

**ELECTROMAGNETIC TRACKER CHARACTERIZATION  
AND OPTIMAL TOOL DESIGN  
(WITH APPLICATIONS TO ENT SURGERY)**

by

Gregory Scott Fischer

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree Master of Science in Mechanical Engineering.

Baltimore, Maryland

April, 2005

© Gregory Fischer 2005  
All rights reserved

## **Abstract**

Electromagnetic tracking systems prove to have great potential for serving as the tracking component of image guided surgery (IGS) systems. However, despite their major advantage over other trackers in that they do not require line-of-sight to the sensors, their use has been limited primarily due to their inherent measurement distortion problem. Presented here are methods of mapping the measurement field distortion and results describing the distortion present in various environments. Further, a framework for calibration and characterization of the tracking system's systematic error is presented. The error maps are used to generate polynomial models of the distortion that can be used to dynamically compensate for measurement errors. The other core theme of this work is related to optimal design of electromagnetically tracked tools; presented here are mathematical tools for analytically predicting error propagation and optimally configuring sensors on a tool. A software simulator, using a model of the magnetic field distortion, is used to further design and test these tools in a simulation of actual measurement environments before ever even being built. These tools are used to design and test a set of electromagnetically tracked instruments, specifically for ENT surgical applications.

**ELECTROMAGNETIC TRACKER CHARACTERIZATION  
AND OPTIMAL TOOL DESIGN  
(WITH APPLICATIONS TO ENT SURGERY)**

Reader: \_\_\_\_\_

Kevin Cleary, Ph.D.  
Imaging Science and Information Systems (ISIS) Center  
Department of Radiology  
Georgetown University

Advisor: \_\_\_\_\_

Russell H. Taylor, Ph.D.  
Computer Integrated Surgical Systems and Technology ERC  
Department of Computer Science (and Mechanical Engineering)  
Johns Hopkins University

## Acknowledgments

I would like to thank everyone that has contributed to this work in one way or another. First off, I am very fortunate to have been a member of the NSF Engineering Research Center for Computer Integrated Surgery (ERC CISST: EEC-9731478). The wealth of resources and knowledge available here has been an incredible boost for this project and my educational experience in general. From the ERC, I would like to thank everyone that has helped me out. In particular, I would like to my advisor, Russell Taylor, for his invaluable guidance and support. From the ERC, I would also like to thank the following students who were directly involved in this work: Gouthami Chintalapani, Xiaofeng Liu, Anand Viswanathan, and Xiaohui Wu. Further I would like to thank Dr. Masaru Ishii from the Johns Hopkins Hospital for his guidance with the design of ENT tools and for his assistance with performing experiments in the operating room.

This work was sponsored by Northern Digital, Inc. (Waterloo, Ontario) and I would like to extend my sincere gratitude to them for their support of my work on this project. In addition to the financial support, I would also like to acknowledge the staff there for their guidance and assistance with the work. In particular, I feel lucky to have had the opportunity of working closely with: Paul MacDonald, Saibal Chakraborty, John Niemenan, Jeff Stanley, Stefan Kirsch, and Christian Schilling.

Of course, I would also like to thank all of my family and friends for supporting me on this. I would really like to thank my parents who have been very supportive of me in all of my endeavors and in particular my education. And last, but far from least, I would like to thank Laurie Dickstein who has been extremely supportive and put up with all of my late nights and long hours when I couldn't be home with her because of my involvement with this project. I hope this proves to be a useful reference for anyone interested in working in this field.

# Table Of Contents

<b>ABSTRACT.....</b>	<b>II</b>
<b>ACKNOWLEDGMENTS.....</b>	<b>IV</b>
<b>CHAPTER 1: OVERVIEW.....</b>	<b>1</b>
BACKGROUND ON IMAGE GUIDED SURGERY .....	1
BACKGROUND ON TRACKING SYSTEMS .....	2
<i>Tracker Types.....</i>	2
<i>Northern Digital Tracking Systems .....</i>	7
ELECTROMAGNETIC TRACKING SYSTEMS .....	9
LITERATURE REVIEW .....	13
<i>Overview .....</i>	13
<i>EMTS Tracker Calibration .....</i>	13
<i>Tracker Tool Design .....</i>	16
<i>Related Analysis Work.....</i>	17
PROJECT OVERVIEW .....	19
<b>CHAPTER 2: MEASUREMENT ERROR ANALYSIS.....</b>	<b>20</b>
OVERVIEW .....	20
TECHNIQUES FOR MAPPING ERROR FIELD .....	21
<i>Dynamic vs. Static Data Acquisition .....</i>	21
<i>Robotic vs. Manual Data Collection.....</i>	21
<i>Tracked (Hybrid) vs. Digitized (Stand-alone fixture) Registration .....</i>	22
TRACKER REGISTRATION.....	23
<i>Overview .....</i>	23
<i>Registration of Aurora to Optotrak.....</i>	24
<i>Registration of Aurora Sensors to Hybrid Rigid Body .....</i>	25
<i>Registration of Stand-Alone Fixture .....</i>	28
FIELD DISTORTION MAPPING RESULTS.....	33
<i>Ideal Environment.....</i>	34
<i>Disturbed Environment.....</i>	40
<i>Robotic Mapping (Lab).....</i>	43
<i>Stand-alone Fixture Mapping (Lab).....</i>	46
<i>Stand-alone Fixture Mapping (Disturbed Environment).....</i>	52
<i>Stand-alone Fixture Mapping (Operating Room).....</i>	53
<i>Stand-alone Fixture Mapping (Simulated OR).....</i>	57
ANALYSIS OF DISTORTION .....	59
<i>Comparisons of Environments.....</i>	59
<i>Curvature of Field Distortion.....</i>	61
CONTRIBUTIONS .....	64
<b>CHAPTER 3: ERROR CHARACTERIZATION AND COMPENSATION .....</b>	<b>65</b>
OVERVIEW .....	65
METHODS FOR CHARACTERIZING MEASUREMENT ERROR.....	66
<i>Definition of Error .....</i>	66
<i>Bernstein Polynomials .....</i>	68

<i>Position Related Error</i> .....	72
<i>Orientation Related Error</i> .....	78
<i>Measurement Compensation</i> .....	82
OPTIMIZATION OF CHARACTERIZATION PARAMETERS.....	83
<i>Optimizing the Polynomial Approximation</i> .....	83
<i>Curvature of the Distortion Field</i> .....	85
<i>Error Analysis of Bernstein Approximation</i> .....	87
CHARACTERIZATION RESULTS.....	93
<i>Ideal Undisturbed Set</i> .....	94
<i>Moderately Disturbed Set</i> .....	97
<i>Robotic Data Collection in Lab</i> .....	102
<i>Stand-alone Fixture in Lab</i> .....	103
<i>Stand-alone Fixture in Operating Room</i> .....	106
CONTRIBUTIONS .....	108
<b>CHAPTER 4: OPTIMAL TOOL DESIGN.....</b>	<b>109</b>
OVERVIEW .....	109
TOOL TRACKING OVERVIEW.....	110
TOOL TRACKING ERROR BOUNDS.....	111
TOOL TRACKING UNCERTAINTY .....	117
<i>Covariance Propagation Overview</i> .....	117
<i>Tool Tracking Covariance Propagation</i> .....	120
FRAME FITTING.....	122
<i>Least Squares Solution</i> .....	125
<i>Validation of 5D to 6D Conversion</i> .....	127
BOUNDS ON FRAME FITTING .....	129
<i>Translation Error</i> .....	129
<i>Rotation Error</i> .....	134
UNCERTAINTY OF FRAME FITTING.....	139
<i>Position Error Covariance of a Rigid Body Fit</i> .....	139
<i>Rotation Error Covariance of a Rigid Body Fit</i> .....	140
<i>Total Covariance for a rigid body fit</i> .....	141
<i>Experimental Validation</i> .....	142
OVERALL ERROR BOUNDS FROM LOCALIZATION ERROR.....	144
THIN-PLATE SPLINE MEASUREMENT COMPENSATION .....	147
<i>Local Distortion Modeling</i> .....	147
<i>Thin-Plate Spline Interpolation</i> .....	149
<i>TPS Compensation Results</i> .....	153
CONTRIBUTIONS .....	156
<b>CHAPTER 5: MEASUREMENT AND SIMULATION SOFTWARE.....</b>	<b>157</b>
OVERVIEW .....	157
AURORA TOOLS SOFTWARE.....	158
<i>Summary</i> .....	158
<i>User Guide</i> .....	158
<i>Function Descriptions</i> .....	159
<i>5D to 6D Frame Generation</i> .....	162

<i>Real-time Measurement Compensation</i> .....	168
<i>Measurement Distortion Simulation</i> .....	171
SIMULATION .....	172
<i>Simulation Algorithm</i> .....	172
<i>Simulation Results</i> .....	173
TOOL DESIGN USING SIMULATOR .....	175
CONTRIBUTIONS .....	176
<b>CHAPTER 6: ENT-SPECIFIC TOOL DESIGN .....</b>	<b>177</b>
OVERVIEW .....	177
LITERATURE AND PRODUCT REVIEW .....	179
ENT SPECIFIC TOOL AND REFERENCE DESIGN .....	181
<i>Overview</i> .....	181
<i>Cranial Reference Body</i> .....	182
<i>Tissue Shaver</i> .....	185
<i>Suction Devices and 5 DoF Pointers</i> .....	188
VERIFICATION OF TOOL DESIGN .....	194
CONTRIBUTIONS .....	197
<b>CHAPTER 7: CONCLUSIONS .....</b>	<b>198</b>
CONTRIBUTIONS .....	198
FUTURE DEVELOPMENT .....	200
<i>Interactive OR Field Characterization</i> .....	200
<i>Updated Field Distortion Model</i> .....	200
<i>Further Tool Design</i> .....	200
<i>Image Guided Surgery Software</i> .....	201
<i>Robot Interaction</i> .....	201
<b>REFERENCES.....</b>	<b>202</b>
BOOKS .....	202
TOOL DESIGN, REGISTRATION, AND UNCERTAINTY RELATED PAPERS.....	203
EM TRACKING RELATED PAPERS .....	205
PATENTS .....	207
OTHER REFERENCES .....	207
<b>APPENDIX.....</b>	<b>208</b>
AURORA TOOLS SOFTWARE – USER GUIDE .....	209
<i>Initialization</i> .....	209
<i>Real Time Tracking</i> .....	210
<i>Configuring Tools</i> .....	212
<i>Data Logging</i> .....	216
<i>Compensation of Measurements</i> .....	217
<i>Simulator</i> .....	218
<b>BIBLIOGRAPHY .....</b>	<b>221</b>
<b>CURRICULUM VITA .....</b>	<b>225</b>

# Chapter 1: Overview

## ***Background on Image Guided Surgery***

Image guided surgery (IGS) is a field that has had significant growth in recent years and is becoming more and more widely accepted by medical professionals. IGS systems allow the surgeon to have more information available at the surgical site while performing a procedure. In general, these systems display 3D patient information and render the surgical instrument in this display with respect to the anatomy and a pre-operative plan. The 3D patient information can be a preoperative scan such as CT or MRI that the patient is registered to during the procedure, or it can be a real-time imaging modality such as ultrasound or fluoroscopy. Such guidance assistance is particularly crucial for minimally invasive surgery (MIS) where the procedure is performed through small openings in the body. MIS techniques provide for reduced patient discomfort, healing time, risk of complications, and overall patient outcomes; however, the sensory information available to the surgeon is greatly limited as compared with the open approach. It is especially crucial to have tracked surgical instruments that are presented in the proper position and orientation in the anatomical image on the display for the surgeon. Procedures where such systems are particularly useful and their use has gained momentum are percutaneous therapy, neurosurgery, and Ear, Nose, Throat (ENT) surgery.

The key to an image guided surgical system is knowing where the patient and the instrument are with respect to each other; this is where tracking systems come into play. Spatial tracking systems are the base technology upon which almost all IGS systems are based, with the exception of ones that track the tools and the patient directly with real-time imaging. Since the effectiveness of an IGS system is directly related to how accurately it can show the target anatomy with respect to the surgical instrument, spatial accuracy of the tracking system is paramount. The work here describes how it is possible to model and account for tracking error to optimize tracking for a particular scenario. Figure 1-1 shows some examples of commercially available IGS systems.



**Figure 1-1**



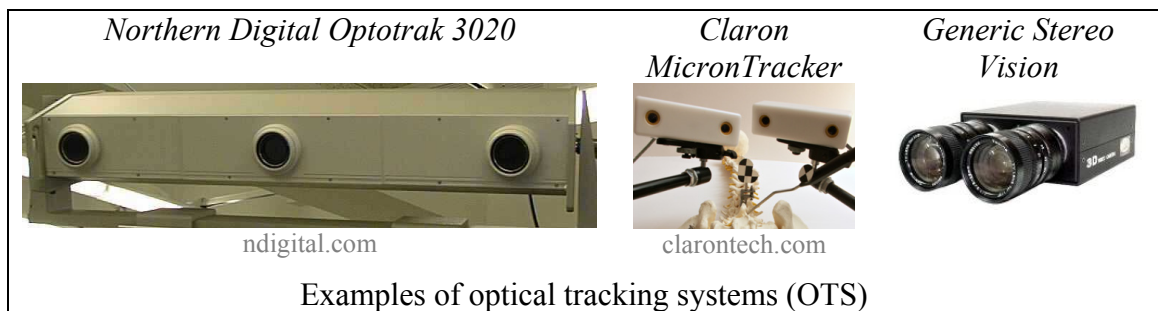
## **Background on Tracking Systems**

### **Tracker Types**

This section details a multitude of available tracking modalities. Obviously this list is not exhaustive, but it gives an overview of many popular types of systems used for medical applications. The first two types mentioned, optical tracking systems (OTS) and electromagnetic tracking systems (EMTS), are the most common for IGS applications.

### **Optical Tracking Systems**

Optical tracking systems are the most prevalent tracking systems used today for image-guided surgery; some example optical tracking solutions are shown in Figure 1-2. In general, these systems provide for a simple tracking solution that has high accuracy, fast sample rate, and relatively constant and isotropic measurement error. These trackers fall into two primary classes: active and passive. An active system works by flashing a set of markers, usually infrared LEDs (IREDs), and either by the pattern or firing sequence the markers are identified and tracked (e.g. NDI Optotrak). Passive systems themselves fall into two categories: 1) ones that use retro-reflective markers that show up clearly as balls in the cameras when infrared (IR) light from an external source is flashed onto them (e.g. NDI Polaris), and 2) ones with a stereo camera pair that picks up a pattern and can track it in space without the assistance of IR markers (e.g. Claron MicronTracker). Custom optical tracking solutions using stereo camera pairs are also potentially useful tools for instrument tracking and guidance.

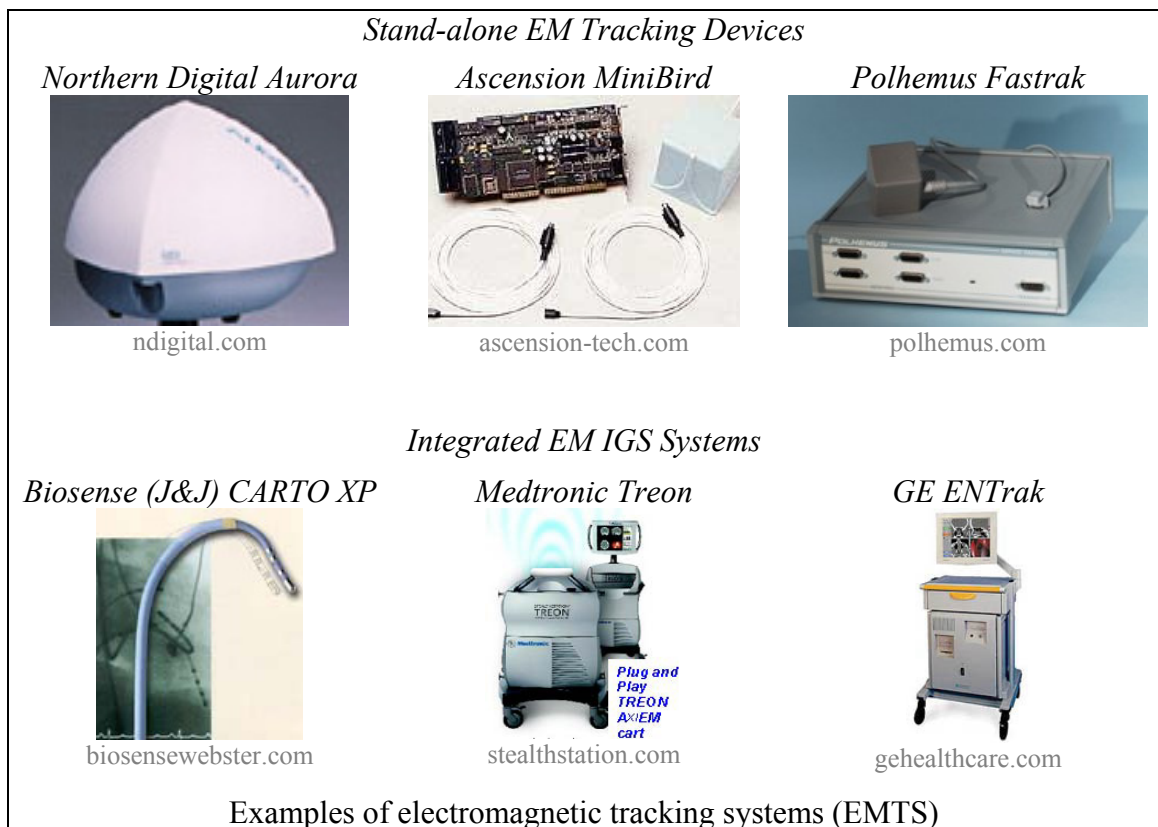


**Figure 1-2**

## Electromagnetic Tracking Systems

Electromagnetic tracking systems (EMTS) are a completely different tracking modality than OTS systems. They have a field generator (FG) producing one or more distinct magnetic fields, and a receiver with one or more sensor coils that picks up the signals. They can determine the position and orientation of the tool with respect to the FG based on the sensor coil's electrical measurements and a physical model of the magnetic field. The obvious advantage of these systems over OTS is the lack of a line-of-sight constraint between the FG and the sensor; this allows tracking of an object inside of the body directly at the distal end, as opposed to tracking markers at a distant handle as is typical when using optical systems. Unfortunately, EMTS have a downside; since the measurement is determined from a physical model of the magnetic field, anything that distorts the generated field (such as metal in an OR) will cause a decrease in accuracy.

