m$ in x direction. But each transformation has a small uncertainty associated with it in the rotation about the z axis. This creates errors in the $x-y$ plane on which the transformations are carried out. All of the 100 transformations are combined to form a single run. 10,000 such runs are carried out for the Monte-Carlo analysis. The final position after each run is recorded and forms a part of the final distribution. The properties of the final distribution obtained by the simulation is compared with the properties predicted by the error propagation method.

The transformation and covariance matrices used in the simulation are:

$$T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

$$\Sigma = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1^\circ \end{bmatrix} \Rightarrow \sigma_{Rz}^2 = 1^\circ \tag{3.24}$$

After carrying out the simulation $10,000$ times, during which each of the 100 transformations has an error in rotation about z axis with standard deviation of 1° the results were plotted and can be seen in Fig. 3.3. In Fig. 3.3, $1,000$ runs chosen at random from the simulation are displayed.

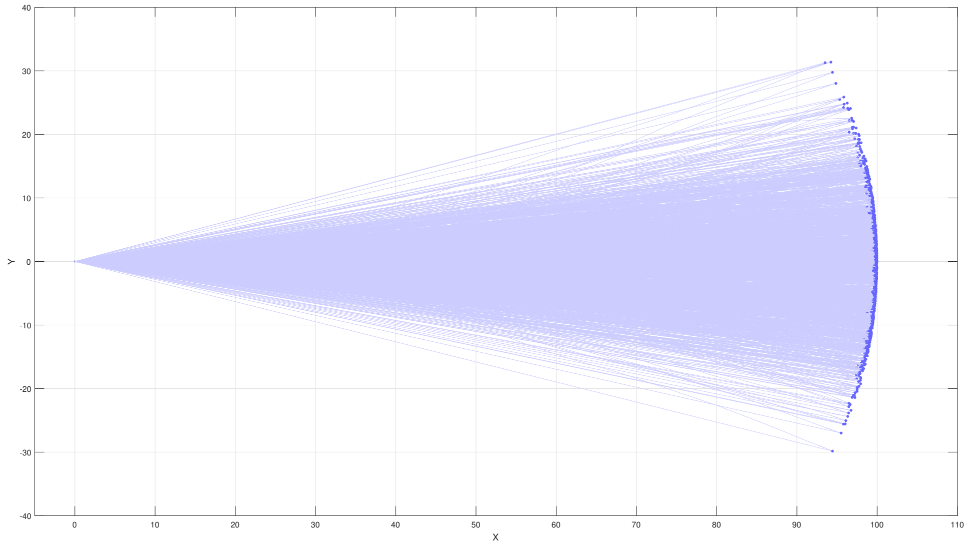


Figure 3.3: Monte-Carlo Analysis

The same propagation of errors in the 100 transformations was performed using the error propagation method proposed in equation 3.20. Using this method, the following covariance matrix for the final pose was obtained:

$$\Sigma_{final} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 103.0673 & 0 & 0 & 0 & 1.5383 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5383 & 0 & 0 & 0 & 0.0308 \end{bmatrix} \tag{3.25}$$

From the covariance matrix given in equation 3.25, it can be seen that the variance in x direction is given at $(1,1)$ as 0 and the variance in y direction is given at $(2,2)$ as 103.0673. However, from the simulation, as shown in Fig. 3.3, it can be seen the variance in the final position exists in both x and y directions. This shows one of the limitations of the first order Taylor series approximation used in section 3.3. By neglecting higher order error terms, the variance information in x direction in this example was lost. Similar results were obtained by Barfoot and Furgale in [37].
The final spread of the pose (in Fig. 3.3) can be described as *banana-shaped*. Long et al. in [38] show that this banana distribution is actually a Gaussian distribution expressed in exponential coordinates. Since this distribution is Gaussian, comparison can be made between the results obtained using Monte-Carlo analysis and the theoretical error propagation method given in this chapter. The distribution of the Y coordinate is compared from both simulations and the results are shown in Fig. 3.4.

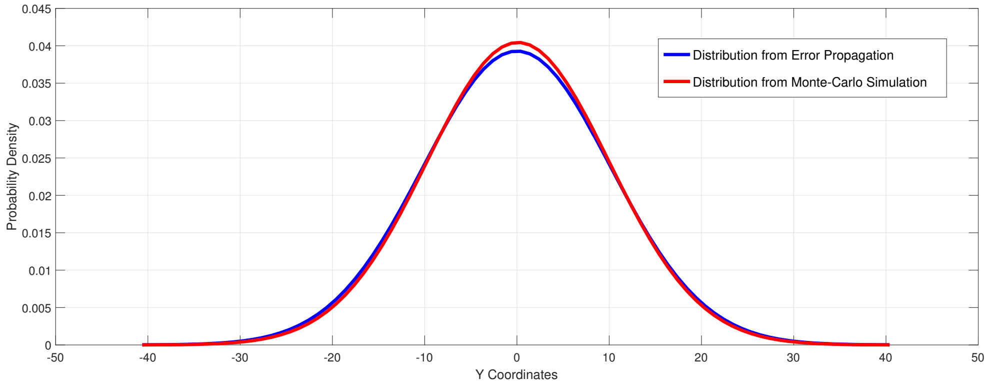


Figure 3.4: Comparison of Theoretical and Monte-Carlo Simulation Results

From Fig. 3.4, it can be seen that the theoretical method of error propagation gives the same result as the Monte-Carlo analysis in this simulation. This validates the method being used for error propagation in this thesis. The comparison from the simulations can be summarized in the Table. 3.1.

Method	μ_x	σ_x	μ_y	σ_y
Monte-Carlo	99.25	0.8642	0.1128	9.8614
Theoretical	100	0	0	10.1522

Table 3.1: Comparison of Monte-Carlo and Theoretical Simulations

3.4. Inverse Error Propagation

While forward error propagation helps in finding the spatial uncertainty of the detected objects, inverting this process can give insights to the specifications of the components required to achieve a certain amount of final accuracy. A method to invert the error propagation method (given by Su and Lee in [34]) is proposed in this thesis. This method is helpful during the design of a vision-aided teleoperation system as it helps in obtaining the knowledge about the specifications required by individual components to achieve a final accuracy goal.
To demonstrate this, a vision-aided teleoperation setup as discussed previously in the thesis is considered. It is assumed that in the design process, one of the parts of the setup (Camera/Robot arm/etc)

has not yet been decided and needs to obtained based on the final accuracy requirements. The final accuracy of this setup can be set to an arbitrary value which will be called (T_f, Σ_f). Using this, the inverse error propagation gives the specifications of the unknown part so that a suitable vendor can be found matching those specifications.
The missing component is chosen to be at a random location in the chain of transformations between the sensor and the gripper. The transformations and their covariances before and after the unknown part and combined using the method given in 3.3 and result in (T_m, Σ_m) and (T_n, Σ_n) respectively. The final transformation and covariance are given by (T_f, Σ_f). The unknown component performs the following transformation along with its covariance:(T, Σ). This can be seen in Figure. 3.5.