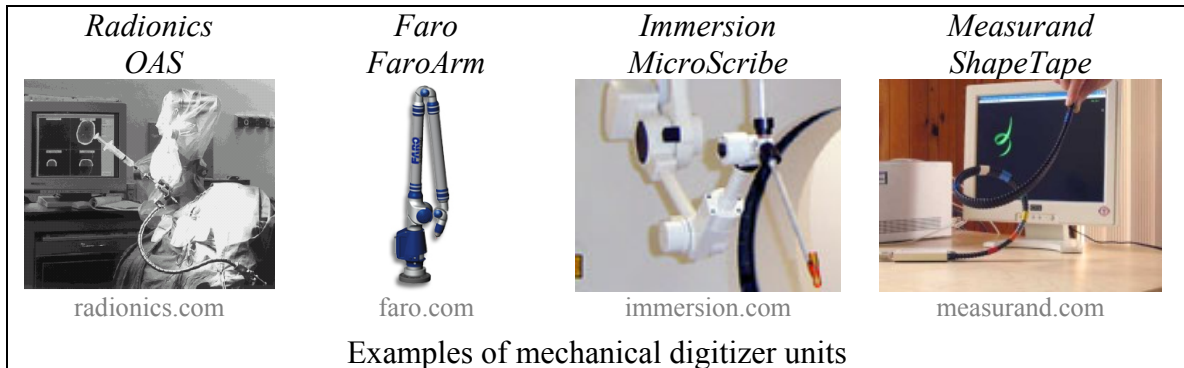
Examples of commercially available stand-alone EMTS are: NDI Aurora, Ascension miniBIRD, and Polhemus Fastrak. Some integrated EMTS solutions offered are: GE InstaTrak (and ENT specific ENTrak) systems, Medtronic AxiEM add-on to the StealthStation, and Johnson and Johnson Biosense-Webster CARTO XP for catheter tracking for cardiology applications. Figure 1-3 displays examples from each of two categories of EM trackers: stand-alone systems and integrated electromagnetically tracked IGS systems.



**Figure 1-3**

## Encoded Mechanical Digitizers

A predecessor to external tracking systems is the mechanical digitizer. Such a device generally consists of a passive arm with encoded joints; using the forward kinematics of the device, it is possible to accurately determine the tip location with respect to a known base frame. The image to the left in Figure 1-4 is of the Radionics Operating Arm System (OAS), and is shown with its IGS software interface in the background. The left-center image shows a modern “portable CMM” manufactured by Faro. This digitizer has extremely high accuracy, quoted at 0.0005”. The image below in the right-center of Figure 1-4 is of a newer style digitizer (Philips / Immersion MicroScribe) that is not passive; the joints are motorized such that it can provide force feedback to the interventionalist. At this point, however, it blurs the line between tracker and robot. Measurand has presented a new digitizer type, whose pose is determined via monitoring optical fibers inside. This ShapeTape is shown on the right.



**Figure 1-4**

## Stereotactic Frames

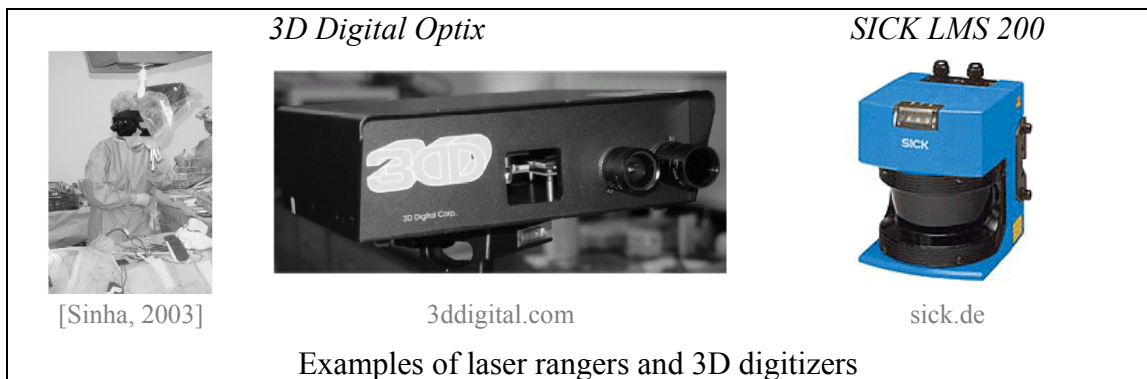
Using a stereotactic frame to help guide an intervention, a surgeon can avoid the use of electronics altogether for tracking in the OR. Such a frame is rigidly mounted to the patient or the OR table and can be aligned such that the guide will direct the tool directly to the target. These devices were precursors to electronic trackers, but can still prove useful for applications not requiring continuous tracking. Examples of such frames are shown in Figure 1-5. Such systems are used primarily for neurosurgery applications.



**Figure 1-5**

## Laser Rangers and Digitizers

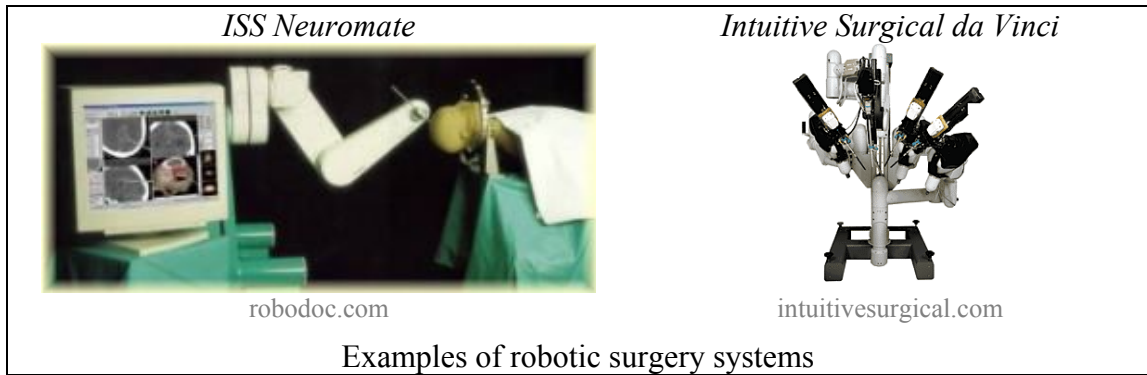
Another method of registration and tracking is accomplished through the use of laser range finders. Not so prevalent in medical applications, the devices emit laser light that is reflected back to the device. Based on time-of-flight information, the shape, position, and velocity of a target or surface can be obtained. Such a device is less useful for tracking tools, but can prove to be very useful for tracking and mapping a surface. In Figure 1-6 are examples of laser rangers. The left photograph displays the 3D Digital system being used for cortical shift tracking at Vanderbilt University [Sinha, 2003]. The right image shows a popular laser ranging system (LRS) for mobile robotics applications. In addition to lasers, ultrasonic and other sources can similarly be used for range finding and tracking.



**Figure 1-6**

## Robotics

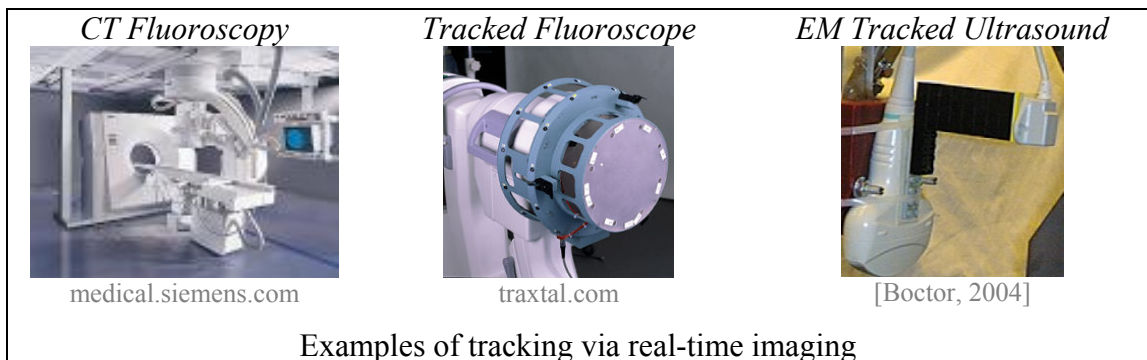
A robot can also prove to be great tool for localization of a tool. Although not a passive tracking system, an accurately calibrated robot allows the user to very accurately place the end effector of the robot in a precise location within the calibrated workspace. Therefore, if the robot is being used to perform/assist with a procedure, then a byproduct of its use is that the tool pose can be relayed to a graphics workstation to render the IGS scene without any additional tracking. Shown on the left in Figure 1-7 is an image of the Integrated Surgical Systems (ISS) Neuromate robot with corresponding IGS software. This system allows for precise manipulation of neurosurgery instruments and replaces the need for any stereotactic frames to be affixed to the patient's skull. A more modern robotic surgery system, Intuitive Surgical's da Vinci robot, is shown on the right; it is more general purpose and is generally used for abdominal and thoracic procedures.



**Figure 1-7**

**Real-time Imaging**

Although not acting as a stand-alone tracking system, a further method of tracking is the use of medical imaging directly. A good example would be the use of fluoroscopy or ultrasound, where a fiducial pattern can be located in the image, and based on that information a registration can be made between the tool and the anatomy. Other modalities that lend themselves to real-time tracking are CT fluoroscopy and MRI. Sometimes, the use of real-time imaging is combined with a stand-alone tracking system. In Figure 1-8 to the left is a combination CT fluoroscopy suite, in the center is an optically tracked fluoroscope head, and to the right is a calibrated, electromagnetically tracked ultrasound probe used for 3D ultrasound imaging and robotic needle guidance from a previous work by Emad Boctor and myself [Boctor, 2004].






**Figure 1-8**

## Northern Digital Tracking Systems

Northern Digital Inc. (Waterloo, Canada) is one of the largest providers of tracking systems for IGS applications. Many commercial systems use these products as their tracking component. In particular, the Polaris has proved to be almost ubiquitous when searching for an IGS solution. The Optotrak provides for significantly higher accuracy and proves to often serve as a “gold standard” for verification of a procedure and for research use. However, in order for it to be more practical in an OR setting, an Optotrak would most likely have to be installed in an OR due to the size and weight constraints. The Aurora is NDI’s contribution to the EM tracking field. Being an EM system, however, it is subject to the electromagnetic field distortion problem mentioned previously. Therefore, the primary focus of this work is to optimize the Aurora tracking system by: modeling the field distortion, compensating for the related tracking error, and ‘optimally’ designing tracked tools for a given environment.

Table 1-1 shows three systems that NDI produces along with their respective relevant system properties for comparison. This is a very good representative group of tracking systems on the market today. The Aurora system is an electromagnetic tracker; these trackers are generally the most convenient to use of the group due to their portability and lack of line-of-site requirements. However, they are unfortunately very prone to measurement errors due to EM interference and magnetic field distortion. Also, the Aurora system has a rather limited range and its accuracy, even in undisturbed environments, is not as high as its optical counterparts. If there were an equivalent of the Aurora in the optical tracking spectrum, it would be the Polaris. This is a very compact, easy to use optical tracking system; it allows for the use of both active markers on the tool that flash IR light themselves, or retro-reflective markers that reflect IR light emitted from the camera unit. IR sensitive cameras receive the light and triangulate the position of the markers, and in turn the associated rigid bodies. This is the most common tracking system today for IGS applications due primarily to its compactness and convenience. The Optotrak system is the “gold standard” of optical tracking; this system far surpasses the other two in terms of accuracy. It has been widely used for research applications, both on phantoms and clinically; however, the lower cost and much smaller bulk of the Polaris have resulted in the latter being adopted by many commercial IGS systems.

	<i>Aurora</i> Electromagnetic	<i>Optotrak Certus</i> Active Optical	<i>Polaris</i> Passive/Active Optical
			
<b>6 DoF Accuracy</b>	0.9mm RMS 0.8° RMS	0.1-0.15mm RMS (Ori. depends on baseline)	0.35mm RMS (Ori. depends on baseline)
<b>Max. Range</b>	0.5M	6M(far) / 3M (near)	2.4 M
<b>Refresh Rate</b>	45Hz (1-5 sensors) 22Hz (6-8 sensors)	4.6 KHz (markers) 1.5 KHz Max. (frames)	60 Hz Max.
<b># Tools</b>	8 (5Dof) 4 (6DoF)	512 Markers (3DoF) 170 RB (6DoF)	3+ Active 9 Passive
<b>Weight</b>	2.2Kg (FG)	18 Kg (Camera)	2Kg (Camera)
Northern Digital tracking systems with comparative specifications <sup>1</sup>			




**Table 1-1**

<sup>1</sup> All specifications are from the “Technical Specification” data sheets provided by the manufacturer. They are available at <http://www.ndigital.com>

## Electromagnetic Tracking Systems

Electromagnetic tracking systems have been available for quite a long time (10+ years). Early on, the focus for such systems was primarily virtual reality and motion capture for the computer graphics community. The recent surge in interest in image guided surgery applications has spawned a renewed interest in these tracking systems. There are several relatively new EMTS on the market today: NDI Aurora, Polhemus Liberty, Ascension microBIRD, GE ENTrak, J&J Biosense-Webster CARTO XP, and Medtronic AxiEM.

Below is a representative set of several popular EM trackers on the market today listed with their respective specifications for comparison. This is by no means an exhaustive list: both Polhemus and Ascension have other EMTS models, and there are also other systems out there such as those from J&J, GE, and Medtronic that are integrated directly in IGS systems.

	<i>Northern Digital Aurora</i> <sup>1</sup>	<i>Polhemus Fastrak</i> <sup>2</sup>	<i>Ascension MiniBIRD</i> <sup>3</sup>
			
<b>Field Type</b>	AC Field	AC Field	Pulsed DC Field
<b>DoF</b>	5DoF / 6DoF	6DoF	6DoF
<b>Accuracy (5DoF)</b>	Pos: 0.7mm RMS Ori: 0.3° RMS	N/A	N/A
<b>Accuracy (6DoF)</b>	Pos: 0.9mm RMS Ori: 0.8° RMS	Pos: 0.76mm RMS Ori: 0.15° RMS	Pos: 1.8mm RMS Ori: 0.5° RMS
<b># Sensors</b>	8 (5DoF) 4 (6DoF)	4 per unit (can daisy chain)	2 per unit (can daisy chain)
<b>Sample Rate</b>	45 Hz (1-5 sensors) 22 Hz (6-8 sensors)	$\frac{120}{\#sensors}$ Hz	120 Hz
<b>Range</b>	500mm	1520mm	+/- 760mm
<b>Sensor Size</b>	0.8∅ x 10mm (5DoF)	23 x 28 x 15mm	10 x 5 x 5mm
Examples of EM trackers with comparative specifications			

**Table 1-2**

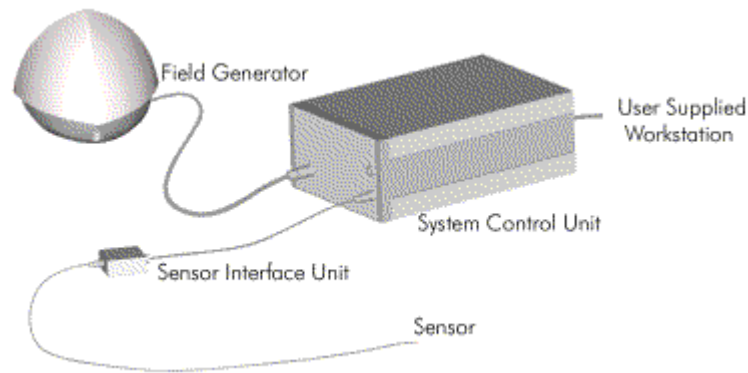
<sup>1</sup> Specifications from Northern Digital datasheet available at <http://www.ndigital.com/aurora.php>

<sup>2</sup> Specifications from Polhemus datasheet available at <http://www.polhemus.com/fastrak.htm>

<sup>3</sup> Specifications from Ascension datasheet available at <http://www.ascension-tech.com/products/microbird.php>



EM trackers, for the most part, all have the same basic components. Figure 1-9 shows the components of an Aurora system from Northern Digital (NDI). The key components of an EMTS are: 1) the field generator (FG) that contains coils of wire in a specified configuration to generate the magnetic fields, 2) the sensor that contains one or more coils of wire that produce electrical signals in the presence of the induced magnetic fields, and 3) the system control unit (SCU) that takes care of controlling the FG, receiving the sensor data and synchronizing it with the FG output, performing calculations, and communicating with the host PC. Some units, such as the Aurora, have a sensor interface unit (SIU) that takes some load off of the SCU and performs amplification and/or digitization of the incoming signal before reaching the main system control unit.



Components of the Aurora EM tracking system  
(Image courtesy of Northern Digital Inc.)

**Figure 1-9**

The key to EM tracker functionality is the way in which it generates a set of magnetic fields from a set of coils in the FG, whose relative poses are known from manufacturing. By generating these fields along different directions, and by synchronizing the output with the received sensor coil electrical response, it is possible to find the relative position and orientation (pose) of the sensor with respect to the FG using the electrical measurements and a physical model. The methods that these fields are generated generally fall into one of two categories: pulsed direct current (DC) fields and alternating current (AC) fields. A pulsed DC field generator creates sets of quasi-static magnetic fields in a given sequence. The sensor picks up responses from each distinct fields, after waiting for any transients to die out, and the magnitudes of the responses are used to calculate the position and orientation of the sensor. An AC FG, on the other hand, outputs a signal pattern on a high frequency carrier wave, and this pattern is repeated for each sample interval; amplitudes of the received signals are used to determine the pose.

Each method (AC vs. DC) has its own benefits and drawbacks; the most significant of which are the way that their signals are affected by magnetic field distortion. A pulsed DC tracking system can be very susceptible to ferrous materials in the field and any external magnetic presence, including that of the Earth's field, if it is not subtracted off. An AC system is not affected by external static (DC) magnetic fields; however, because of the AC nature of the signal, eddy currents are generated in nearby conductive materials. These eddy currents then generate their own magnetic fields that distort the

measurement field. There is debate as to which system is better, and it really comes down to testing them out in a given application to determine which is the right one for the job.

One possible configuration of EM trackers uses three orthogonal field generating coils, and another three orthogonal receiver coils in the sensor. This arrangement allows for 6 DoF measurement, but also rather large sensors in general since three coils must be in the sensor. For clarification, in this 3x3 configuration, that means a total of nine distinct measurements are taken and used for calculation each sample period. This corresponds to all pairs of readings between each sensor coil and each FG coil. An example of this arrangement is the J&J Biosense-Webster tracked catheter, which is illustrated in Figure 1-10. As described in [Ben-Haim, 2004], three orthogonal field generating coils produce AC EM fields at three distinct frequencies. These signals are received by each of three orthogonal sensor coils, amplified, and sent to DSP unit. In the DSP unit, the signal intensities from each FG coil for each sensor coil is obtained. A possible method to do this would be filtering out the desired frequency and determining magnitude; another option would be to perform an FFT and see the magnitude of the signal at the 3 peaks. For the actual system, the signals are separated through cross correlation between the emitter and receiver signals in order to increase update rate.

The images in Figure 1-11 are representative of the internal coil configuration of an EMTS field generator. These particular images are of the Northern Digital Aurora system; the left shows an illustration from an NDI patent [Kirsch, 2003a] describing the system, and the right shows the internal structure of the corresponding physical device. For this system, there are nine generating coils arranged in a tetrahedral configuration; sensors lie along each edge of the tetrahedron and on three of the four faces. With this configuration, it is possible to have only a single sensor coil and still resolve the pose in 5 DoF, thereby making it possible to have very small sensors. There are a total of nine measurements between the FG coils and the sensor coil that are used to determine the 5 DoF measurements (the lost DoF corresponds to rotation about the axis of the sensor coil which can not be resolved). Measurements in 6 DoF can be achieved by combining two or more 5 DoF sensor measurements from sensors that are rigidly coupled to one another.

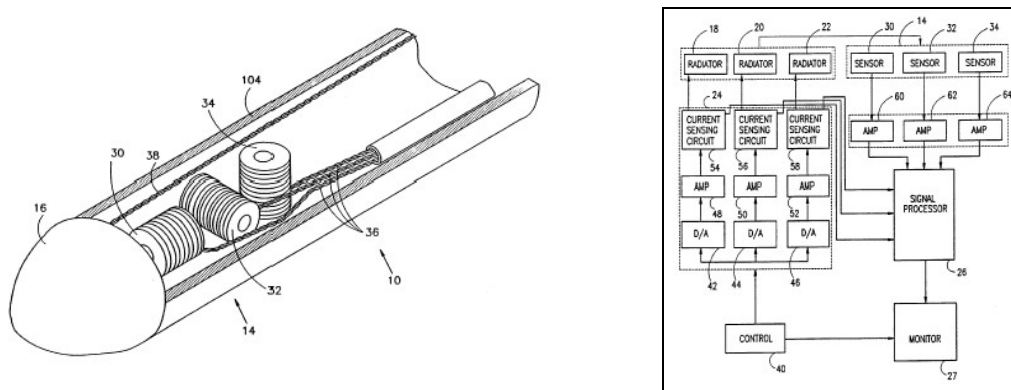


Diagram of sensor coil configuration inside of a Biosense Catheter (left) and a flow chart showing the signal flow (right) [Ben-Haim, 2004]

**Figure 1-10**

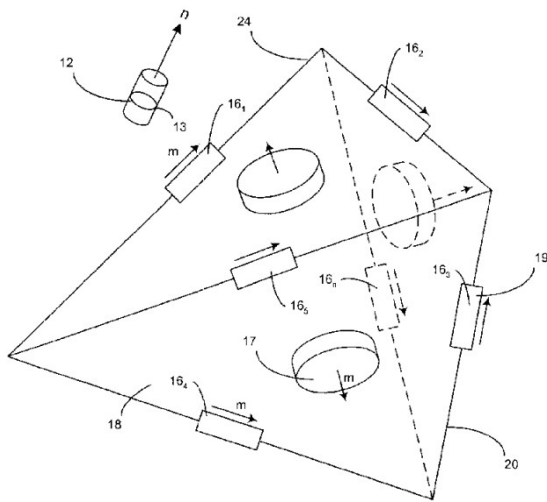


Diagram of FG coil configuration of NDI Aurora from [Kirsch, 2003a]



Photograph of the corresponding coils in the Aurora field generator

**Figure 1-11**

It is clear that there are quite a few possible combinations of ways that EM tracking systems can be constructed. Primarily, the main choices are: the number of emitter coils, the number of receiver coils, the arrangement of both sets of coils, and the field type (AC or DC). The ‘ideal’ choice is really dependent of the tracking scenario; in general, DC fields are better for dealing with conductive, but non-magnetic materials near the working field, and AC fields are better for working in areas where there is a static EM field. As for coil configuration, in general placing three coils on the tool and three in the FG will give better accuracy and measurements can be resolved in 6 DoF, but the trade-off is sensor size and/or complexity.

## **Literature Review**

### **Overview**

The work discussed in this document is rather broad reaching, and its extent spans several fields of study including tracker error characterization, tracker calibration, tool design, registration, and error propagation. There has been much research presented in the literature in each of these fields that relates in one way or another to what is being presented here. This section details much of the related work and explains the relationship to the problem at hand.

### **EMTS Tracker Calibration**

There are several groups that have addressed the need for characterization and compensation of the measurement distortion of EMTS systems. Much of the early work is heavily weighted towards the computer graphics community, in particular for virtual reality (VR) and augmented reality (AR) applications where it is necessary to track head mounted displays (HMDs), motion of a body, or motion of a camera. There are two classes of papers in this field: 1) those that quantify the error of EMTS systems, and 2) those that propose some sort of error compensation technique.

Early work studying the accuracy of EM tracking related to medical devices explores the accuracy of tracking ultrasound (US) probes for 3D US applications using the Ascension Flock of Birds (FOB) system [Leotta, 1995]. The work quantifies and compares the measurement uncertainty for configurations of one and two 6 DoF sensors. A study by [Birckfellner, 1998b] details the effects on EM tracking accuracy of materials and equipment commonly found in OR environment. They study the effects on both the AC field Polhemus system and the pulsed DC field FOB, and do so by measuring distortions in position and orientation of the sensors as disturbances are placed in the field and moved with respect to the sensors. Another similar study describes errors in an OR environment using an Ascension MotionStar system [Poulin, 2002]. Further studies of tracking system accuracy in the OR were performed by [Wagner, 2002]; this paper gives a relatively detailed statistical analysis of several trackers exposed to several OR-related factors while focusing primarily on cranial surgery. Another similar study uses the NDI Aurora system and is presented in [Hummel, 2002]. This paper also focuses on the distortion effects of medical equipment such as US scan heads and fluoroscopes.

Recent work by Traxtal, a manufacturer of tracked instruments, and Georgetown's ISIS Center also uses the NDI Aurora system to study the accuracy of EM tracked needles [Glossop, 2002]. In these trials, the NDI Polaris was used as a reference for determining measurement error, with the primary application being spinal nerve block needle placement. A clinical trial of several tracking systems (included one EM system, the GE InstaTrak) is described in [Cartellier, 2001]. This paper is unique because it deals directly with the scenario related to IGS; they determine the tracking error between the

tool and registered CT images based on patient anatomy. In particular, they study accuracy for ENT applications, which is the target application for my work.

Related to the idea of studying the accuracy and mapping errors of tracking systems, the paper by Frantz, et. al. discusses more specific ways of quantifying error in a tracking system [Frantz, 2004]. Further work by the same group [Frantz, 2003] details several methods for quantifying the accuracy of EMTS in particular. They also include some error field characterization for the NDI Aurora system.

In addition to quantifying error, there are several papers that discuss methods of compensating for magnetic field distortion. Many techniques are proposed for such compensation, and also many different levels of depth with which compensation is addressed. Several simpler methods assume that error is only a function of the position of the sensor in the field; even further, some only deal with compensating for the translational component of the error. It is known that, at least for the EMTS systems with which we are working (NDI Aurora), that there is in fact a very strong relationship between the orientation of the sensor and its error. In fact, this error can be even more pronounced than that related to the sensor's position in the field as will be seen later.

The earliest work studying methods for calibrating EM trackers appears to be [Bryson, 1992]. After performing error analysis, this paper details how to calibrate the system with two methods: 1) by modeling error with a least squares polynomial fit, and 2) by using a weighted look-up table (LUT). The error analysis and compensation is only concerned with 3 DoF position error as a function of 3 DoF sensor position only. As mentioned earlier, much of the earlier work with EMTS calibration is concerned with VR and AR applications. [Livingston, 1997] discusses global polynomial fits vs. local LUT methods, but unfortunately there is little detail about the methods used. One variation in this technique is the use of a Faro digitizing arm (as shown earlier in the Encoded Mechanical Digitizers summary) as opposed to an OTS as the measurement reference. [Zachmann, 1997] uses Hardy's Multi-Quadratic method (HMQ) for an interpolation technique optimized for scattered data. A hybrid calibration using the Ascension FOB EMTS and the Flashpoint OTS, based on LUT algorithms is described by [Birkfellner, 1998a]. In all three of the previous works, the method is not concerned with orientation dependence on the error.

[Kindratenko, 2000b] provides for a very thorough survey of EMTS calibration techniques including tri-linear interpolation, shape functions, polynomial fits, and HMQ. The same author also details a framework for tracker calibration, in particular for rotation error correction, for VR applications [Kindratenko, 2000a].

The first paper to discuss EM tracker calibration for medical applications appears to be [Tian, 2000]. This paper also appears to be one of the first to take the orientation of the sensor into account when predicting the 6 DoF error. The authors use quaternion representations to parameterize the orientation in the polynomial fit, and they predict errors in the form of Euler angles. Methods for LUT interpolation are also discussed. Another paper attempting to present a calibration framework is [Ikits, 2001]. The authors

do a good job of quantifying error and presenting a polynomial fit, but again the orientation of the tool is not taken into account when predicting error. Depending on the tracker used, this may or may not be a valid assumption.

From our group, [Wu, 2003b] discusses a complete framework for calibration of an EMTS using polynomial interpolation of 6 DoF errors in a 6 DoF measurement space. [Wu, 2003a] goes further to discuss novel ways of dealing with estimating the error as a function of orientation by subdividing the orientation space and using spherical interpolation between a set of basis orientations, thus simplifying the dimensionality and size of the problem considerably. [Nakada, 2003] is another work describing calibration through the use of a hybrid EMTS-OTS tracking system. Although the compensation algorithms used are rather simple, the salient feature of this paper is the incorporation of a temporal calibration component for dealing with time skew between measurements of the two systems. This is a critical component for dynamic/continuous data collection. Recent work in [Chung, 2004] does take into account 6 DoF error as a function of both sensor position and sensor orientation. They do so through the use of radial basis splines that are parameterized by the orientation quaternion and the 3D sensor position.

## Tracker Tool Design

Thus far, there appear to be no papers that directly discuss optimal tool design for electromagnetic tracking systems. However, there is similar work performed relating to optimal design of optically tracked tools. In fact, these are the works we hope to complement with the addition of mathematical tools for dealing with modeling EMTS tool tracking errors. There are two key fundamental differences between design of optical and EM tracked tools: 1) the error distribution is generally close to being isotropic for optically tracked sensors, and is generally far from it for EM tracked tools, and 2) OTS provide only the 3 DoF position of each sensor, while EM tools can provide those plus rotational information (an additional 2 DoF in the case of the NDI Aurora).

The related work is presented primarily in a series of papers by Fitzpatrick, West, and Maurer. In general, they provide for ways to predict the accuracy of a tool tip for a given tool design in a given isotropic error field. Target registration error (TRE) is the name given to the error at the tip of a tracked tool; translational and rotational TRE is given as a function of the tool configuration and the fiducial localization error (FLE), which is representative to the tracking error of a point by the OTS. Therefore, this TRE is exactly what must be calculated for the EMTS case. Unfortunately, this is a much more complicated problem, primarily due to the lack of a nicely distributed, isotropic error assumption. Because of this, it is not possible to return a straightforward closed form solution for TRE as in the previous work.

The first work in this series, [Fitzpatrick, 1998], provides for a good mathematical introduction to the point-based registration problem, and derives solutions for the registration error. It discusses TRE error for a tool alone, but then extends the work to determine the relative TRE when a coordinate reference frame (CRF) is introduced. This is the key step because for any IGS scenario because the patient will be attached to a CRF, and it is critical to locate the tool with respect to this CRF. In [Batchelor, 2000], the authors study the anisotropically Weighted Orthogonal Procrustes Problem (WOPP), which is the mathematical term for a point cloud to point cloud registration. [Fitzpatrick, 2001] is a detailed paper that describes the mathematics behind the registration problem and error analysis of this problem with applications towards tool tracking. It concludes with a nice, closed-form solution of TRE as a function of marker distribution and the FLE. Unfortunately, however, it assumes an isotropic error field; this may be a pretty good approximation for OTS, but does not hold so well for EMTS. The most recent paper, [West, 2004], provides for a very good overview of the useful information in the previous papers and specifically describes a design methodology for designing optically tracked tools. It also provides simulations and verification with experimental data collected using the NDI Polaris and the NDI Optotrak OTS. The next section gives an overview of papers that present the mathematics behind the above analysis, including statistics of the registration problem and error propagation techniques.

## Related Analysis Work

Determining the optimal rigid body motion to align two sets of points has been intensely studied by many groups, many for the purpose of image registration; this type of fitting is generally known as the Orthogonal Procrustes Problem. Very early work dealing with this problem was part of the PhD thesis of Peter Schonemann. His paper [Schonemann, 1966] discusses how to obtain a solution in the least squares sense to the orthogonal Procrustes problem. He describes the problem as minimizing the sum of squares error in the equation:  $B = AT + E$ , where  $E$  is the residual error from transforming point set  $A$  into point set  $B$  with the transformation  $T$ . In his later paper [Schonemann, 1970], he generalizes the problem to include scaling such that now the goal is to minimize the sum of squares error in the equation:  $B = cAT + J\gamma + E$  through the choice of a transformation  $T$ , contraction factor  $c$ , and a vector  $\gamma$  to account for the central dilation. With a method of aligning to sets of points, the next step was to determine how accurate of a fit is being made. Therefore, in Sibson's paper [Sibson, 1978], the "Procrustes Statistics" reviews the methods of the procrustes problem and details the development of the objective function that is to be minimized. His following paper [Sibson, 1979] includes perturbation analysis into the solution and details how the effects of error in one configuration will affect the least squares solution of the optimal rigid body motion. He introduces the fact that if the introduced error  $\varepsilon Z$  in the alignment of point sets  $X$  and  $Y$  as in this equation:  $Y = X + \varepsilon Z$  are zero mean, normally distributed random variables, then the residual errors fall into a  $\chi^2$  distribution. This is important for the later works that use this to determine the fiducial registration error (FRE) and target registration error (TRE) of a tracked instrument. This work was taken even further by Langron and Collins [Langron, 1985], who perform Analysis of Variance (ANOVA) on the procrustes statistics. Further work on Procrustes statistics in dealing with the registration of shapes was presented in [Goodall, 1991].

A popular method for performing point cloud to point cloud registration is detailed in [Arun, 1987]. This paper gives a step-by-step algorithm for finding the optimal rigid body motion (translation and rotation) to align two point sets in the least squares sense using singular value decomposition (SVD). An issue with this method is the degenerate case where the determinant of the rotation is  $-1$ , this is the case of a reflection. Therefore, Umeyama details a related method that accounts for and corrects this problem [Umeyama, 1991].

In addition to studying the errors induces in a point cloud to point cloud registration, the propagation of error through serial connections of frame transformations is also critical for determining tool tracking error. Therefore, there are several papers in this field that are very applicable to my work here. An early paper studying the propagation of covariances through frame transformations with applications to mobile robotics is presented in [Smith, 1987]. This is a classic paper that discusses how to find the approximate transformation (AT) of one coordinate frame relative to another and the covariance matrix associated with that transformation. Generation of this covariance calculation comes from two methods: *Compounding* two series transformations, and



*Merging* two parallel transformations. Further detail by Smith and Cheeseman is in [Smith, 1990].

Xavier Pennec from INRIA has done much work in determining covariance propagation through frame transformations and also in 3D registration; the work is presented in [Pennec, 1995]. This document details two critical components of covariance propagation, including determining the covariance of a point cloud to point cloud registration, and further how this covariance propagates through frame transformations. It provides great mathematical detail with very good clarity for how to perform such calculations. Complementing this work and the earlier analysis work is a very recent paper by Leo Dorst. In [Dorst, 2005], the author details explicitly how to get the covariance of a best fit rigid body transformation aligning two 3D point clouds. This work is extremely useful and is directly applicable to my research. By combining the work of Dorst and Pennec, it is clear how we can determine the uncertainty of the tool frames and then determine how this is propagated through frame transformations. This allows us to determine how accurately we can track a surgical instrument with respect to a patient-fixed reference frame, which is exactly the key to the tool tracking scenario.

## ***Project Overview***

It is clear that there is much work that can be done to optimize the use of electromagnetic tracking systems. The goal of this project is to: 1) address the issues of magnetic field distortion that affects the measurement accuracy of EM tracking systems and study its effects, 2) investigate methods of compensating for the measurement distortion, and 3) use this information in optimizing the design of EM tracked tools for particular scenarios. The overriding goal is to produce a method for optimal design of surgical tools and reference frames for ENT surgery.