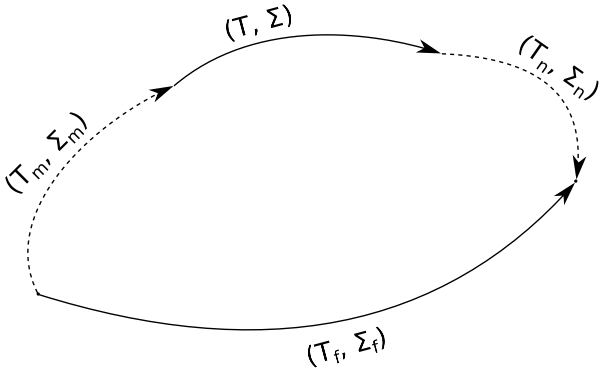


Figure 3.5: Transformation Chain with Unknown Transform (T, Σ)

(T_m, Σ_m) - Transformations and their Covariance before the unknown Component Covariance
(T, Σ) - Unknown Covariance
(T_n, Σ_n) - Transformations and their Covariance after the unknown Component Covariance
(T_f, Σ_f) - Final Transformation and its Covariance

To find a relationship that estimates the requirements for the unknown covariance from the final accuracy requirements, first the transformations before the unknown transformation and the unknown transformation are combined using the compound relationship given in section 3.3 equation 3.20. These give the transformation (T_1, Σ_1) as shown:

$$(T_1, \Sigma_1) = compound((T_m, \Sigma_m), (T, \Sigma)) \tag{3.26}$$

$$T_1 = T_m \cdot T \tag{3.27}$$

$$\Sigma_1 = J_T \Sigma_m J_T^T + \Sigma \tag{3.28}$$

The transformation (T_1, Σ_1) is combined with the transformations after the unknown transformation to give the final transformation and covariance as shown below:

$$(T_f, \Sigma_f) = compound((T_1, \Sigma_1), (T_n, \Sigma_n)) \tag{3.29}$$

$$T_f = T_1 \cdot T_n = T_m \cdot T \cdot T_n \tag{3.30}$$

$$\Sigma_f = J_{T_n} \Sigma_1 J_{T_n}^T + \Sigma_n \tag{3.31}$$

From equation 3.31, the terms can be rearranged to find the unknown covariance Σ:

$$J_{T_n} \Sigma_1 J_{T_n}^T = \Sigma_f - \Sigma_n = \Sigma_{fn} \tag{3.32}$$

$$\Sigma_1 = J_{T_n}^T (\Sigma_f - \Sigma_n) J_{T_n} \tag{3.33}$$

Substituting the value of Σ_1 from equation 3.28, the following is obtained:

$$J_T \Sigma_m J_T^T + \Sigma = J_{T_n}^T (\Sigma_f - \Sigma_n) J_{T_n} \tag{3.34}$$

$$\Sigma = J_{T_n}^T (\Sigma_f - \Sigma_n) J_{T_n} - J_T \Sigma_m J_T^T \tag{3.35}$$

From Equation 3.35, the unknown covariance Σ can be found. To obtain the final accuracy requirement of Σ_f, the component with the unknown error should not have its uncertainty matrix larger than Σ.

Inferences

From equation 3.35, it can be seen that the unknown covariance depends on the jacobian of the transform (J_T) in which the uncertainty is present. This shows that in order to have the accuracy requirement of a component in the system, the transformation being performed by that component should be known. And since the component is not static in some of the cases, the work-space of the component needs to be known before the specification of its uncertainty can be calculated.
Thus, to calculate the accuracy specification of the individual component, the knowledge about its work-space is required. The accuracy specification can be calculated at various points covering the entire work-space and highest accuracy required to fulfill the final uncertainty requirement can be selected. This would ensure that the final accuracy is always better than or equal to the requirements of the system.

3.5. Sensitivity Analysis

As the individual error models are available from the earlier chapter and a method for error propagation is given in this chapter, a sensitivity analysis of the system can be carried out. The aim of this sensitivity analysis is to find the effect of increasing or decreasing uncertainties in various components (transformations) on the total system performance of the vision-aided teleoperation system.
Since the real system will be in motion continuously and there are an infinite number of possible points in the workspace for various components, a static position is chosen and the sensitivity analysis is performed using the transformations of this static pose. This static position of arms can be described as the the robot being mounted on a flat surface with a distance of $0.5m$ between their bases. The object to be manipulated is placed in between the 2 bases along their negative x axis. The arms are tilted 45° to the front to face the object. This gives rise to the following transformation chain to find the pose of the object (knob) in gripper frame for further manipulation:

$$^{Grip}T_{Knob} = {}^{Grip}T_{Tool2}\,{}^{Tool2}T_{Base2}\,{}^{Base2}T_{Base1}\,{}^{Base1}T_{Tool1}\,{}^{Tool1}T_{Cam}\,{}^{Cam}T_{Knob} \tag{3.36}$$

The transformations in this chain are given as follows:

The object with respect to the camera has no rotation and is translated $0, -0.3m, 1m$ in x, y, z directions respectively:

$$^{Cam}T_{Knob} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.37}$$

The camera in tool frame of the first robot arm is displaced by $0.05m$ in z direction:

$$^{Tool1}T_{Cam} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.38}$$

The tool frame of the first robot arm is rotated $225°$ about the Y axis of the base and translated $-0.7m, 0.1m, 0.6m$ in x, y, z directions respectively from the base:

$$^{Base1}T_{Tool1} = \begin{bmatrix} -0.7071 & 0 & 0.7071 & -0.7 \\ 0 & 1 & 0 & 0.1 \\ -0.7071 & 0 & -0.7071 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.39}$$

The base frame of second robot arm is obtained by translating $0.5m$ about the y direction of the base frame of the first robot:

$$^{Base2}T_{Base1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.40}$$

The next term in the transformation chain is the pose of second robot base in the tool frame of the second robot. This obtained by inverting the tool in base frame which is given by translating $-0.7m, -0.1m, 0.6m$ in x, y, z directions respectively and rotating $225°$ about Y axis:

$$^{Tool2}T_{Base2} = (^{Base2}T_{Tool2})^{-1} = \begin{bmatrix} -0.7071 & 0 & 0.7071 & -0.7 \\ 0 & 1 & 0 & -0.1 \\ -0.7071 & 0 & -0.7071 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} -0.7071 & 0 & -0.7071 & -0.07 \\ 0 & 1 & 0 & -0.1 \\ 0.7071 & 0 & -0.7071 & 0.9192 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.41}$$

The final transformation is the tool of the second robot arm in the gripper frame. This is given by translation of $-0.05m$ in the z direction:

$$^{Grip}T_{Tool2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

Each of these transformations have uncertainties associated with them represented using covariance matrices. There are labelled as Σ_1 up to Σ_6 representing uncertainties in transformations from equation 3.37 to 3.42 respectively. To scale the noise in the transformation matrices, the covariance matrices were multiplied with a gain *alpha* (α). The gain alpha was varied from 0.2 to 5 to simulate five times worse or better accuracy in the transformations.

$$\alpha = 0.2, 0.21, 0.22 \ldots. 5 \tag{3.43}$$

$$\Sigma_i = \Sigma_i \cdot \alpha \tag{3.44}$$

The performance metric used in the analysis is the *Frobenius Norm* of the resulting covariance matrix of the final transformation $^{Grip}T_{Knob}$. Three transformations where selected to be studied in this analysis: camera system, hand-eye calibration, and robot kinematics accuracy. These were chosen based on the fact that these system are easily replaceable with better or worse systems giving rise to wide range of variety in these components. This creates the need for insight in subsisting the components with their cheaper (low accuracy) or more expensive (high accuracy) counterparts. Similar case is made for the hand-eye calibration accuracy, since various methods for hand-eye calibration exist, the effect of the accuracy these methods on the final uncertainty necessitates this analysis. The other transformations do not meet these necessities. Hence their results are not discussed in-depth.

For the initial sensitivity analysis, all the covariance matrices were set to identity matrices. This was done to observe the results of the sensitivity analysis if all the transformations had the same level of uncertainty i.e all components of the system are equally faulty.

$$\Sigma_{1,2,\ldots 6} = I_{6\times 6} \tag{3.45}$$

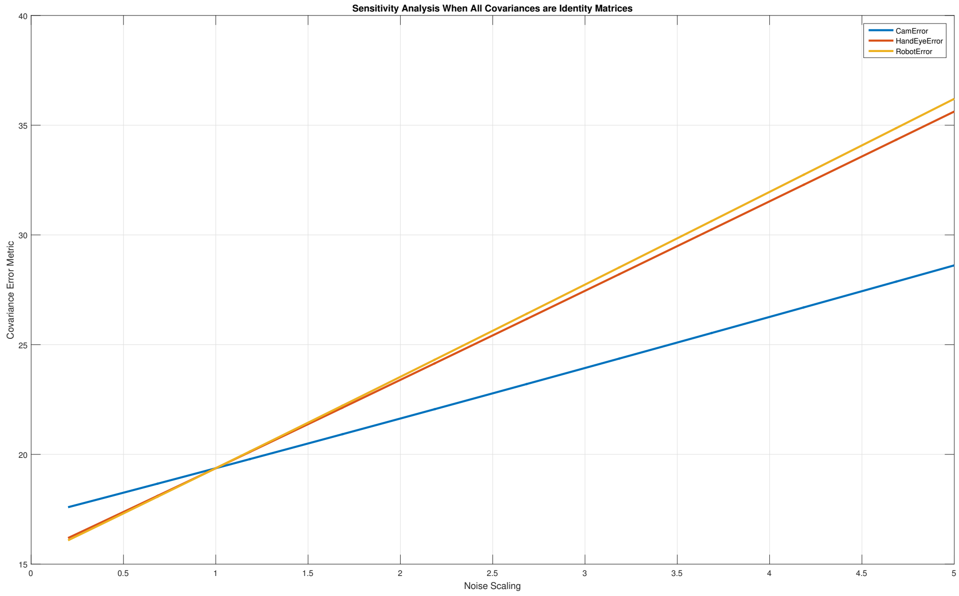


Figure 3.6: Sensitivity Analysis With All Uncertainties as Identity Matrices

The result of this sensitivity analysis can be seen in Fig. 3.6.

From Fig. 3.6 it can be seen that the final uncertainty is most sensitive to uncertainties in robot kinematics in this analysis. Thus, it can be said that in the case that all the parts of the teleoperation system having equal uncertainty, selecting a better robot hand is of higher priority and the camera accuracy can be taken as a low priority to achieve high accuracy in the final result.

A second sensitivity analysis was performed using the error models of the components described in Chapter 2. In contrast to the the previous analysis with all identity matrices, this analysis gives a more realistic picture about how the final uncertainty is affected by change of uncertainties in individual components. The results of this analysis are given in Fig. 3.7.

From Fig. 3.7, it can be seen that in comparison to previous sensitivity analysis in Fig. 3.6 the robot kinematics are no longer the the most sensitive component of the final uncertainty. In fact, they sparsely affect the final uncertainty if they get worse. When using the error models that better represent the reality, it can be seen that the final uncertainty in transformation ${}^{Grip}T_{Knob}$ is most sensitive to the camera accuracy. Hence, it would be recommended from this analysis to upgrade to a better camera for better system performance. This is in contrast to suggesting an upgrade of robot arm based on previous sensitivity analysis.

From both the sensitivity analysis (Fig. 3.6 and 3.7), it can be seen that the final uncertainty is linearly dependent on the noise or uncertainties in individual transformations. This linear relationship stays true irrespective of the transformations involved in the system. This is due the relationship obtained in equation 3.19 which is linear in nature. This can be further traced back to the assumptions in section 3.3 which linearize the error transformation relationship using differential transforms and spatial jacobians. However, the parameters of the linear relationship (slope and y axis intercept) depend on the transformation chain used i.e the system and the workspace. Due to this, this sensitivity analysis is only valid for the given transformations and in their neighbourhood. If the transformations are varied to a large extent, a new sensitivity analysis needs to carried out with those transformations to obtain results relevant in that case.

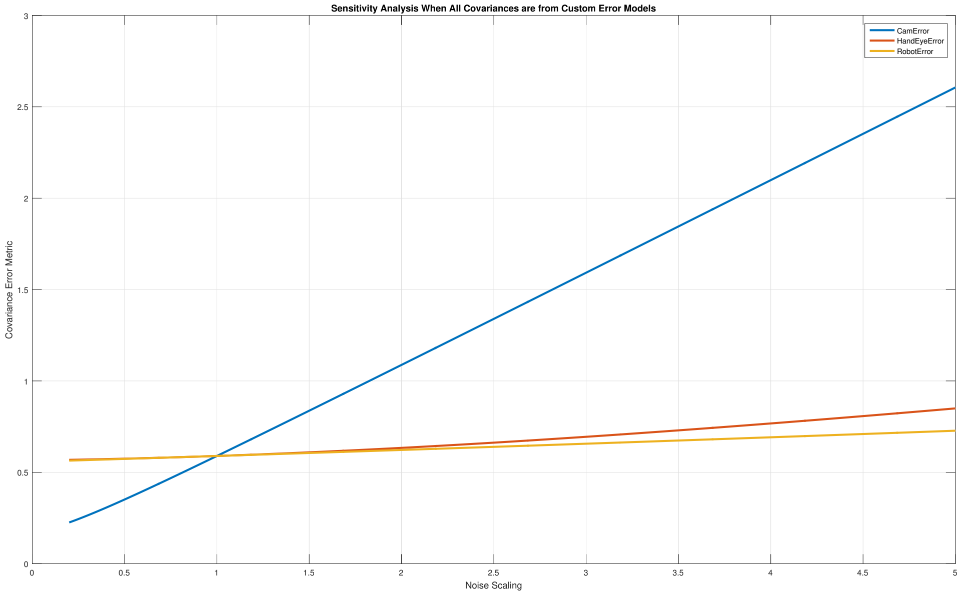


Figure 3.7: Sensitivity Analysis with Modelled Uncertainties

3.6. Conclusion

In this chapter, a method for error and uncertainty propagation through spatial transforms is proposed. This method can be used for online error propagation during vision-aided teleoperation. The uncertainty propagation method uses spatial jacobian and differential transforms to propagate uncertainty using transformation matrices. The given method is validated using Monte-Carlo analysis. The Monte-Carlo analysis also provides some of the limitations of this method along with its validation. Furthermore, a method to invert this process is derived. This can be used to calculate individual component specifications from overall system accuracy requirements given that the knowledge of the workspace is known. Finally a sensitivity analysis is performed on the vision-aided teleoperation system being considered in this thesis. Two cases were considered one of which assumes equal accuracy of all parts and the other uses the uncertainty models from the previous chapter. The sensitivity analysis show a linear relationship between component uncertainty and the final uncertainty. The parameters of this linear relationship depend on the transformations in the system as well as the error models of the components used.

4

Online Uncertainty Estimation

In this chapter, a method to carry out online error and uncertainty estimation during teleoperation is provided and validated experimentally. It utilizes the models and theory developed in the previous chapters and integrates them in an error estimation framework. This framework is then used to determine the uncertainty in errors during teleoperation. Furthermore, various applications for the final uncertainty estimate are proposed.

4.1. Online Error Estimation Framework

The Online Error Estimation Framework proposed in this thesis is shown in Fig. 4.1. The inputs for the framework are the transformations sensed by the robot-vision system. These, along with the error and uncertainty models are used the generate the uncertainty in the pose of the object in the final frame of reference (Eg. Gripper). The final frame of reference is chosen as the one which is most applicable for the user/control system for the given task. This measure of uncertainty can be then used either by the control system to adjust system parameters or shown to the operator to make decisions based on the current system performance.

In Fig. 4.1, an error propagation model block is given which runs continuously in a deterministic real-time manner. The error propagation is said to be deterministic as the number of matrix operations during uncertainty estimation in the system is constant. This makes the process deterministic and hence applicable in a real-time teleoperation system.