My main contributions start off with characterizing the field distortion; this is done by gathering large quantities of measurements in the measurement volume and comparing them to reference measurements, which can be used to generate error maps. These maps are generated for many different environments, including the operating room. Data collection is performed using several different data collection techniques including optically tracked robotic collection and collection with a digitized stand-alone calibration fixture.

Having the map of translation and rotation errors as a function of sensor position and orientation, this information is used to characterize the distortion field with polynomial models. I have done much work in the optimization of these polynomial fits in terms of computation speed, accuracy, and practicality. When making these models, there are many factors that affect the results; these include the number of data points collected, the spatial and orientation distribution of the data points, the order of the polynomial fit, and the number of base orientations that the orientation space is subdivided into. The goal is to determine the best solution to allow for practical compensation in an OR setting. This work is based off of the preliminary work by Xiaohui Wu [Wu, 2003c].

Using these distortion models, the core of my work involves design of tools for use with EM trackers. This includes analysis of error propagation through the tool registration frames, as well as determining the optimal sensor configuration that minimizes tool uncertainty. In addition to analytic work, I am presenting a tool simulator that uses this distortion model; this allows for an arbitrary tool and reference frame to be 'built' and tested empirically in a virtual environment before ever even building the actual tools. All of this information is to be used to generate tool configurations that provide optimal tracking accuracy.

The final aspect is to use the tool design and simulation framework to develop instrumentation for ENT applications. The idea is to generate a patient-fixed reference frame and several tools, including tissue shavers, pointer, and endoscopes that can be used as accurately as possible for image guided sinus surgery.

## **Chapter 2: Measurement Error Analysis**

### ***Overview***

It is very important to be able to map the errors in the distorted measurement field of an electromagnetic tracker. Such a mapping is essential for understanding how error is affected by the environment, and is also critical for any further error analysis. As described in the Literature Review section of Chapter 1, the idea of mapping the distortion field has been studied fairly extensively by several groups. What we have done here is to further study the distortion fields for the Aurora EM tracker in a variety of environments using several different data collection techniques. This data is extensively analyzed to determine trends and relationships in the underlying error fields.

This section first details how to accurately determine a map of the error field for an EM tracking system. The key to this procedure is registering the EMTS sensors to a ground truth that can be used for determining the reference position and orientation of the EM sensors. In our trials, the NDI Optotrak OTS is used as a reference for the NDI Aurora EMTS. This is a legitimate choice because the Optotrak has a specified accuracy that is about one order of magnitude better than that of the Aurora; also, it is effectively immune to field distortion. Therefore, the EMTS sensors are moved throughout the working volume of the Aurora, while being attached to an OTS rigid body (RB). By registering the EM sensors to this OTS RB, and registering the Aurora, which is also tracked by the Optotrak, it is possible to know the ‘ideal’ position and orientation of each sensor. These ‘ideal’ reference measurements are then compared to the sensor measurements obtained from the EMTS for the same time interval. This section details the registration procedure, methods of data collection, methods for determining the error map, visualization of the error field, and statistical analysis of the error.

## ***Techniques for Mapping Error Field***

In order to generate a mapping of the error field of a tracking device, a data collection protocol must be chosen. There are many ways in which data can be collected, and in the first part of this section some of these techniques are discussed. As would be expected, the ‘optimal’ data collection technique is not necessarily universal; it will depend very much on the particular application.

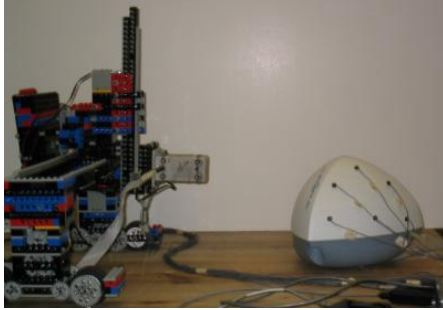
### **Dynamic vs. Static Data Acquisition**

Dynamic data collection refers to gathering data points as a hybrid, combined OTS-EMTS rigid body is moved through the measurement volume without stopping to collect the measurements from the two systems. This sounds ideal because it allows for an enormous amount of data to be collected in a very short time period. However, there are several problems that arise: 1) inherent decreased tracking accuracy for moving targets, 2) latency between the readings and the actual measurements, 3) temporal calibration between the two tracking systems becomes necessary, and 4) uneven distribution of data (if collected manually). It is possible to compensate for the problems two and three by performing a temporal calibration that measures the skew between the times that a measurement is available in each system. Prior work by Xiaohui Wu [Wu, 2003c] discusses potential methods for performing such a temporal calibration to solve for the time skew. However, it was deemed that such a data collection was not ideal and instead a quasi-static approach is used. Here, the tool is placed in a given position, a reading is taken from both systems, and then it is moved to the next location. This is a particularly ideal scenario for robotic data collection.

### **Robotic vs. Manual Data Collection**

Data can be collected using automatic means or completely manually. Manual data collection falls in two categories: 1) freehand where a calibration wand is waved in the field, and 2) guided data acquisition by a calibration fixture. The first method is fast and collects a lot of data, but it is prone to lower accuracy due to its dynamic collection nature and will result in an uneven distribution of data. The second method, which is described in more detail next, does provide for well-distributed data that can be collected in a quasi-static method; however, it is extremely tedious. Robotic data collection, on the other hand, is ideal for collecting a large number of data points with a predetermined data distribution using a quasi-static collection mode, while requiring minimal user interaction. Examples of two methods of robotic data collection used in our lab are shown in Figure 2-1. The disadvantage of robotic collection is that it requires an entourage of equipment including the robot, PC, controller, and usually the Optotrak; therefore it is probably not practical for data collection in an OR setting, at least not with current configuration.

*Plastic Lego Robot Data Collection*



[Wu, 2003c]

*LARS Robot Data Collection*



Examples of tracked robotic data collection techniques

**Figure 2-1**

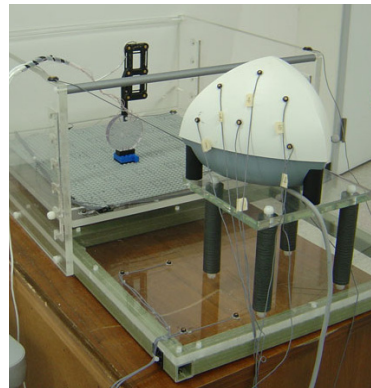
### **Tracked (Hybrid) vs. Digitized (Stand-alone fixture) Registration**

Obviously, in order to get a mapping of systematic error, we need a ground truth. In general, the Optotrak OTS proves to be a great resource due its high accuracy that is about one order of magnitude better than that of the Aurora EMTS. When using the Optotrak for data collection, registration is performed as mentioned previously in the registration section. However, due to its lack of portability, the Optotrak is not ideal for bringing into an OR setting for digitizing the field. And a robot tracked by the Optotrak is especially prohibitive. Therefore, the Stand-alone (SA) calibration fixture was constructed to allow for portability. The system is digitized with the Optotrak, and due to its repeatability that is on the order of the Optotrak measurements, no external tracking is necessary during data collection to serve as a reference. Images of the tracked, robotic data collection equipment are shown on the left in Figure 2-2; on the right is the Stand-alone fixture while being calibrated / digitized with the Optotrak.

*Tracked Optotrak Collection  
(with LARS robot)*



*Digitized Stand-alone Collection*



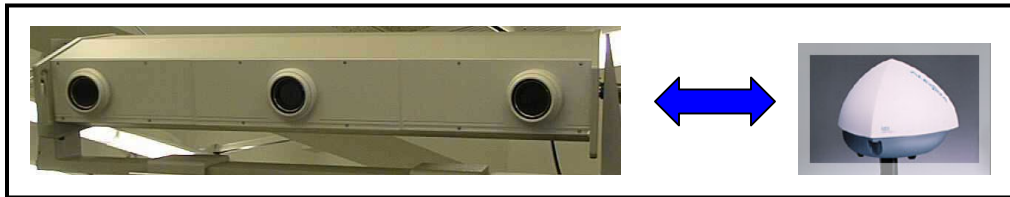
Real-time tracked data collection vs. pre-digitized data collection

**Figure 2-2**

## ***Tracker Registration***

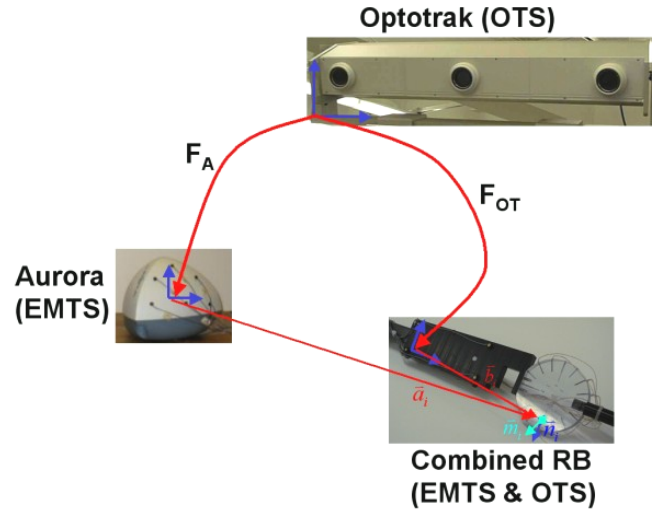
### **Overview**

The key to accurate error measurements is having an accurate reference measurement to pair with each sensor measurement. For this, the Northern Digital Optotrak 3020 is used as a ground truth reference for the actual measurement. However, it is clear that it is impossible for the Optotrak OTS to directly track the same sensor as the Aurora EMTS, so the two systems must co-registered together to form what is often termed a *hybrid system*. There are two methods of forming such a system that are described here: the first details the standard method of registering the Aurora system to the Optotrak so that we can get real-time reference measurements, and the second is to digitize a Stand-alone fixture with the OTS such that it can later be used without an external tracker to provide the reference.



## Registration of Aurora to Optotrak

The goal of registration is to determine the position and orientation of the EMTS sensors as measured by both the Aurora and the Optotrak such that they are in a common coordinate system. Figure 2-3 shows the frame transformations involved in determining the corresponding EMTS measurements and the OTS references.



Hybrid OTS-EMTS tracking system frame transformations

**Figure 2-3**

The relationships that relate the position and orientation as reported from the two measurement systems is as follows:

**For Position:**

$$F_A \bar{a}_i = F_{OT} \bar{b}_i$$

**For Orientation:**

$$R_A \bar{m}_i = R_{OT} \bar{n}_i$$

Where:  $[\bar{a}_i, \bar{m}_i]$  Measured sensor position and orientation with respect to the EMTS base frame (FG frame)

$[\bar{b}_i, \bar{n}_i]$  Sensor position and orientation with respect to the OTS tool RB (known tool configuration)

$$F_A = \begin{bmatrix} R_A & \bar{p}_A \\ 0 & 1 \end{bmatrix} \text{ EMTS base frame represented in OTS coordinates}$$

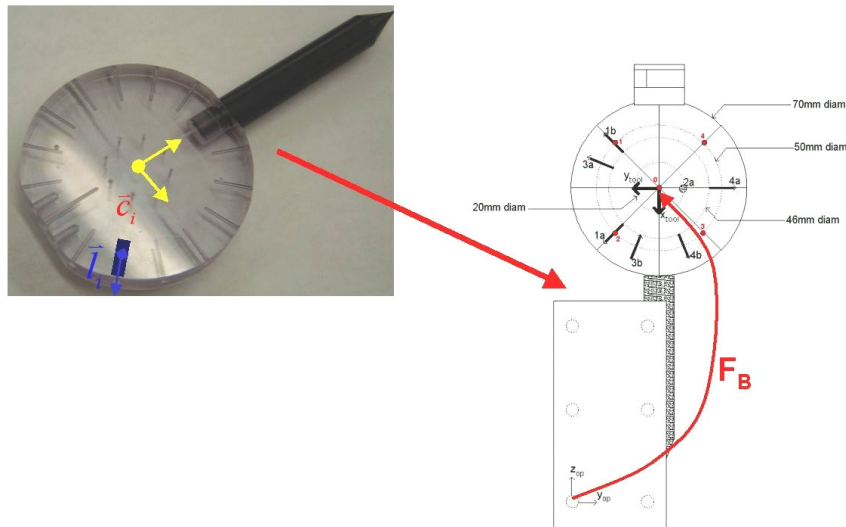
Note: This is really a composite frame as described later containing the measurement of the OTS RB on the Aurora FG unit and the transformation from that RB to the origin of the Aurora FG base.

$$F_{OT} = \begin{bmatrix} R_{OT} & \bar{p}_{OT} \\ 0 & 1 \end{bmatrix} \text{ OTS tool RB frame as measured by the OTS}$$

## Registration of Aurora Sensors to Hybrid Rigid Body

The above description jumps right in and makes the assumption that the position and orientation of the Aurora sensors with respect to the Optotrak rigid body (RB),  $[\bar{b}_i, \bar{n}_i]$ , are known. These values can be calculated, but they cannot be measured or known directly without some computation as will be described below.

The combined RB used here was designed using CAD software and precisely manufactured in-house using our lab's CNC machining capabilities with an angular accuracy of  $0.05^\circ$  and translational accuracy of 0.01mm. This is about one order of magnitude better than the OTS and about two orders of magnitude better than the EMTS, so the accuracy is sufficient for these trials. Therefore, sensor locations and orientations are assumed to be known accurately with respect to a fiducial pattern that was machined on the surface of the rigid body using the same process. After affixing an Optotrak RB to the calibration device, the Aurora sensor coils can be registered to the Optotrak rigid body (RB) by touching the known 'divots' on the device with a pre-calibrated Optotrak pointer. A diagram of the hybrid optical and EM tool is displayed in Figure 2-4. This particular configuration is used for digitizing the Stand-alone fixture. Figure 2-5 shows the same tool in the configuration used for hybrid tracked data collection.



Aurora 'pinwheel' tool with six precisely placed 5 DoF sensors and attached to an Optotrak rigid body

Figure 2-4



The pattern of sensors with respect to the device's frame is described as follows:

**For Position:**

$$\vec{b}_i = F_B \vec{c}_i$$

**For Orientation:**

$$\vec{n}_i = R_B \vec{l}_i$$

Where:

$[\vec{c}_i, \vec{l}_i]$  Sensor position and orientation with respect to the frame of the physical tool (precisely known from design and manufacturing specifications)

$F_B = [R_B, \vec{p}_B]$  Transformation to the physical tool's frame from that of the OTS RB (determined by digitizing known points tool with an OTS pointer)

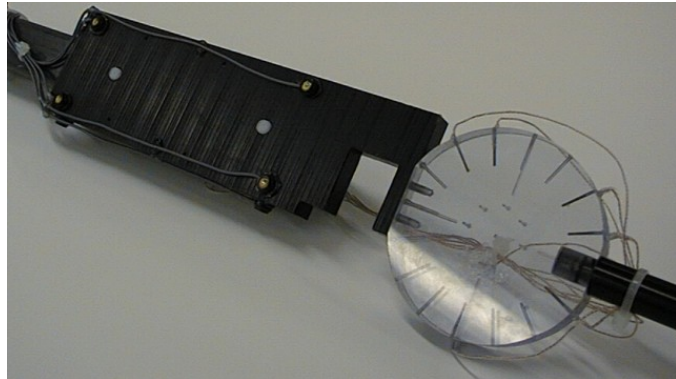
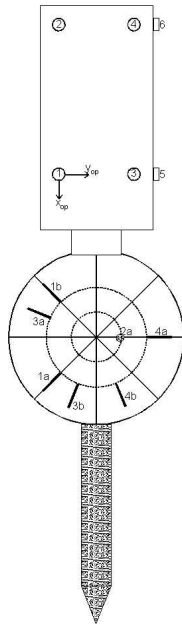
Plugging these transformations into the equations from before:

**For Position:**

$$F_A \vec{a}_i = F_{OT} F_B \vec{c}_i$$

**For Orientation:**

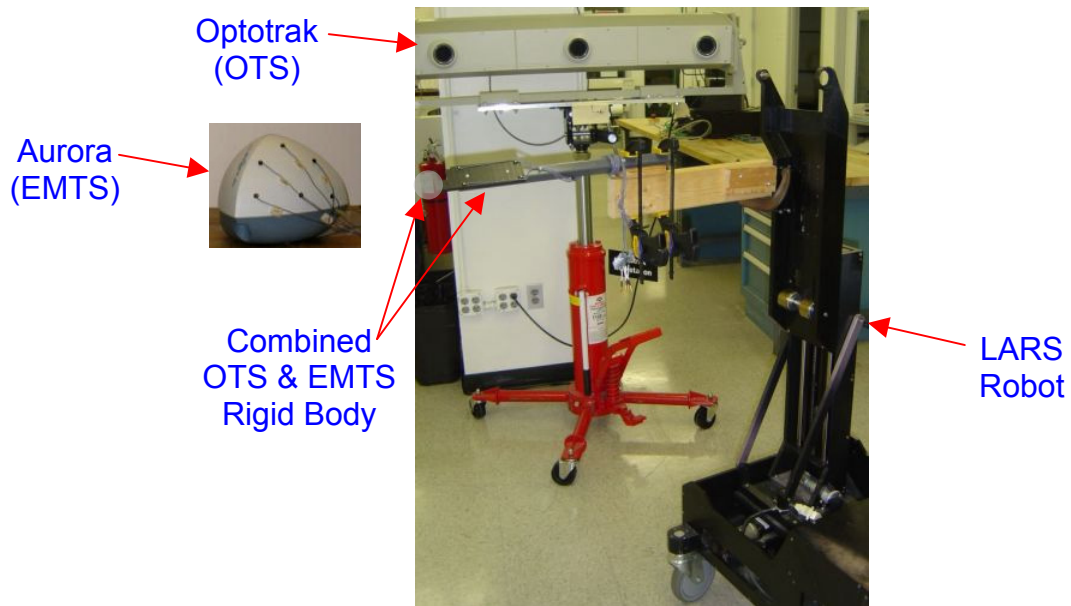
$$R_A \vec{m}_i = R_{OT} R_B \vec{l}_i$$



Hybrid OTS / EMTS calibration wand  
(Sensor configuration labeled on left)

**Figure 2-5**

The hybrid electromagnetic-optical rigid body used for tracked data collection applications is shown in Figure 2-5. This comprises of six separate 5 DoF Aurora sensors placed in known locations and orientation in the ‘pinwheel’ tool. The sensors labeled *1a*, *1b*, *3a*, *3b*, *4a*, and *4b* are used. The custom Optotrak RB containing six IREDs on two sides of a Delrin block is affixed to the Aurora tool and the EM sensors are registered to the OTS RB as just mentioned previously. The whole combined RB, known as the *calibration wand*, is mounted on a 1” square box fiberglass arm with nylon hardware, that is in turn held by the robot at a distance of at least 1 meter away. Figure 2-6 shows the LARS robot holding the calibration wand in front of the Optotrak.