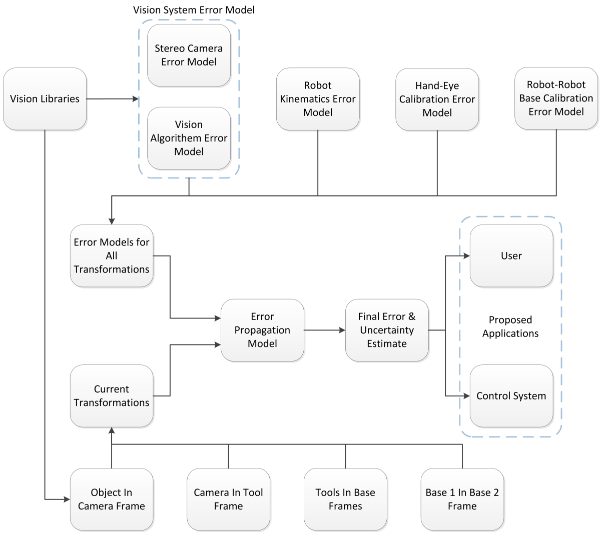

Vision System Error Model
Stereo Camera Error Model
Vision Algorithem Error Model
Vision Libraries
Robot Kinematics Error Model
Hand-Eye Calibration Error Model
Robot-Robot Base Calibration Error Model
Error Models for All Transformations
Current Transformations
Error Propagation Model
Final Error & Uncertainty Estimate
User
Proposed Applications
Control System
Object In Camera Frame
Camera In Tool Frame
Tools In Base Frames
Base 1 In Base 2 Frame

Figure 4.1: Uncertainty Estimation Framework

The error propagation model receives inputs about the current system transformations and the respective uncertainties from the individual sub-system error modules such as Vision System Module, Hand-eye calibration module, Robot module, etc. It then compounds these transformations along with propagating the uncertainties in them to give a final uncertainty estimate. This can be seen in Fig. 4.1 on page 43. As the system is modular, the method of propagation is independent of the error models. It can be updated if better propagation methods are made available. Similarly, the individual error models can be updated if better error models are developed or if different error models are needed due to change in the hardware/software. By separating the functions of the framework into modules, it can be ensured that each module can run at its own execution rate while taking the latest available information from other modules. This allows for the asynchronous execution of the modules. For example, the vision system module senses the pose of the object at a much lower rate when compared to the robot feedback of its end effector pose.

In the current setup, the uncertainty in the vision system is the only dynamic uncertainty. It changes as the camera, which is mounted on the robot arm, moves in the teleoperation workspace. The other uncertainty models such as Hand-eye calibration, robot kinematics, and robot-robot base calibration are constant during teleoperation as shown in Chapter 2. Due to this, they can be hard-coded into the program and can run at the same rate as the error propagation module. In Figure. 4.1, the execution rate was chosen to be $100\,Hz$ as this rate is realized in the current setup. The results are provided to the control system at the same rate at which the propagation model is executed. The rate for the user interface output block can be adjusted to $30/60\,Hz$ to match the display screen refresh rate.

4.2. Experimental Validation using Simulink

The framework shown in Fig. 4.1 was implemented in Mathworks Simulink (shown in Fig. 4.2) as a proof of concept for the proposed idea. It was also used to perform experimental validation of the system. For the validation, it was decided to use only one camera and one robot arm mounted on the stand for logistical reasons. This is considered sufficient for the validation of the error models of stereo camera and the robot arm along with the given propagation method. The results of this validation can be extended to include the full system (explained in Chapter. 1) as the other elements of the system include another robot arm and the robot-robot base calibration which is dependent on the accuracy of the robot arm. Following this, the experiment was carried out in which a stereo camera is mounted on the robot arm. The robot arm was places in multiple configurations while the camera detects an object at various distances. The camera and robot arm make measurements of the object and the robot tool frame respectively. About 1000 (± 10 depending on dropped frames by the camera or communication links) measurements were taken by the camera and the robot arm at each distance the object was placed. The distance between the object and the camera varied from $0.62\,m$ to $1.05\,m$. At each distance, the robot configuration was change to be different from the previous configurations.

The simulink model used for validation can be seen in Fig. 4.2. The uncertainty estimation block (shown in the middle) has three inputs: Object in Camera pose, Robot in tool pose, and hand-eye calibration. It then propagates these along with the uncertainties to output the pose of the object in the base frame along with the covariance.

The uncertainty estimation block (in Fig. 4.2) contains the error models of individual components of the robot system. The inner workings of the block can be seen in Fig. 4.3. As seen in the figure, the *PropagateUncertainty* function takes all the transformations and error models from the system components as inputs and outputs the final transformation along with the Covariance of the pose vector. This is an implementation of Equation. 3.20 in Chapter. 3, restated here:

$$(T_3, \Sigma_3) = compound((T_1, \Sigma_1), (T_2, \Sigma_2)) \tag{4.1}$$

where:

$$T_3 = T_1 \cdot T_2 \tag{4.2}$$

$$\Sigma_3 = J_2 \Sigma_1 J_2^T + \Sigma_2 \tag{4.3}$$

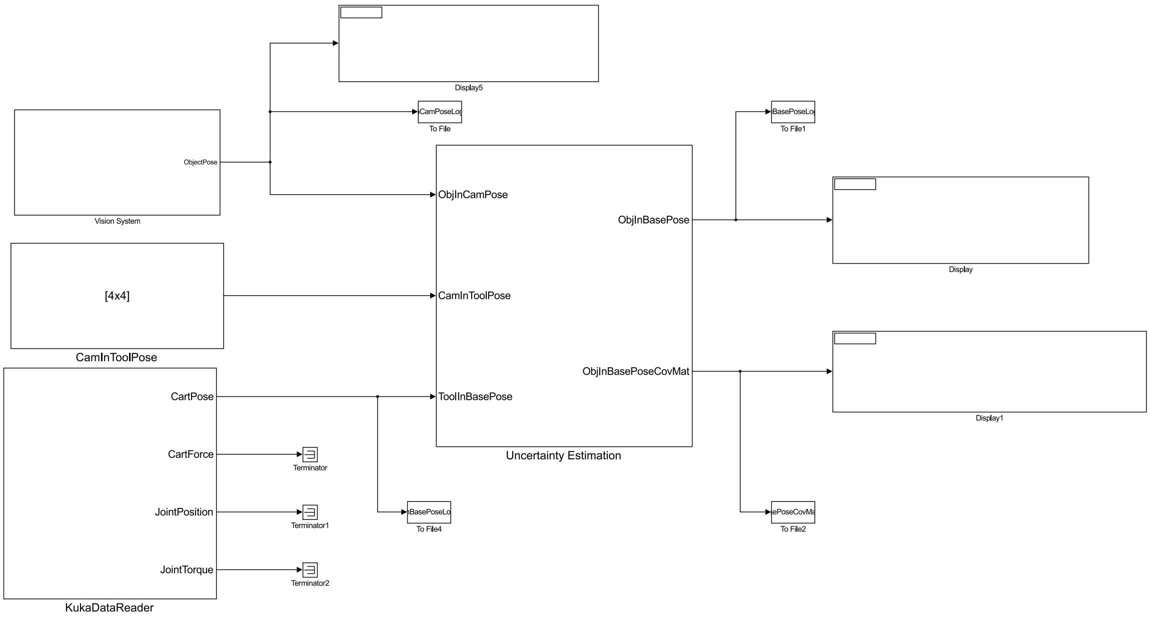


Figure 4.2: Simulink model For Uncertainty Estimation Validation. From Fig. 4.1

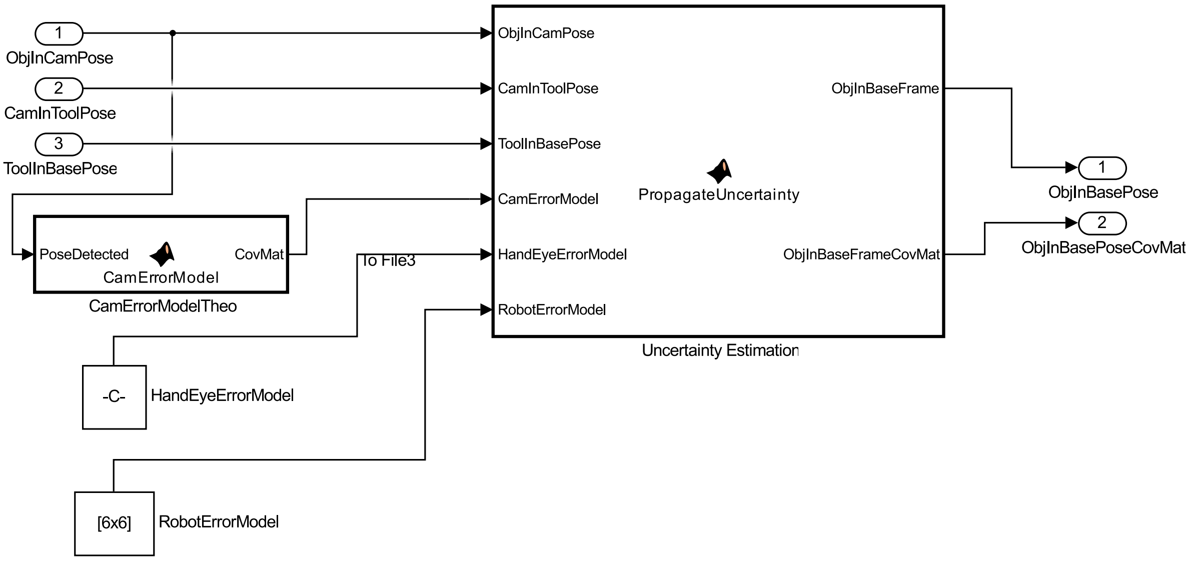


Figure 4.3: Uncertainty Estimation Simulink Function. From 4.2

Another observation that can be made from Fig. 4.3 is that the Hand-eye calibration error model and the robot error model are static/constant during operation whereas the camera error model is a function of the pose of the object detected. The reasoning for this is given in Chapter 2. For the current experiment, the hand-eye transformation between the camera and the robot is hard-coded and hence does not change in-between measurements. Also, no external ground truth was used. Due to this, the uncertainty in the hand-eye transformation was modified to zero. The uncertainty would only be visible if an external ground truth is used. Due to lack of one, this uncertainty cannot be measured and hence was adjusted to zero. This allowed for the validation of the system taking into account the measurements from the vision system and the robot sensors which change with time.

4.2.1. Data Analysis and Results

The first step in data analysis was the verification of the output distribution. It was verified that the resulting distribution of the samples is Gaussian in each axis using the Lilliefor's test [32]. The pose of interest in this experimental validation is the pose of the object being detected in the robot base frame. This pose of determined by 3 transformations: Pose of object in camera frame, hand-eye calibration, and robot tool pose in base frame.

For the validation of the uncertainty estimation framework, the comparison between the uncertainty estimation framework output with one measurement and the empirical distribution is required. This can be used to validate if the uncertainty estimation framework can estimate the empirical distribution and to which extent.

The first measurement was chosen to be the input to the uncertainty estimation framework. The output from the uncertainty estimation framework was compared against the distribution obtained from the following 1000 readings. This was carried out multiple times while varying the distance between the camera and the object and the robot configuration. The results from this comparison done using the experiment data can be seen in Table. 4.1. It can be seen that the worse-case difference between the predicted and the observed uncertainty is $1.39\,mm$ in the Y direction at a distance of $1.05\,m$.

Distance (m)	X Uncertainty (σ_x) (mm)		Y Uncertainty (σ_y) (mm)		Z Uncertainty (σ_z) (mm)	
	Predicted	Experimental	Predicted	Experimental	Predicted	Experimental
0.620	0.4214	0.9864	0.689	0.6261	0.2754	0.2539
0.790	0.5135	0.7837	0.4164	1.0019	0.5546	0.6585
0.940	0.4761	1.2911	0.4820	0.7343	0.6189	0.5940
1.053	0.5040	1.7918	0.5012	1.8618	0.6344	1.0468

Table 4.1: Comparison Between Predicted and Experimental Uncertainties

From the results given in Tab. 4.1, the best and the worst case scenarios can be plotted using the experimental data and the theoretical predictions as shown in the Figures. 4.4 and 4.5 respectively.

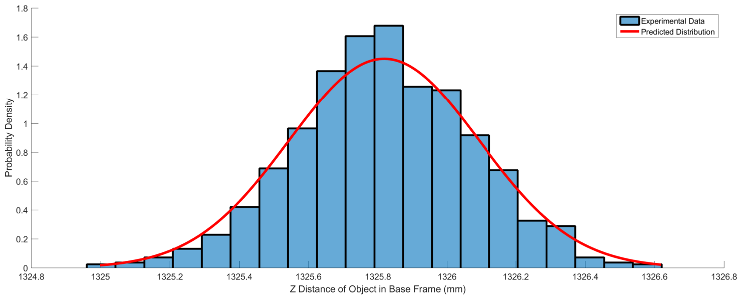


Figure 4.4: Best fit as seen in Z direction at $0.62\,m$ Distance

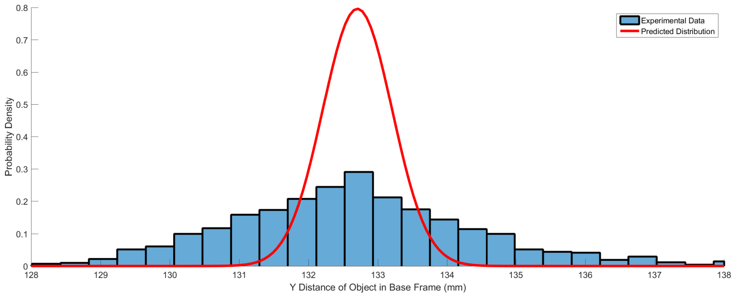


Figure 4.5: Inferior fit as seen in Y direction at $1.05\,m$ Distance

4.2.2. Inferences From Results

From the results shown in Table. 4.1, it can be seen that the uncertainty estimate is accurate up to less than $1.5\,mm$ for distances as far as $1.05\,m$. This translates to an error of less than 0.2% relative to the distance. This shows that the estimate provided by the error estimation framework is of high accuracy even at relatively large operating distances.

However, there are still few discrepancies between the predicted values and the values observed from experiments. The observations do no match exactly to the predictions and in some cases, the observed standard deviations are 3 times higher (difference being $1.36\,mm$) than the predicted uncertainty in the worst case. These disparities are theorized to be caused by the following factors:

- ***Absence of Rotational Uncertainty in Component Models:*** The current uncertainty models of the components in the teleoperation system contain only translational uncertainties and not rotational uncertainties. Due to this, unmodeled rotational uncertainties could propagate though the system and manifest themselves as translational uncertainty due to the nature of the transformations in the system. This could cause high errors in certain axis as seen in the experiment results.
  Better component models could be created which account for the rotational uncertainty in the transformations sensed by the component and hence could improve the final uncertainty estimate.

- ***Neglecting Higher Order Terms in Uncertainty Propagation Method:*** It can be seen in Chapter 3, that during the derivation of the mathematical model used to propagate uncertainty in spatial transformations, higher order terms of Taylor expansion were neglected. This was done to get a closed form analytical solution for the uncertainty propagation.
  As the higher order terms were discarded, this leads to a loss of information contained in those terms. This can also be seen in the Monte-Carlo validation of the method given in the same chapter. Addition of higher order terms would lead to more information being retained during the uncertainty propagation and could give better estimates of the final uncertainty in the system.

- ***Accuracy of Component Models:*** The uncertainty models of the individual components that form the teleoperation system could be the cause for some of the disparity between the predicted and the observed system uncertainty. The models obtained in this thesis are not perfect and the errors in these models could manifest themselves as errors in the final uncertainty estimate. Better estimates can be obtained by developing uncertainty models of higher accuracy.

4.3. Potential Applications

This section explores and suggests potential applications envisioned for the uncertainty estimation framework proposed in this thesis.

During teleoperation, the user has multiple ways of obtaining feedback from the remote environment to assist him/her and perform the task more efficiently. Currently, at the Human Robot Interaction Laboratory (previously known as Telerobotics and Haptics Laboratory) [39, 40], the user is provided with visual feedback using cameras (Mono and Stereo) as well as haptic feedback using a force-reflecting joystick to provide the forces experienced by the robot in the remote environment. Experiments were performed in the past to explore the effects of haptic guidance on task performance [13, 41, 42] and thus methods to carry out haptic shared control using haptic guidance were implemented and added to the existing framework in the lab. Thus, currently the user is able to receive feedback as visual information (Video) and haptic information (Force feedback and Haptic Guidance) with the setup present at the laboratory.