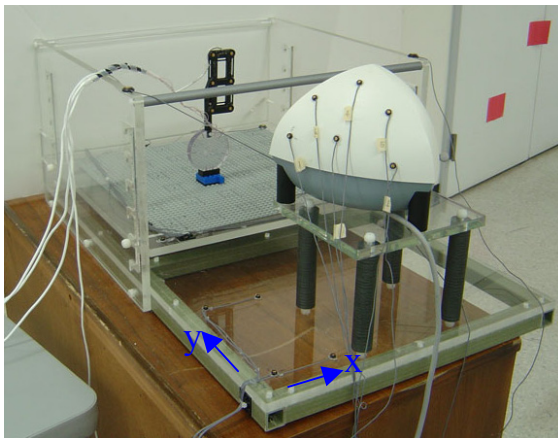


Configuration for tracked robotic data collection

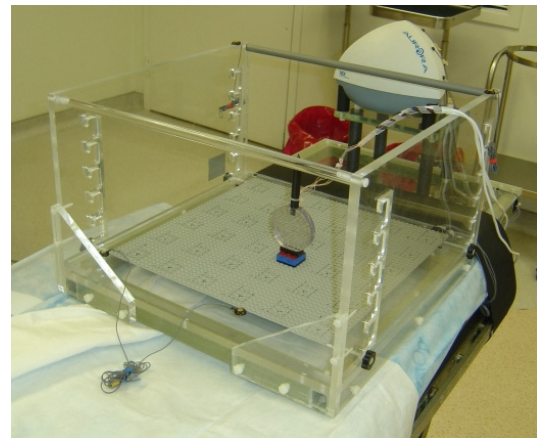
**Figure 2-6**

## Registration of Stand-Alone Fixture

As mentioned earlier, the alternative to real-time external tracking is to digitize a calibration fixture that can place sensors in a set of known, repeatable poses that were previously digitized. Figure 2-7 shows the Stand-alone (SA) calibration fixture. It is constructed of laser cut 12mm thick acrylic plate mounted on a 1" square box fiberglass frame to enhance the rigidity; all mounting hardware is made of nylon. The Aurora unit is very repeatably mounted on a Delrin base. To the left is the configuration used for calibrating/digitizing the fixture. To the right is the fixture being used in the OR.



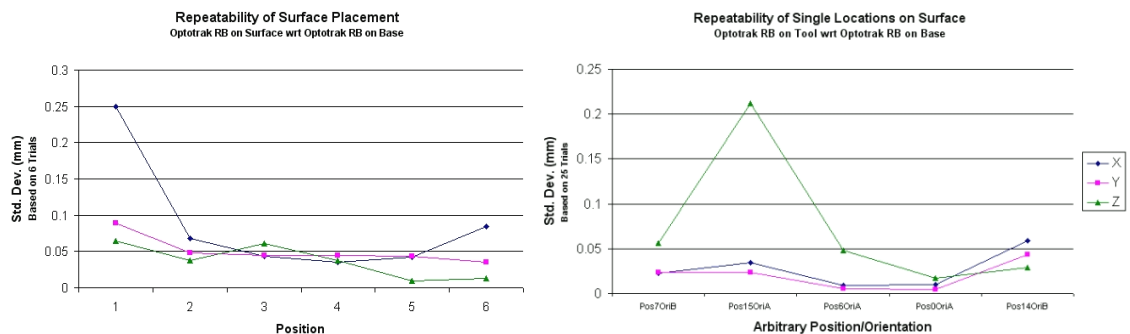
Stand-alone fixture calibration



Stand-alone fixture in an operating room

**Figure 2-7**

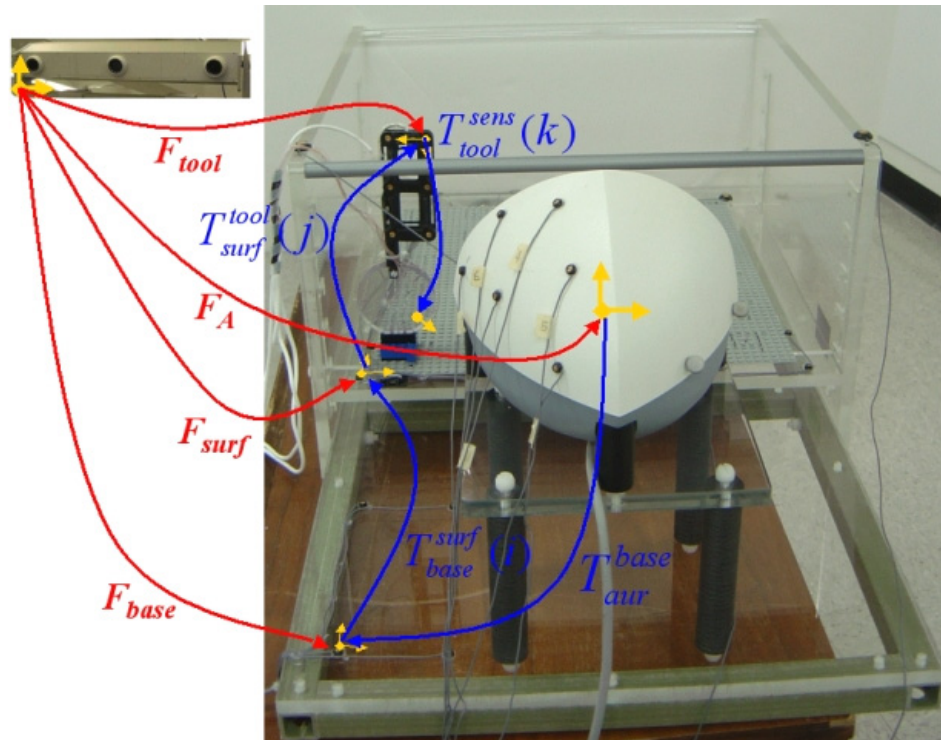
The following plots in Figure 2-8 show the placement repeatability of the surface (12mm thick acrylic plate with the gray LEGO sheet affixed) on the left, and the placement repeatability of the tool on the surface (12mm thick,  $\varnothing 70$ mm polycarbonate cylinder with Optotrak RB affixed) on the right. The surface was placed five times at each of the six heights and the results show the standard deviation for each translational direction. The tool placement results, on the right, show the standard deviation of the positions for 25 trials at each of five random poses out of the 72 possible poses on the surface. This includes uncertainty in OTS measurements and LEGO block placement.



Repeatability of tool placement using the Stand-alone fixture

**Figure 2-8**

Calibration of the SA system is different than that of the standard optical-EM hybrid tracker system. The registration of the sensors in the RB is performed as before to locate the Aurora sensors with respect to the Optotrak RB. However, to determine where these sensors are supposed to be at each possible configuration in the fixtures, as series of frame transformations must be determined. Figure 2-9 shows the setup with the frame transformations overlaid, where Red transformations are measured by the Optotrak and used for digitizing fixture, and Blue transformations are calculated, logged, and used for later data collection. Since the system has various discrete configurations, some of the frame transformations are a function of the current fixture configuration (i.e. the transformation from the base to the moving surface and the transformation from that surface to the tool).



Frame transformations of the digitized Stand-alone calibration fixture  
**Figure 2-9**

Where:

$i = \{1 \dots 6\}$  Surface heights

$j = \{1 \dots 72\}$  Tool poses

$k = \{1 \dots 6\}$  Sensors in tool

$F_i$  is used to denote frames measured by the Optotrak

$T_i$  is used to denote calculated relative frame transformations

Note that  $F_A$  is actually a composite transformation:

$$F_A = F_{OT}^{AurRB} F_{AurRB}^{FGOrigin}$$

Where:  $F_{AurRB}^{FGOrigin}$  is a known, constant transformation between the Optotrak RB frame on the Aurora FG and the Aurora FG's base measurement frame

$F_{OT}^{AurRB}$  is the measurement for the frame of the Optotrak RB on the Aurora FG with respect to the Optotrak

Writing out equalities that are equivalent to those in the earlier registration section for the calibration wand in the hybrid system, we get the following equations. These equations represent the ideal sensor positions and orientations for a given configuration of the SA fixture.

**For Position:**

$$\vec{b}_{i,j,k} = T_{aur}^{base} T_{base}^{surf}(i) T_{surf}^{tool}(j) \vec{p}_{tool}^{sens}(k)$$

**For Orientation:**

$$\vec{n}_{i,j,k} = R_{aur}^{base} R_{base}^{surf}(i) R_{surf}^{tool}(j) \vec{R}_{tool}^{sens}(k)$$

Where:

$\vec{p}_{tool}^{sens}(k) = R_B \vec{c}_k + \vec{p}_B$  is the sensor position in the Optotrak RB's frame.

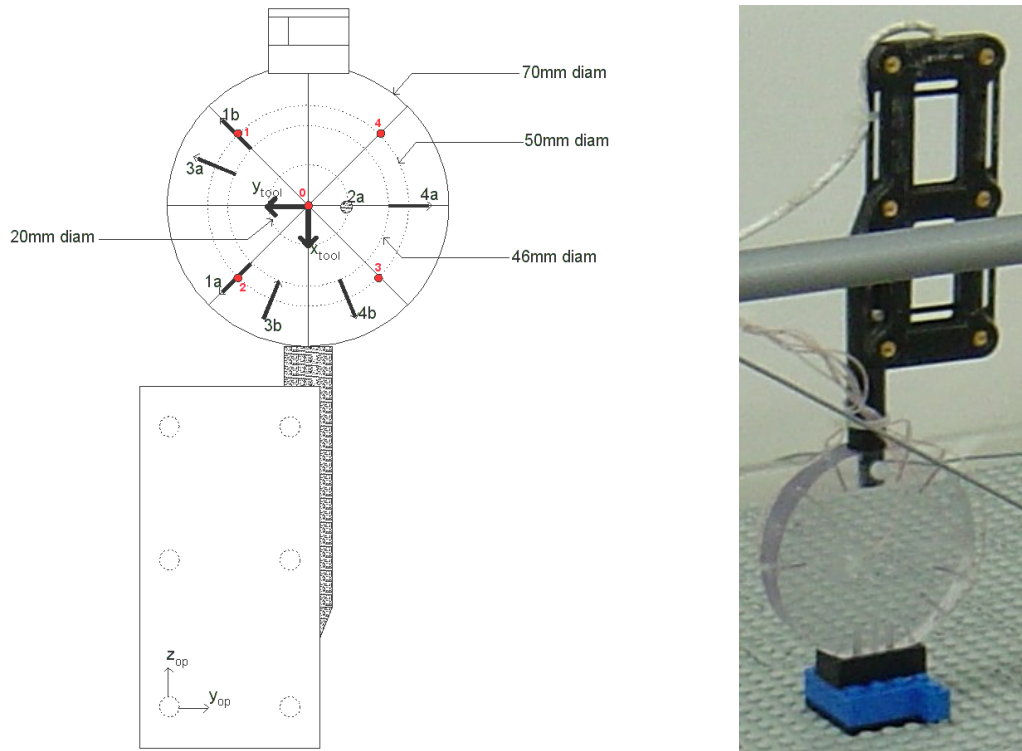
$\vec{R}_{tool}^{sens}(k) = R_B \vec{l}_k$  is the third column of a rotation matrix representing the orientation of the sensor in the Optotrak RB's frame.

As before:

$[\vec{c}_i, \vec{l}_i]$  Sensor position and orientation with respect to the EM tool's frame (known accurately from design and manufacturing specifications)

$F_B = (R_B, \vec{p}_B)$  is the transformation from to the EM tool's frame from that of the attached Optotrak RB.

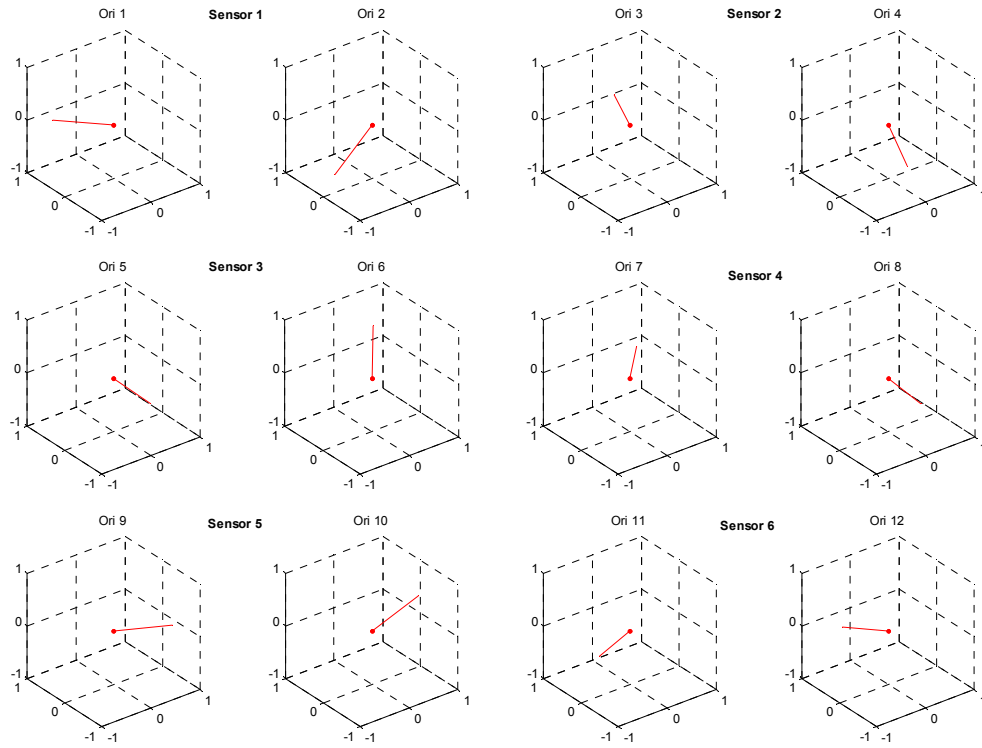
A close-up photo of the hybrid rigid body and a schematic of its configuration is shown in Figure 2-10. This combined rigid body is based upon the same ‘pinwheel’ EM tool that was used in the previous section. To the left are design specifications for the tool, definitions of frames for the tool and Optotrak RB, and sensor numbering. To the right is a photograph of the actual device being used to digitize the SA fixture. All of the optical markers are unnecessary after the fixture has been digitized; the rigid body shown in the figure is removed for actual data collection for field characterization. If it were not removed, we have noticed that the optical markers can cause distortion and errors in the EM measurements. When not actively strobing/flushing the IREDS on the RB, the effect is present but limited; when the IREDS are flashing the error is more dramatic.



Combined optical/EM rigid body for digitizing the stand-alone calibration fixture  
(Optotrak RB is removed after digitizing is complete)

**Figure 2-10**

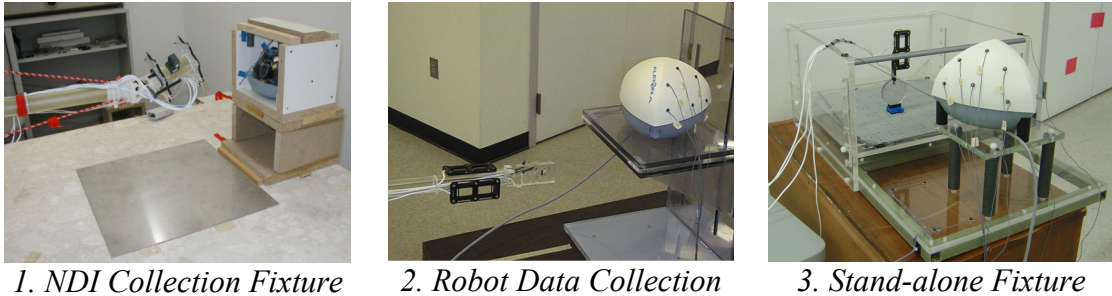
The sensor orientations that are attainable by using the ‘pinwheel’ tool in the Stand-alone fixture are shown in Figure 2-11. The fixture is arranged such that measurements of the 6 sensors on the tool can be taken at two separate tool orientations at each point in the grid. Therefore, a total of 12 orientations are achievable. These orientations, with respect to the FG, are consistent though the mapping of the volume. Since all 12 orientations are present for each of the 216 locations, we get a very even distribution of data.



Sensor orientations obtained using the Stand-alone fixture  
**Figure 2-11**

## Field Distortion Mapping Results

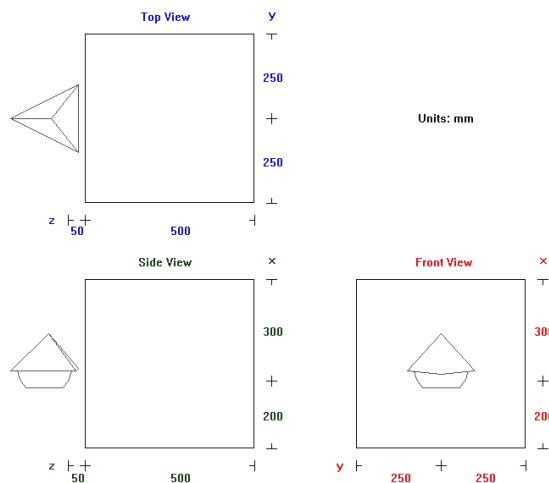
Mapping the measurement error present in the working volume of the tracker is key to understanding the effects of field distortion on measurement accuracy. This section displays results that include error fields and error analysis for various environments that were characterized using the previously mentioned techniques. These results show the systematic error as obtained through accuracy measurements; this is before any compensation is applied.