With the addition of uncertainty estimation framework to the existing teleoperation setup, a new source of information is created that can be utilized by the system. This new information can be either used to improve the already available feedback to the user or provide a new type of feedback to the operator. These options are explored in the following sub-sections:

4.3.1. Tuning Gains for Haptic Guidance Using Uncertainty Estimate

This part proposes the use of the uncertainty estimate to fine tune the guidance force gains in real-time during teleoperation to enable better feedback to the user about the system's confidence in the reference used for the haptic guidance.

Introduction to Haptic Guidance

Haptic guidance is a form of haptic feedback that can be provided to the user for assistance during teleoperation tasks. An introduction to haptic guidance (using virtual fixtures) and its applications is given by Abbot et. al in [43]. To summarize haptic guidance, a force is applied on the user using a haptic-feedback input device. This force can be used to either guide the user to help him perform the task (described as Guidance Virtual Force (GVF) in [43]) or help user by providing a repulsive force when the user tries to breach a forbidden region in task space (described as Forbidden Region Virtual Fixtures (FRVF) in [43]).
Either of the GVF or FRVF methods require references in the remote environment in the robot's task space to calculate the correct virtual forces and apply them on the user. Since, not all environment models can always be made available, teleoperation along with haptic guidance using these methods has to carried out in unstructured environments. To acquire an environment model, sensors such as cameras (2D/3D), LIDARs, etc. can be used. Since all physical measurements are approximations of the real world, the models created using the sensors inherently contain errors and uncertainties. An example of haptic guidance provided using both right and wrong references can be seen in Fig. 4.6. The effect of uncertainties in haptic guidance has been discussed previously in Chapter 1, sub-section 1.2.3. From the discussion, the effect of errors in teleoperation with haptic guidance were summarized as:

- Decrease in Task Efficiency
- Exertion of forces on the remote environment by the robot without the knowledge of the operator.

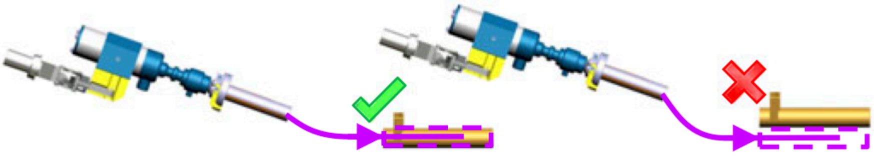


Figure 4.6: Example of Correct and Incorrect Reference for Haptic Guidance from [1].

Proposed Approach: Tuning Guidance Gains Based on Object Uncertainty

To account for the uncertainties in the environment models during teleoperation, it is proposed that the haptic guidance forces are adjusted based on the current system uncertainty. The system uncertainty is computed in real-time during teleoperation. This adjustment is suggested with the aim of reducing the effects of the haptic guidance reference errors on task performance. The uncertainty estimate of the system is expressed as a covariance matrix in the frame of reference that is relevant to the current teleoperation task, e.g. Object in Gripper Frame Covariance. Since the covariance matrix is a measure of the uncertainty in the transformation, the inverse of the covariance matrix is the measure of *precision*, called the *precision matrix* (Σ^{-1}) [44].
The precision matrix can be used as a gain matrix for the guidance forces. It can be used to to scale the forces between a nominal guidance force and no force based on the behaviour of the elements of the matrix. Lower uncertainty in the covariance matrix would lead to higher precision and hence higher guidance forces. Conversely, if the system is notably uncertain about its measurements, it would automatically lower the guidance forces and rely more on the user to make the decisions to execute the task.

An example of this behaviour is visualized in Fig. 4.9 on page 50. The task shown in the figure is a peg-in-hole task. The figure shows the behaviour of the system as distance between the camera and the object reduces, and its effects on the operator and the visual display. Since the distance between the camera and the object reduces during operation, the accuracy/precision of the object (hole on the taskboard) increases. The increase in accuracy causes a the values in the covariance matrix to reduce and conversely increases the values of the precision matrix. As the precision matrix is used as the gain matrix, higher gains are applied for haptic guidance. This causes an increase in the guidance forces being applied on the operator arm as shown in the Figure 4.9.

4.3.2. Presentation of Errors and Uncertainties to the User

From the uncertainty estimation framework proposed in this thesis, new information about the teleoperation system and its performance is made available. This information can be communicated to the operator to make decisions during teleoperation based on system performance. This sub-section proposes a method on how this new information can be communicated to the user during teleoperation.

Visualization of 3-Dimensional 6-DoF Uncertainties

Uncertainties in the teleoperation system are seen to be Gaussian in nature (from Chapter. 2). A covariance matrix gives the covariance of the i^{th} and j^{th} elements of the pose vector at the i, j position in the matrix. Due to 6 degrees of freedom of the robot tool and user input, the uncertainties are represented using a 6×6 covariance matrix. Intuitively, the covariance matrix generalizes the notion of variance to multiple dimensions. To visualize variance in a Gaussian distribution, a single dimensional distribution can be represented by a curve on a plane as seen as the theoretical (predicted) distribution shown in Fig. 4.4 and Fig. 4.5. In case of two dimensions, a 2×2 covariance matrix is generated as an example and plotted in Fig. 4.7

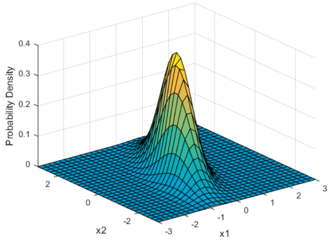


Figure 4.7: Bivariate Normal Distribution. Source: MathWorks

It can be seen from the discussion that the visualization of 1-dimensional distribution requires a 2D plot, and the visualization of a 2-dimensional distribution requires a 3D plot. Following this, the visualization of a pose which is 6 dimensions would require a 7D plot, which is not practical. It is shown in [45] by the authors that for $n-$dimensional probability distributions, the contours of constant probability density are hyper-ellipsoids in $n-$space. From this, it can be seen that the covariance matrix can be represented by an ellipsoid in 6 dimension space. Thus, by taking a contour of constant probability, the visualization of the covariance can be reduced from 7 to 6 dimensional contour. But even this is neither easy to visualize nor interpret. Thus, a method needs to be determined that can be used to show the uncertainties in the pose of the object being tele-manipulated to the user.

Proposed Approach: Overlays on Video Stream

Since the user receives the video from the remote robot during teleoperation, the proposed method uses the video and applies overlays on the video (on top of the object) to represent the uncertainty in the position of the object of interest. Hoff and Vincent in [46] describe a method to visualize uncertainties in 6 degrees of freedom pose as an overlay in augmented reality systems. The 3-dimensional position covariance is shown as a 3-dimensional ellipsoid (of constant probability). This ellipsoid parameters can be changed to represent different confidence/probability regimes such as 1σ or 68.268% Probability, 2σ or 95.45% Probability, etc. Similar methods have been used in spacecraft operations to find the probabilities of collision of spacecraft in intersecting orbits as shown by Woodburn and Tanygin in [47]. An example where constant probability uncertainty ellipsoids are used is in Space Debris collision analysis. A image from spacecraft operation given in [2] is seen in Fig. 4.8. This shows the intersection of probability ellipsoids which help in determining the collision probabilities for the shown objects in orbit.