Data collection methods used for error field characterization

**Figure 2-12**

Data sets used here come from three separate sources: 1) measurements provided by Stefan Kirsch and Christian Schilling from NDI Europe, 2) measurements collected by the LARS robot in the lab and monitored by an NDI Optotrak as a reference, and 3) measurement collected using the digitized Stand-alone fixture. Figure 2-12 shows representative images of the three collection methods used. The common feature, regardless of the collection technique, is that at the end we have two sets of measurements for each point: a reference data set and a measurement data set. The coordinate system of the Aurora for all of the data sets is shown in Figure 2-13.



Aurora coordinate system and working volume  
(Image capture from NDI ToolViewer software)

**Figure 2-13**

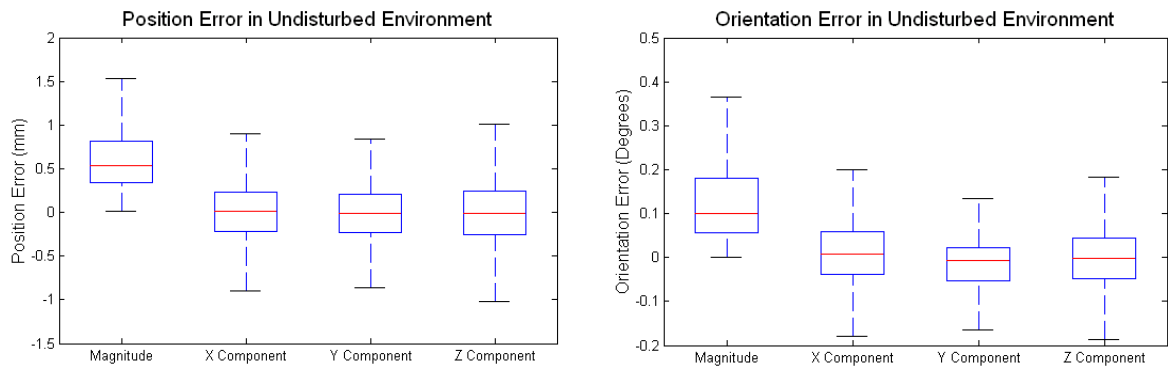


## Ideal Environment

As a baseline for accuracy of the Aurora system, the data set provided by Northern Digital Europe (NDI Data Set #1) is used. This set, which includes EMTS measurements and calibrated OTS reference measurements for two separate sensors, was collected in as distortion-free an environment as possible and demonstrates the close to the ideal operation of the Aurora system. Therefore, the results that follow represent the systematic error of the tracker itself.

## Error Distribution

Looking at the entire data set, the position and orientation errors for each of the 11,704 measurements were calculated. The box plots in Figure 2-14 represent this error. On the left of each plot is the distribution of the magnitude of the error; the three columns on the right of each plot represent the distribution of the error along each component. For position, this representation is straightforward with the error being split into the X,Y,Z components. For orientation, this represents the angle corresponding to the X,Y,Z components of the Rodriguez vector representing the orientation. Note that the components are signed errors, and that the total error is an unsigned magnitude. For more detail on this representation, see the Definition of Error section in the beginning of Chapter 3.

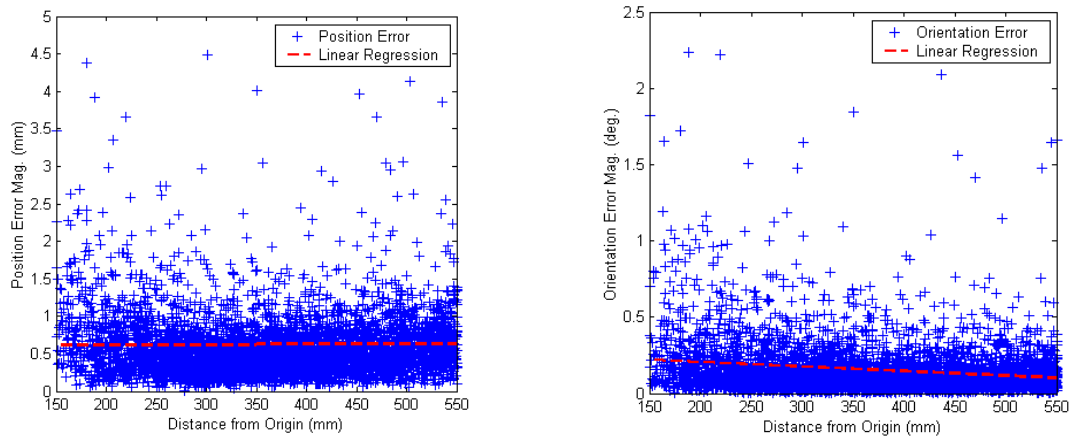


Measurement error in the 'ideal' environment separated into components  
(Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)

**Figure 2-14**

## Error as a function of distance from Field Generator

By plotting the measurement error as a function of Euclidean distance from FG (Figure 2-15), it is noted that there is **not** a significant positive relationship for this data set. This result is counterintuitive and deviates from the results described by several earlier studies with EM tracking systems. There is quite a bit of literature describing error being proportional to the distance of the sensor from the FG, including [Birkfellner, 1998], [Livingston, 1997], and [Leotta, 1995]. This discrepancy is most likely due to the fact that this is a relatively undisturbed environment and therefore the magnetic fields are actually very close to that of the physical model upon which the system was calibrated. Thus, the remaining error is mostly uncalibrated noise. The following plots and data analysis are based on all 5531 data points from one physical sensor, *Sensor 0*, used for the data collection. The other sensor produces similar results, but does have a slightly larger positive slope for the position error relationship.



Measurement error plotted vs. distance from field generator in ‘ideal’ environment  
**Figure 2-15**

For the above plots, the equations representing the displayed linear regression are:

$$e_{pos} = 0.6094 + 4.473e-005 * d_{FG}$$

$$e_{ori} = 0.2662 - 2.966e-004 * d_{FG}$$

The p-values representing the relationships are:

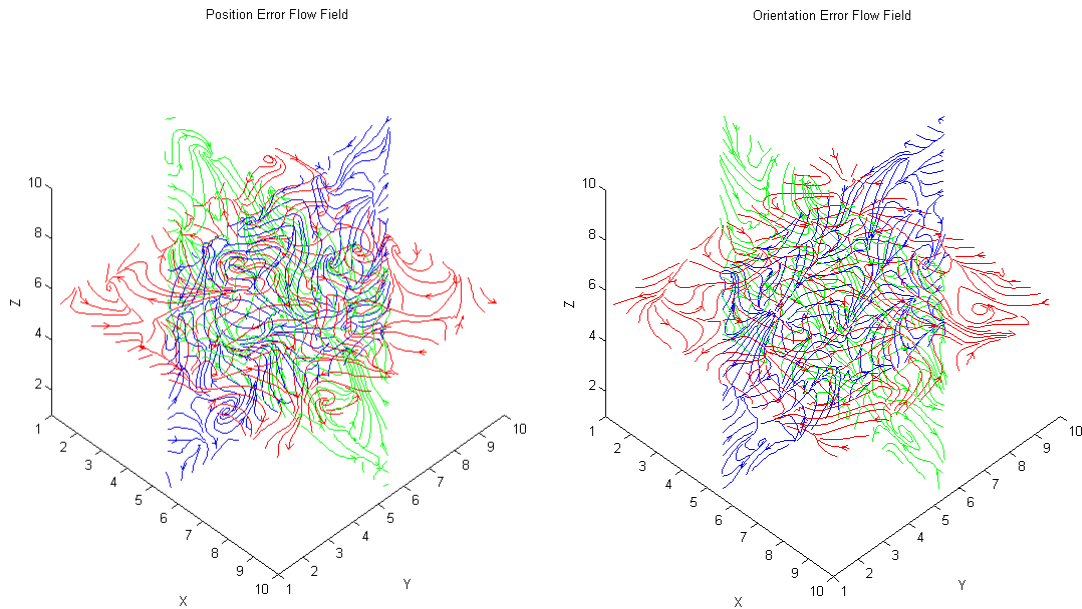
$$p - value_{pos} = 0.4213$$

$$p - value_{ori} < 0.0001$$

Therefore, for the experiments in an undisturbed, ‘ideal’ environment, we can say that distance of the sensor from the FG does not have a significant effect on the position error. We also see that distance from the FG does in fact affect the orientation error, but error and distance have a negative correlation.

## Error Field

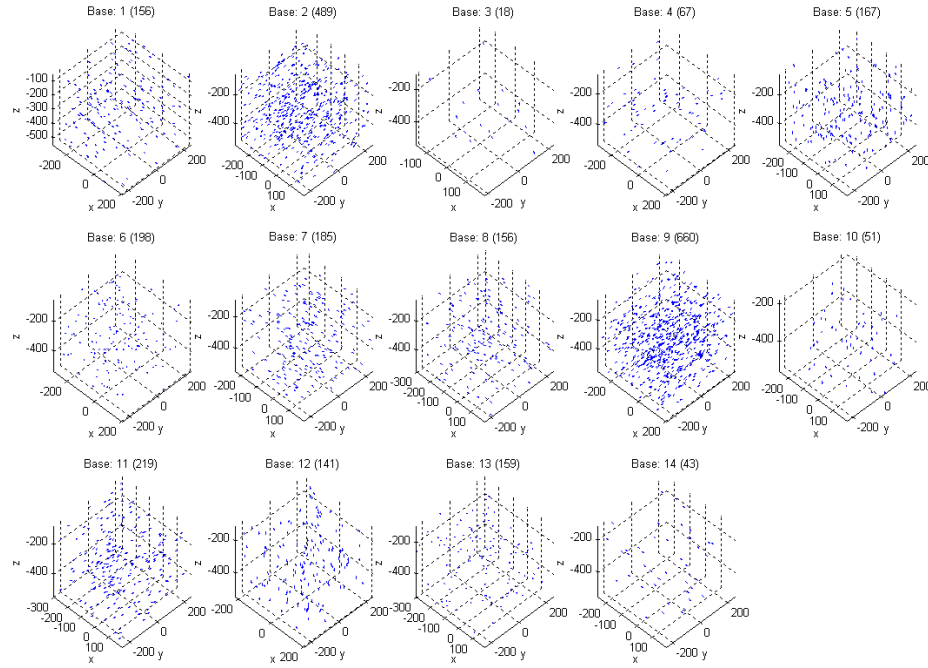
The plots in Figure 2-16 are representative of slices through the vector field generated by plotting the 3D position and 3D orientation errors of 1000 random points from this data set in the appropriate locations, normalized into a 10x10x10 grid; all orientations are included. **The results show that in the undisturbed environment, the systematic error, although small, has very high field curvature and large relative spatial variation.** It is clear that the plots do not show a smooth pattern like that of the results obtained distorted environments. There is also very little consistency to the systematic distortion as is later verified when polynomials are fit to the distortion data and are unsuccessfully applied to an independent set of data points. Therefore, we are just about at the theoretical limit of the accuracy of the system and most of the remaining error is due to noise. However, it should be noted that this plot includes all orientations. When the measurements are grouped into similar orientations, the results do have somewhat more of a distinct pattern as can be seen later in Figure 2-21 and Figure 2-22.



Vector flow fields representing the curvature of the error fields in three planes for data collected in an undisturbed environment (normalized into a 10x10x10 grid)  
**Figure 2-16**

## Separating Undisturbed Set by Orientation

In order to further investigate the error distribution, the sensor readings are subdivided by their corresponding orientations. Measurements are split such that they are grouped by measurements that fall within  $20^\circ$  (i.e. a  $40^\circ$  cone) of each base orientation; the definition of base orientations is given later when discussing error compensation, but for now they can be assumed to a set of 14 evenly distributed orientations. The plot in Figure 2-17 shows the distribution of data points for *Sensor 0*; *Sensor 1* has a similar distribution. Figure 2-18 shows the distribution of the base orientations used.

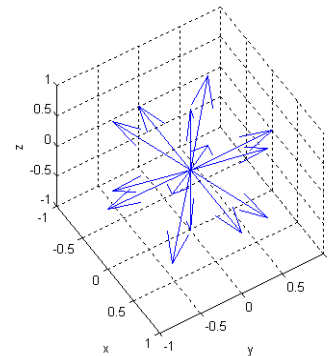


Measurements from undisturbed data (NDI set #1) separated by orientation (grouped measurements within  $20^\circ$  of base orientations)

**Figure 2-17**

Base orientations:

	<b>X</b>	<b>Y</b>	<b>Z</b>
1:	1	0	0
2:	0	1	0
3:	0	0	1
4:	0.57735	0.57735	0.57735
5:	-0.57735	0.57735	0.57735
6:	0.57735	-0.57735	0.57735
7:	-0.57735	-0.57735	0.57735
8:	-1	0	0
9:	0	-1	0
10:	0	0	-1
11:	-0.57735	-0.57735	-0.57735
12:	0.57735	-0.57735	-0.57735
13:	-0.57735	0.57735	-0.57735
14:	0.57735	0.57735	-0.57735

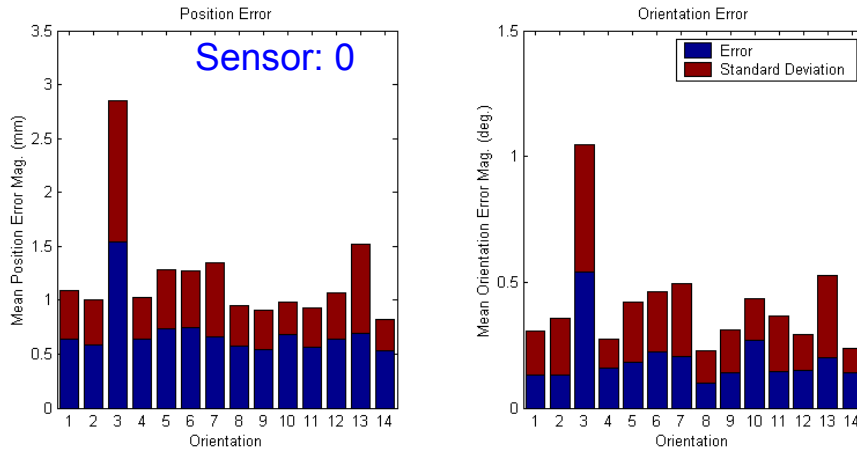


Base orientation vectors used to divide the orientation space of the measurements

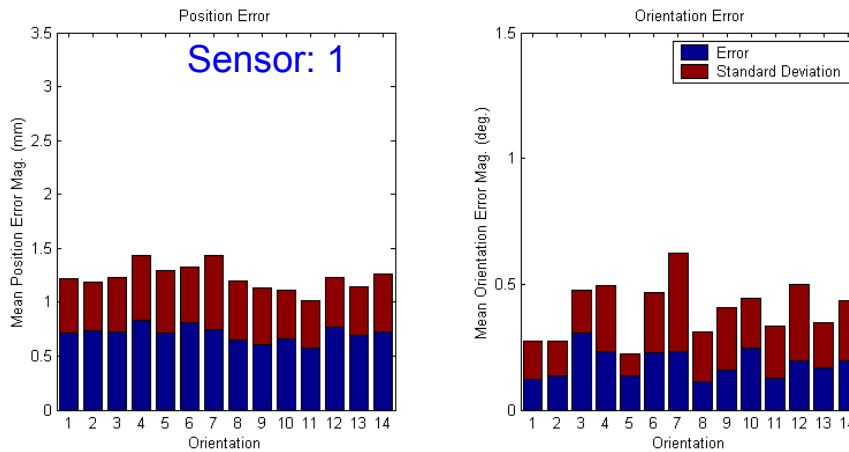
**Figure 2-18**

## Error Analysis by Orientation

The following plots represent the distribution of error as a function of orientation. Again, they are separated into groups of measurements that fall within  $20^\circ$  of each of 14 base vectors. The mean errors are given by the height of the lower, blue bars. The upper, red bar spans one standard deviation past the mean error. The results are displayed for both of the two sensors used during this trial; Figure 2-19 shows the results for *Sensor 0*, and Figure 2-20 shows the results for *Sensor 1*.

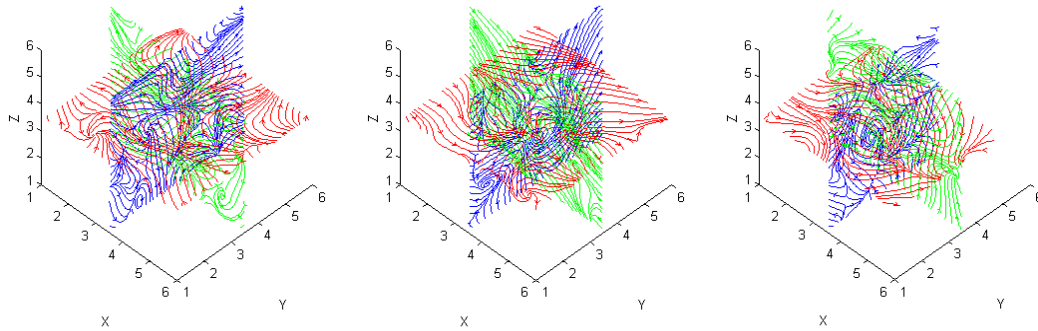


Error magnitude for *Sensor 0* in ‘ideal’ environment separated by orientation  
**Figure 2-19**

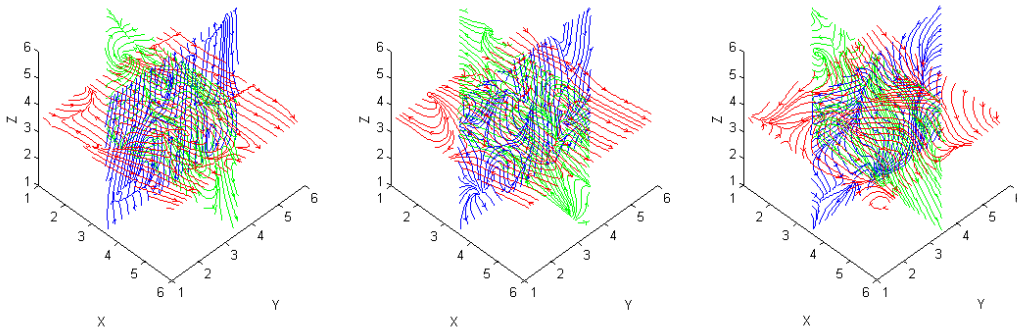


Error magnitude for *Sensor 1* in ‘ideal’ environment separated by orientation  
**Figure 2-20**

The following figures show results similar to those of Figure 2-16 from above, but here the measurements were grouped by orientation. These results show three representative orientations and show the curvature of the error map for both position (Figure 2-21) and orientation (Figure 2-22). These plots show more of a pattern than in the previous plots where all orientations were grouped together, further showing that the orientation of the sensors plays an important part in predicting the error. However, these distortions are still much less smooth than those that will be shown later for the disturbed environment.



**Position** error field curvature in ‘ideal’ environment for three orientations (2,11,12)  
**Figure 2-21**



**Orientation** error field curvature in ‘ideal’ environment for three orientations (2,11,12)  
**Figure 2-22**

## Disturbed Environment

To contrast the previous ‘ideal’ measurements, the same measurement fixture was used to collect a data set in the same environment, but the measurement field was artificially distorted with a 50 cm<sup>2</sup> sheet of steel at the base of the working volume (See Figure 2-23). This data set is again provided by Northern Digital Europe (NDI Data Set #3). As before, included are EMTS measurements and calibrated OTS reference measurements for two separate sensors. The results here represent a highly distorted environment, and are pretty close to the worst-case results for the system.

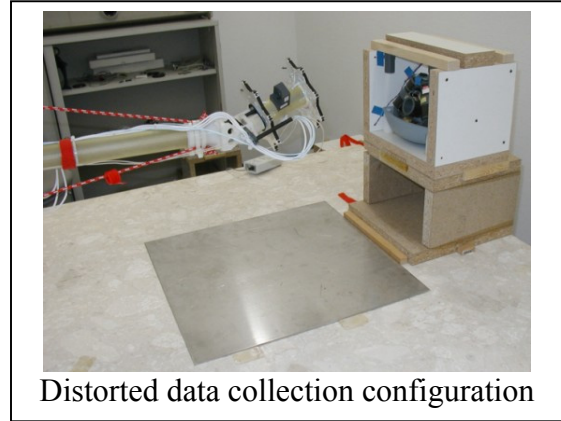
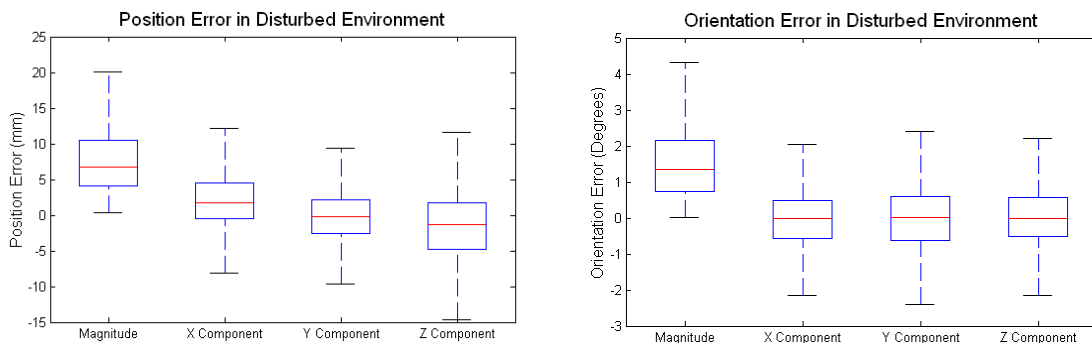


Figure 2-23

## Error Distribution

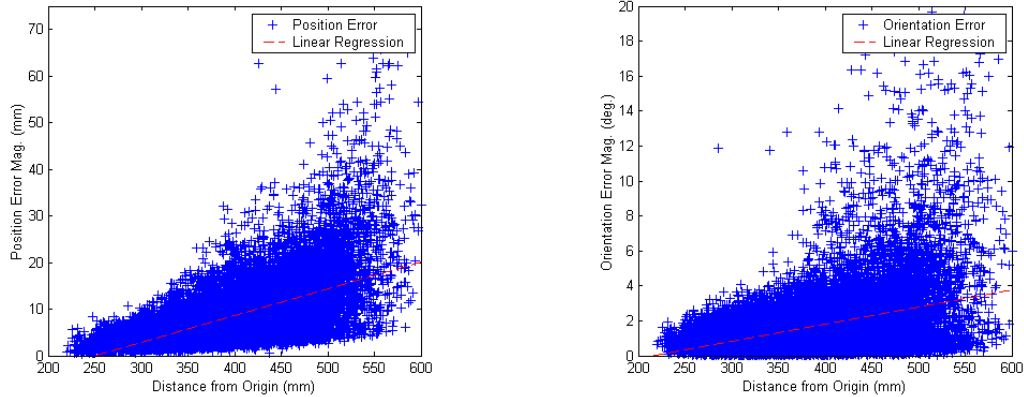
Looking at the entire data set, the position and orientation errors for each of the 22,529 measurements are calculated. The box plots in Figure 2-24 represent this error. As before, on the left of each plot is the distribution of the magnitude of the error; the three columns on the right represent the distribution of the signed error along each component. Notice that the errors are significantly higher for this data set. The RMS magnitude of the position error is 10.450 mm, and the RMS magnitude of the orientation error is 2.465 degrees (0.0430 radians).



Measurement error in the artificially disturbed environment separated into components (Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)  
Figure 2-24

## Error as a function of distance from Field Generator

The trends in Figure 2-25 are much closer to what would be the predicted result than those in Figure 2-15. Here, the error is in fact proportional to the Euclidean distance of the sensor from the FG. This correlates well with other group's findings. The following plots and data analysis are based on all 11,295 data points from one physical sensor, *Sensor 1*, used in the data collection.



Measurement error vs. distance from FG in artificially disturbed environment

**Figure 2-25**

For the above plots, the equations representing the linear regression shown are:

$$e_{pos} = -14.4363 + 0.0577 * d_{FG}$$

$$e_{ori} = -2.1030 + 0.0098 * d_{FG}$$

The p-values representing the relationships are:

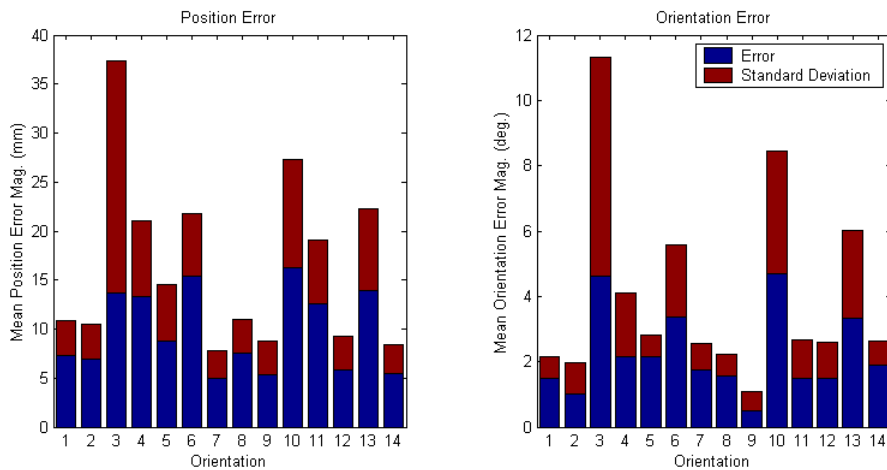
$$p\text{-value}_{pos} < 0.0001$$

$$p\text{-value}_{ori} < 0.0001$$

## Error Analysis by Orientation

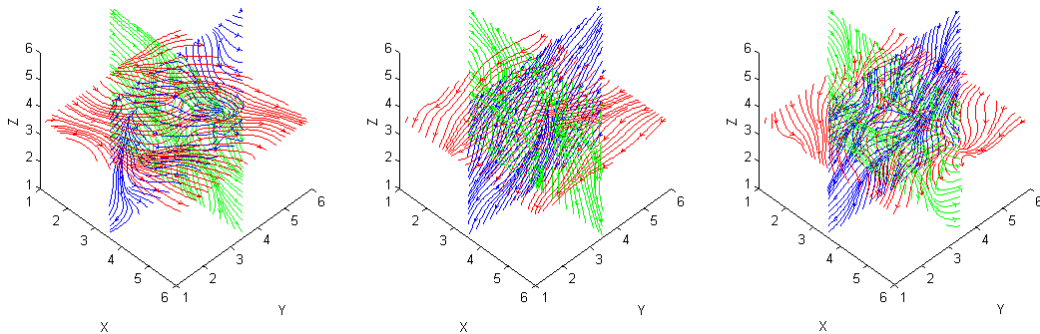
The following plots represent the distribution of error as a function of orientation. Again, they are separated into groups of measurements that fall within  $20^\circ$  of each of 14 base vectors. The mean errors are given by the height of the lower, blue bar and the upper red bar spans one standard deviation past the mean error. The results are displayed in Figure 2-26, and show the results for *Sensor 1*, the results for *Sensor 0* are very similar.



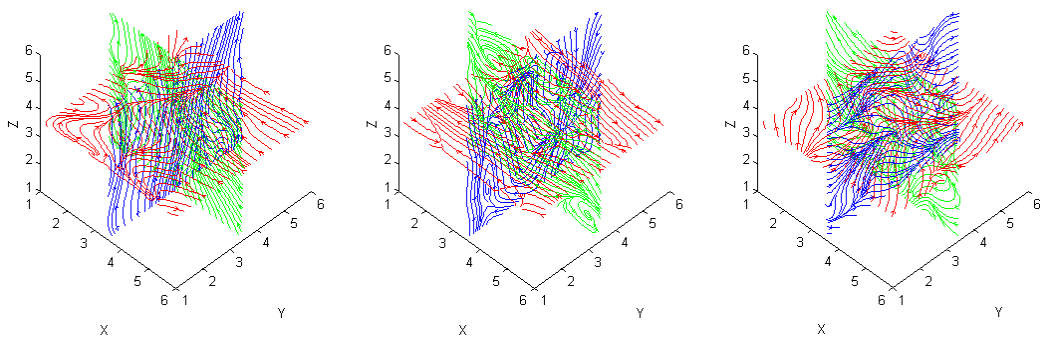


Error magnitude for *Sensor 1* in disturbed environment separated by orientation  
**Figure 2-26**

The following figures (Figure 2-27 and Figure 2-28) show the curvature of the error field for this artificially distorted data set for position and orientation respectively. Although the magnitudes of error are much higher than those of the undisturbed set, notice that the **field curvature is smoother** and seems to be much easier to model.



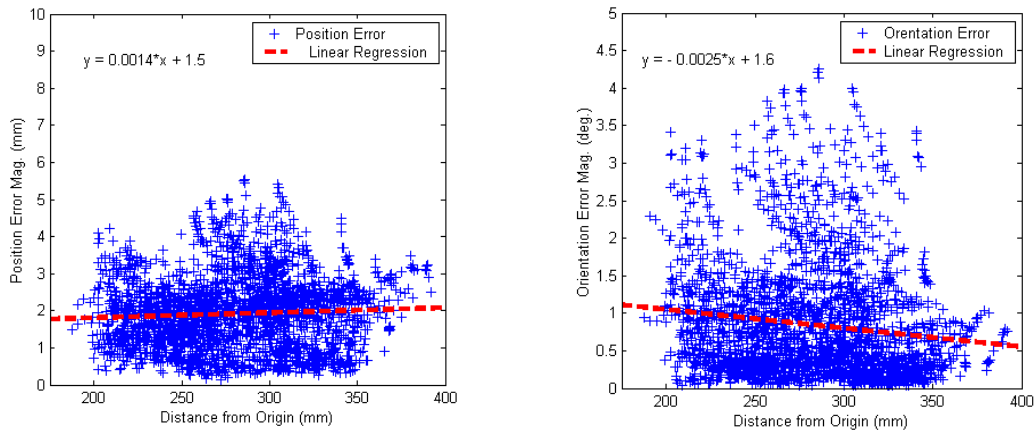
**Position** error field in disturbed environment for three orientations (2,11,12)  
**Figure 2-27**



**Orientation** error field in disturbed environment for three orientations (2,11,12)  
**Figure 2-28**

## Robotic Mapping (Lab)

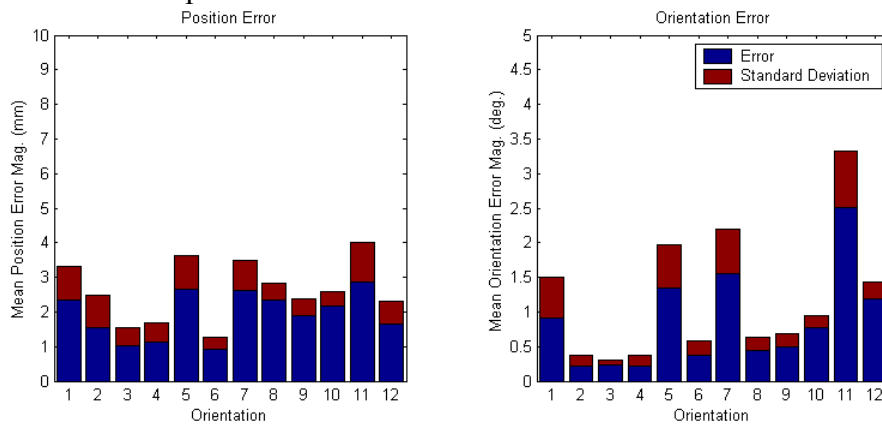
The following results represent those obtained through quasi-static data collection using an optically tracked calibration wand attached to the LARS robotic manipulator. Data is collected using a calibration wand that contains 4 sensors on it; the wand is attached in three configurations and the data is merged to form a data set with 12 distinct, consistent orientations. Since this is really a combination of three data sets, the measurement volumes do not completely overlap. 2000 points (12 orientations each) are selected at random for analysis from the total of 4037 collected measurements. Figure 2-29 presents the magnitude of the measurements error as a function of distance from the FG. The regression equations representing trends in the data are overlaid on the plot. This is a rather undisturbed environment and the results show errors that are higher, but with similar trends to the ‘ideal,’ undisturbed environment from before.



Measurement error plotted vs. distance from field generator for data collected with robot in the lab (2000 random points)

**Figure 2-29**

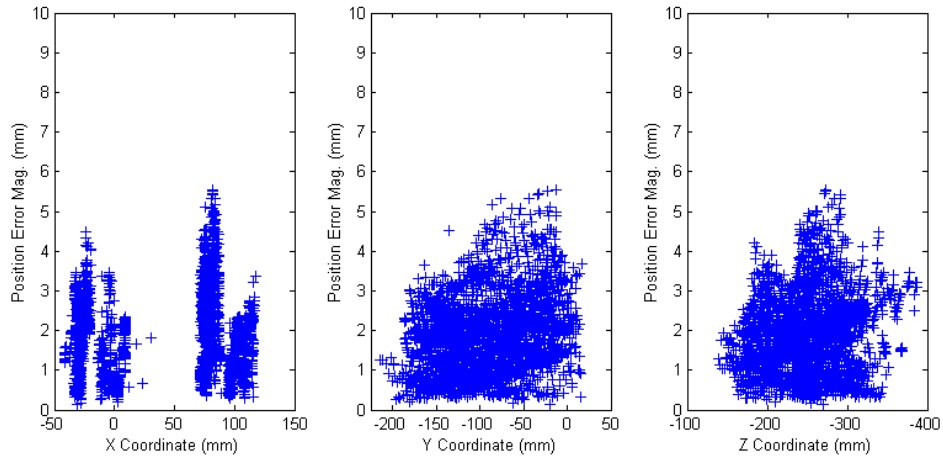
As before, we plot in Figure 2-30 the error as a function of orientation. In this case, each orientation represents one of four sensors from one of three orientations.



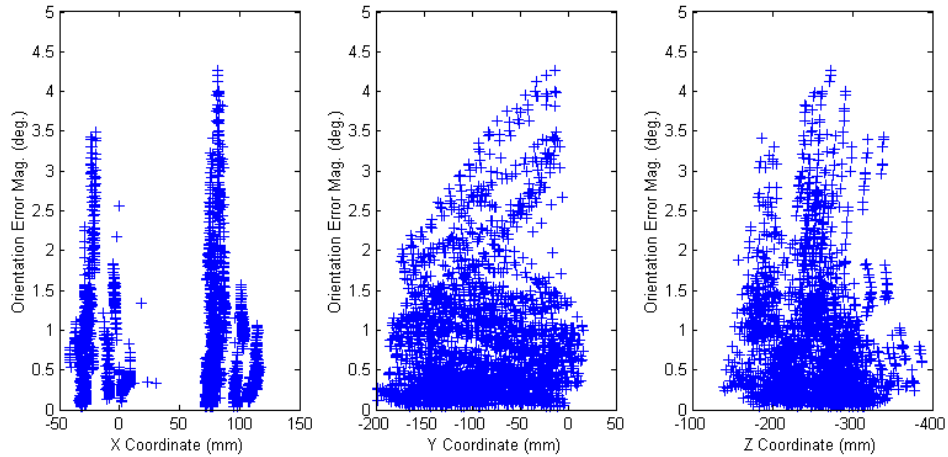
Error magnitude all orientations measured with hybrid robotic system in the Lab  
**Figure 2-30**

In Figure 2-31 we show the trends in error when the magnitude is plotted against each of the three Cartesian components (X,Y,Z) of the sensor position in FG coordinates. The trends are clearer later when we present the data for the Stand-alone fixture because there we have a more even distribution of points in a larger working volume. However, the trends here seem to match those shown later fairly well. If comparing trends, note that the measurement volume (and x-axis scale) are not the same as later with the SA fixture.