Such ellipsoid overlays have shown to increase performance other fields as well. An example of the use of these overlays in medicine is given by Simpson et. al. in [48]. The authors show a performance increase during orthopaedic surgical procedures in experiments done in [48] while using ellipsoids as overlays for uncertainty. From the above discussion, it can be seen that ellipsoid overlays

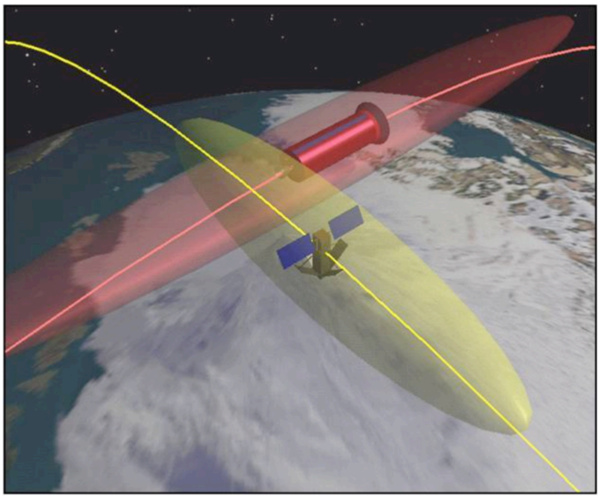


Figure 4.8: Probability Ellipsoids In Spacecraft Operations. From [2].

are a viable method for communicating the uncertainties to the user. Due to this, the visual overlays are proposed as a method for communicating uncertainties in the system to the user. However, the proposed visual overlays are not limited to ellipsoids. In the case when the model of the object in the remote environment is made available (such as a knob), the overlay can also be an enlarged model (enlargement based on the covariance matrix) of the object overlaid on the video. This way, the user can the uncertainty in the position of the object with respect to the size of the object.
An example of the overlays can be used and how they vary during the task can be seen in Fig. 4.9. The overlay's properties such a colour and size depend on the current level of precision computed by the system. This, in turn, is dependent on the distance between the object (hole) and the camera in this example peg-in-hole task.

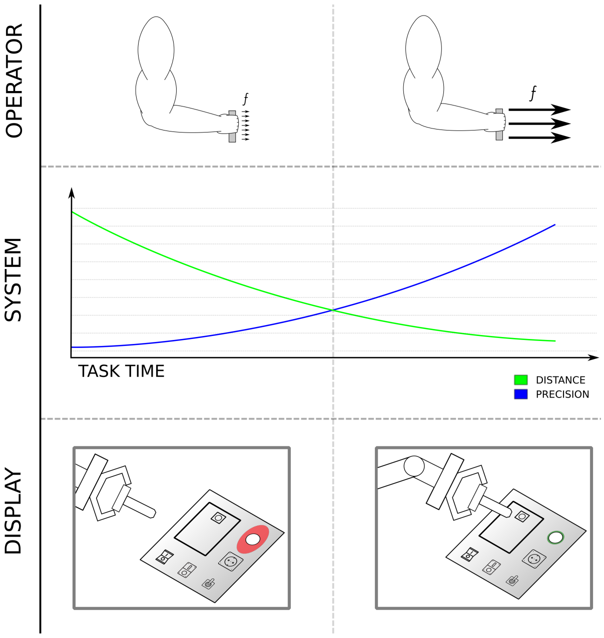


Figure 4.9: Application of Uncertainty Estimation during a Peg-in-Hole Task

4.3.3. Relevant Applications in Other Fields

In other fields such as satellite constellations or swarm robotics, the uncertainty estimation framework can be beneficial if applied. It can help improve the knowledge about the sensed spatial information in these areas.

In both satellite constellations and swarm robotics, information sensed by one of the members of the swarm can be transmitted to the other members of the swarm. If this information is the position of an object relative to the sensor, then this has to be transmitted along with the knowledge of relative positions between the members of the swarm. The relative positions can be estimated by use of systems such as Global Positioning System (GPS) or Inertial Navigation System (INS). Both GPS and INS provide spatial information along with certain uncertainty. The uncertainty estimation framework can be used in this case to combine the uncertainties in the sensed spatial information along with the uncertainties in the relative spatial information while transmitting the location of the sensed object. An example scenario is shown in Fig. 4.10.

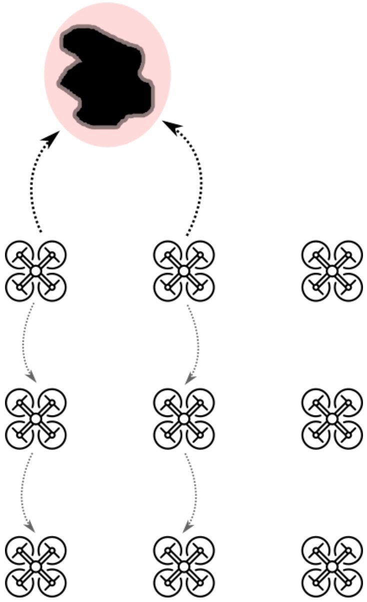


Figure 4.10: Communication of Spatial Information in a Aerial Robot Swarm

In Fig. 4.10, an object, for example, an obstacle, is detected by the two robots on the top left and center positions. These robots sense the obstacle in their respective frames of reference. They then communicate this information along with their positions (determined by GPS) to the robots behind them so that they can avoid them while maintaining the constraints of the swarm constellation. The uncertainty estimation framework can be used to communicate the spatial position of the obstacle along with the uncertainties in the detection as well as the uncertainties in the relative positions of the robot due to GPS. A similar case can be made for swarm satellites which need to observe a spot on the ground while taking into account the sensor and the orbital parameter estimation uncertainties.

4.4. Conclusion

An online uncertainty estimation framework is presented in this chapter and is validated using experiments. The analysis of the results obtained lead to the understanding of the limitations of the current method. These are given along with suggestions for improvement. It is also shown that the error estimation can be performed in a deterministic real-time manner and can output the uncertainty in the transformation that is of interest to the user and the system. Furthermore, multiple uses for this computed uncertainty in the teleoperation system are proposed to assist in decision making for the user and help in improving task performance. Relevant application in other fields are also provided to show the versatility of the proposed framework.

5 Conclusions

In this thesis, the uncertainty of each component of a vision-aided teleoperation system was analyzed and propagated to obtain a final value of the error. Furthermore, a framework was conceived so that the methods given in the thesis can be implemented in the existing teleoperation setup. Each of these steps were validated using experiments. The main conclusions that can be drawn from this research are:

- It is possible to estimate and model the spatial uncertainty in the individual components of the teleoperation system to a relatively high accuracy. Except for the uncertainty in vision components, the rest of the system exhibits static behaviour for varying distances. Covariance matrices are suitable for the representation of the multi-dimensional uncertainties and for the propagation of these uncertainties.

- A method of propagating uncertainties can be derived using the Taylor series approximation which is deterministic and validated using Monte-Carlo analysis. The analysis also shows the limitations of the current method as higher order terms are omitted. Furthermore, this method can be inverted to determine component specifications from given system requirements for a known robot workspace. The sensitivity of the system uncertainty depends not only on the individual uncertainties of the components but also on the spatial transformations involved in the system.

- All of the methods for uncertainty estimation given in this thesis can be incorporated in a teleoperation system. This is showcased using the given uncertainty estimation framework. The full system is validated using experiments and it is shown that the uncertainty estimates are of high accuracy relative to the object distances. Multiple applications for the final uncertainty estimate have been envisioned to increase user and system knowledge during teleoperation.

5.1. Future Work

This thesis demonstrates promising results towards uncertainty estimation in robot teleoperation systems. This does not expend the research area but rather shows that there is much more to be researched and understood. It raises new questions and a clear need and opportunity for future research can be foreseen. These can be listed as follows:

- The current uncertainty models only include the translational uncertainty of the teleoperation system components. These can be made better by determining and including the rotational uncertainties. This could make the final uncertainty estimate more accurate as more information would be taken into account during the computations. Determination of rotational uncertainties will require designing new and innovative experimental methods that can be used to identify these minuscule uncertainties. This can be challenging given the limitations of the current equipment available.

- In this thesis, while deriving the method for uncertainty propagation, first-order Taylor series approximation was used. This implies that the higher order terms were neglected. This caused a loss in information about the system and reduced the accuracy of the uncertainty estimate. To obtain a higher accuracy estimate, an analytical solution using higher order terms is required. One such solution was given by Barfoot and Furgale in [37] using Lie Algebra. One of the next steps after this thesis could be the inclusion of the theory developed by Barfoot and Furgale in the uncertainty estimation framework. This would increase the accuracy of the final uncertainty estimate as the new estimation would include terms up to the fourth order.

- Various applications for the final uncertainty estimate were proposed in this thesis. One of the most interesting future research areas would be to implement these applications and see their effects on operator performance during teleoperation. For example, how much performance improvement can be seen by actively varying haptic guidance forces and letting the user know about the system performance during teleoperation. Furthermore, a more refined Human-Robot interface can be developed to show uncertainty overlays in a way that most benefits the operator performance.

Bibliography

[1] J. van Oosterhout, J. G. Wildenbeest, H. Boessenkool, C. J. Heemskerk, M. R. de Baar, F. C. van der Helm, and D. A. Abbink, "Haptic shared control in tele-manipulation: Effects of inaccuracies in guidance on task execution," *IEEE transactions on haptics*, vol. 8, no. 2, pp. 164–175, 2015.

[2] "Space Debris Basics, The Aerospace Corporation." http://www.aerospace.org/cords/all-about-debris-and-reentry/space-debris-basics/. Accessed:18-11-2017.

[3] A. Schiele, M. Aiple, T. Krueger, F. van der Hulst, S. Kimmer, J. Smisek, and E. den Exter, "Haptics-1: Preliminary results from the first stiffness jnd identification experiment in space," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pp. 13–22, Springer, 2016.

[4] "INTERACT." http://esa-telerobotics.net/meteron/flight-experiments/interact. Accessed:14-11-2016.

[5] "INTERACT - ESA EXPERIMENT RECORD." http://eea.spaceflight.esa.int/portal/exp/?id=9497. Accessed:31-12-2016.

[6] "INTERACT Online Brochure." http://esa-telerobotics.net/uploads/documents/Interact%20Brochure%20-%20Online.pdf. Accessed:01-01-2017.

[7] A. Schiele, T. Krüger, S. Kimmer, M. Aiple, J. Rebelo, J. Smisek, E. den Exter, E. Mattheson, A. Hernandez, and F. van der Hulst, "Haptics-2—a system for bilateral control experiments from space to ground via geosynchronous satellites," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 000892–000897, IEEE, 2016.

[8] Oxford Metrics, "Vicon Motion Capture System," 2005.

[9] J. Maki, J. Bell, K. Herkenhoff, S. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, *et al.*, "Mars exploration rover engineering cameras," *Journal of Geophysical Research: Planets*, vol. 108, no. E12, 2003.

[10] J. Maki, D. Thiessen, A. Pourangi, P. Kobzeff, T. Litwin, L. Scherr, S. Elliott, A. Dingizian, and M. Maimone, "The mars science laboratory engineering cameras," *Space science reviews*, vol. 170, no. 1-4, pp. 77–93, 2012.

[11] A. W. Van der Vaart, *Asymptotic statistics*, vol. 3. Cambridge university press, 1998.

[12] N. Mol, J. Smisek, R. Babuška, and A. Schiele, "Nested compliant admittance control for robotic mechanical assembly of misaligned and tightly toleranced parts," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002717–002722, IEEE, 2016.

[13] N. Mol, *Teleoperation Support System Robust to Uncertainties in Haptic Guidance*. Msc thesis, Delft University of Technology, 2016.

[14] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 1419–1420, Springer, 2001.

[15] P. Marayong, A. Bettini, and A. Okamura, "Effect of virtual fixture compliance on human-machine cooperative manipulation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, pp. 1089–1095, IEEE, 2002.

[16] A. Bettini, S. Lang, A. Okamura, and G. Hager, "Vision assisted control for manipulation using virtual fixtures: Experiments at macro and micro scales," in *2002. Proceedings. ICRA'02. IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3354–3361, IEEE, 2002.

[17] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1139–1144, IEEE, 2005.

[18] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 2–12, 2013.

[19] N. Stefanov, C. Passenberg, A. Peer, and M. Buss, "Design and evaluation of a haptic computer-assistant for telemanipulation tasks," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 4, pp. 385–397, 2013.

[20] P. G. Griffiths and R. B. Gillespie, "Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits," *Human factors*, vol. 47, no. 3, pp. 574–590, 2005.

[21] M. Mulder, D. A. Abbink, and E. R. Boer, "Sharing control with haptics seamless driver support from manual to automatic control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, no. 5, pp. 786–798, 2012.

[22] J. Smisek, M. van Paassen, and A. Schiele, "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy," in *World Haptics Conference (WHC), 2015 IEEE*, pp. 500–505, IEEE, 2015.

[23] "Ensenso N35 Stereo Camera Specifications." https://en.ids-imaging.com/ensensofinder.html?id=N35-602-16-IR. Accessed:15-06-2017.

[24] BIPM, IEC and IFCC, ISO and IUPAC, IUPAP, *Guide to the Expression of Uncertainty in Measurement*. International Organization for Standardization, 1995.

[25] Joint Committee for Guides in Metrology, "Jcgm 100: Evaluation of measurement data - guide to the expression of uncertainty in measurement," tech. rep., JCGM, 2008.

[26] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[27] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *13th International Conference on Pattern Recognition*, pp. 318–322 vol.1, 1996.

[28] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[29] R. Di Marco, S. Rossi, F. Patanè, and P. Cappa, "Technical quality assessment of an optoelectronic system for movement analysis," *Journal of Physics: Conference Series*, vol. 588, p. 012030, 2015.

[30] FLIR, "FLIR i3 Infrared Camera." http://www.flir.com.hk/instruments/display/?id=65813. Accessed:15-06-2017.

[31] "Weisstein, Eric W. "Octant." From MathWorld–A Wolfram Web Resource.." http://mathworld.wolfram.com/Octant.html. Accessed:15-06-2017.

[32] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[33] Oxford Metrics, "Vicon Tracker Software." https://www.vicon.com/products/software/tracker. Accessed:15-06-2017.

[34] S. Shun-Feng and C. S. G. Lee, "Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1376–1389, 1992.

[35] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, 1981.

[36] K. O. Arras, "An introduction to error propagation: Derivation, meaning and examples," technical report, Swiss Federal Institute of Technology Lausanne (EPFL), September 1998 1998.

[37] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[38] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, "The banana distribution is gaussian: A localization study with exponential coordinates," *Robotics: Science and Systems VIII; MIT Press: Cambridge, MA, USA*, pp. 265–272, 2013.

[39] "ESA Telerobotics and Haptics Lab." http://www.esa.int/Our_Activities/Space_Engineering_Technology/Telerobotics_Haptics_Laboratory. Accessed:08-12-2016.

[40] "ESA Telerobotics and Haptics Lab." http://esa-telerobotics.net/. Accessed:08-12-2016.

[41] S. Kimmer, J. Smisek, and A. Schiele, "Effects of haptic guidance and force feedback on mental rotation abilities in a 6-dof teleoperated task," in *2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3092–3097, IEEE, 2015.

[42] J. Smisek, *Systematic Framework for Teleoperation with Haptic Shared Control*. Thesis, Delft University of Technology, 2017.

[43] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," *Robotics research*, pp. 49–64, 2007.

[44] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[45] R. H. Gersten and I. A. Gura, "On analysis of n-dimensional normal probabilities," report, DTIC Document, 1970.

[46] W. Hoff and T. Vincent, "Analysis of head pose accuracy in augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 4, pp. 319–334, 2000.

[47] J. Woodburn and S. Tanygin, "Position covariance visualization," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2002.

[48] A. L. Simpson, B. Ma, E. M. Vasarhelyi, D. P. Borschneck, R. E. Ellis, and A. James Stewart, "Computation and visualization of uncertainty in surgical navigation," *Int J Med Robot*, vol. 10, no. 3, pp. 332–43, 2014.