*Position Error Distribution*



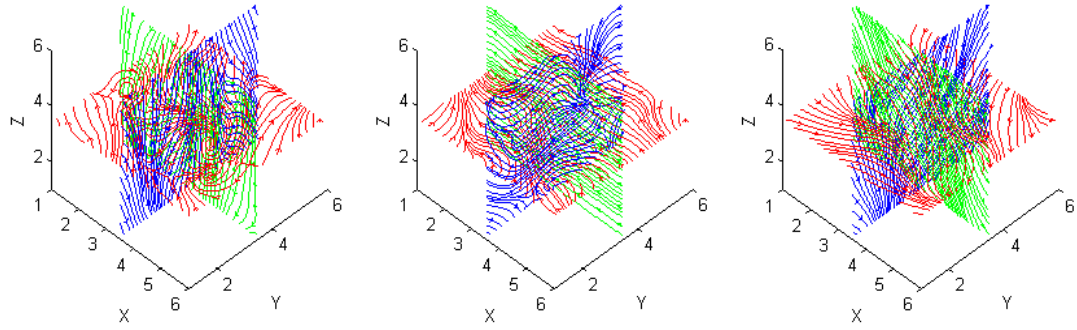
*Orientation Error Distribution*



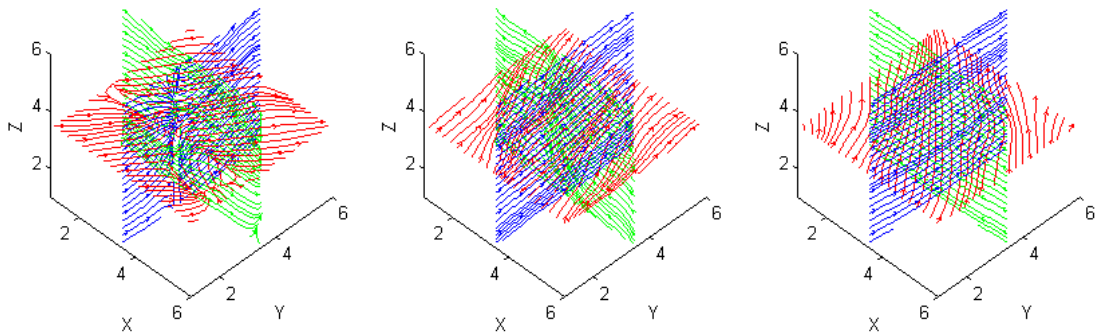
Distribution of error magnitude by X,Y,Z component of the sensor position  
(As measured in the Lab with minimal distortion using the hybrid tracked robotic system)

**Figure 2-31**

The following figures (Figure 2-32 and Figure 2-33) show the curvature of the error field for this natural environment in the lab as measured by the hybrid OTS-EMTS tracked system. This is the relatively undisturbed environment obtained in the lab when there is as little distortion as possible. These are three representative orientations out of the total of 12 measured.

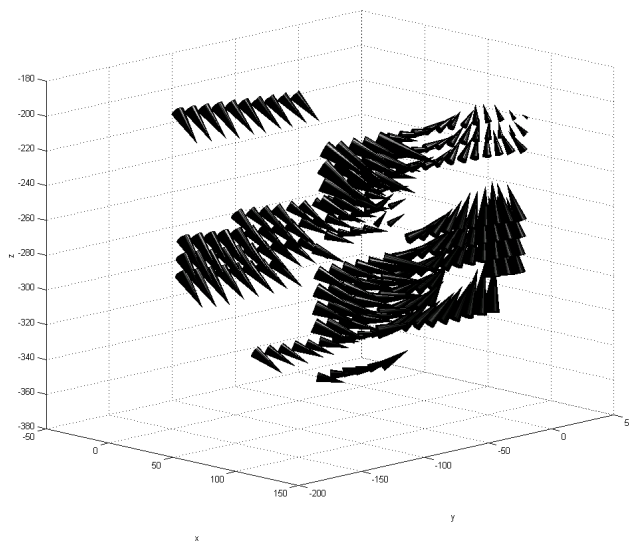


**Position** error field in undisturbed environment for three orientations (2,11,12)  
**Figure 2-32**



**Orientation** error field in undisturbed environment for three orientations (2,11,12)  
**Figure 2-33**

Figure 2-34 shows a representative plot of the position errors for a given orientation. This plot shows 216 points from the full set. The cones directions represent the direction of the error; their lengths represent the relative magnitudes.



Representative plot of position error distortion map for a given sensor (Orientation 9)  
**Figure 2-34**

## Stand-alone Fixture Mapping (Lab)

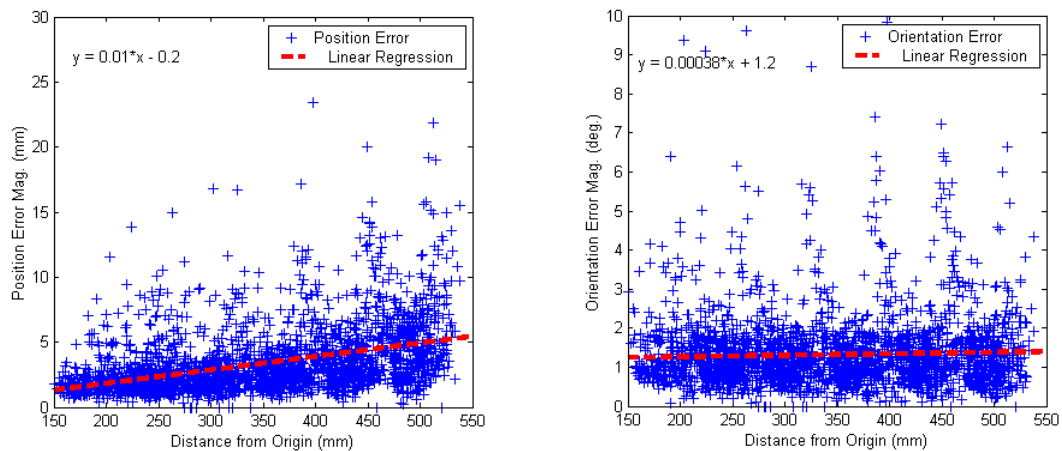
The Stand-alone (SA) fixture allows for consistent collection of data sets just about anywhere. Each data set contains measurements of exactly (within the repeatability of the unit) the same set of sensor positions and orientations. The device collects 216 evenly spaced measurements in a 32x32x20cm grid for each of 12 constant orientations; this is a total of 2592 data points collected per trial.

The first set of results presented is representative data set from one of three trials in nearly identical configurations in our lab (NEB B26) with Aurora FG 100cm above the floor. These measurements are **not** intentionally distorted, however due to the environment around the Aurora the magnetic field still must endure some error-causing distortion. This is not an undisturbed ideal environment, but probably represents a reasonably realistic data collection scenario.

### Error as a function of distance from FG

These following results are from one representative set of data collected with the SA fixture in the lab. They come from Lab Set #3, and this correlates very well with the other two data sets, Lab Set #1 and Lab Set #2, which were collected 5 months earlier. For this data set, 10 of the 2592 total data points were removed as spurious outliers, the remaining 2582 data points remain.

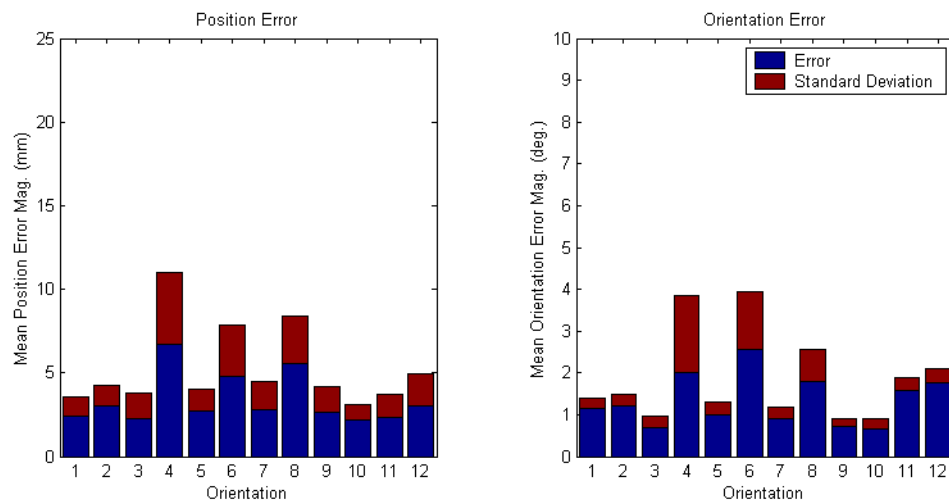
Below in Figure 2-35 is a plot showing the error as a function of distance of the sensor from the FG again. Comparing this to the earlier two data sets for ‘ideal’ and ‘disturbed’ environments, it appears that the slope of the regression lines fall in the middle of the two extremes as would be expected. Their respective equations are overlaid on the plots.



Measurement error plotted vs. distance from field generator in SA fixture in the lab  
**Figure 2-35**

## Error as a function of sensor orientation

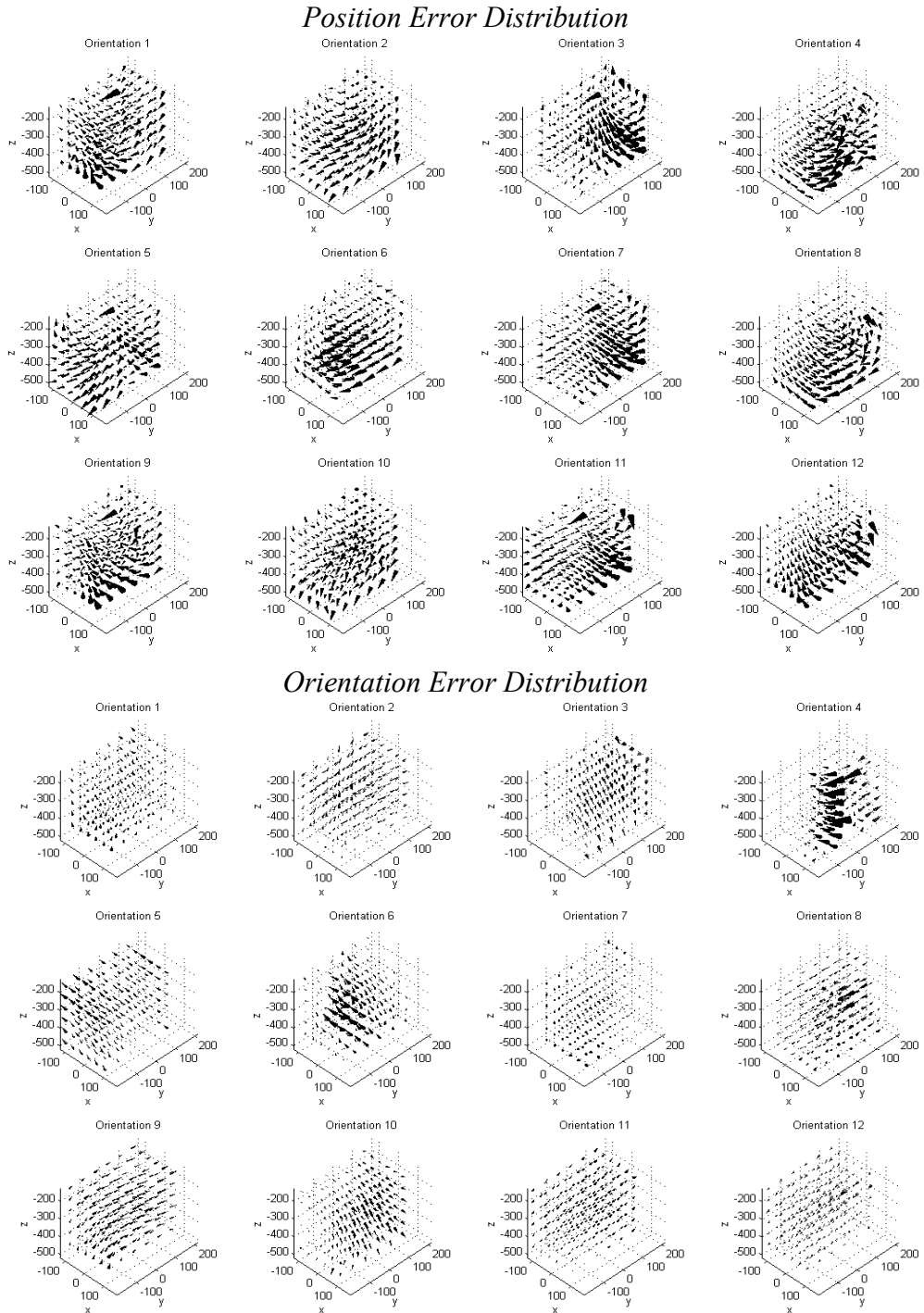
As before, it is important to break the accuracy down by similar orientations. However, the method for grouping the orientations has changed; it is now much more straightforward. There are total of twelve orientations collected, and these orientations remain static throughout the experiment and among all of the experiments; these orientations come from six physical sensors that appear in two orientations each. They are numbered such that *Orientation 1&2* are from *Sensor 1*, *Orientation 3&4* are from *Sensor 2*, and so on. The average error magnitudes for each orientation and their standard deviations are presented in the bar graphs in Figure 2-36 for both position and orientation error.



Error magnitude all orientations measured in SA fixture in the Lab  
**Figure 2-36**

### 3D Visual Representation of Errors (Lab Environment)

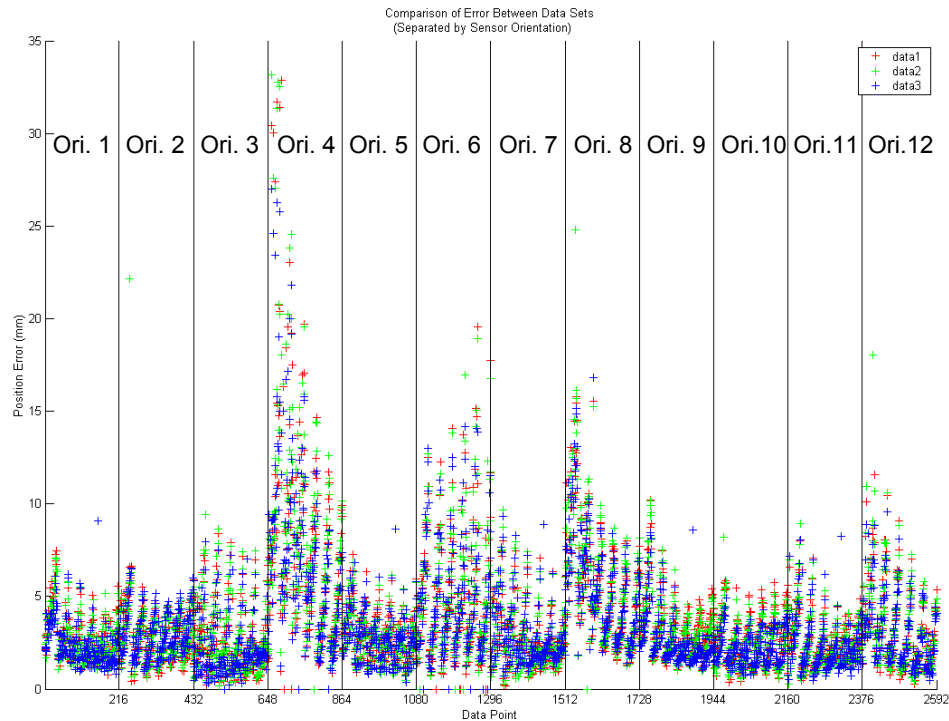
Figure 2-37 displays the distributions of the position and orientation error in the measurement volume. The directions the components of the error; the magnitudes are normalized for each orientation to show the relative error distribution.



Distribution and direction of measurement error separated by orientation  
**Figure 2-37**

## Measurement Consistency

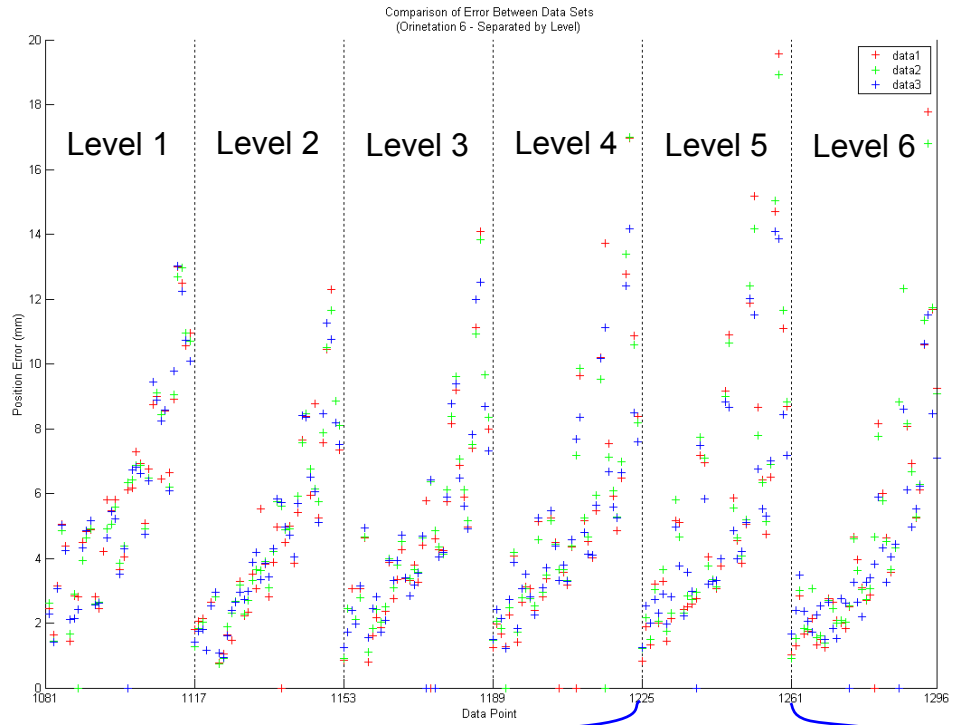
These plots represent the results of three trials with the Stand-alone fixture in nearly identical configurations in the lab. The first figure, Figure 2-38, shows the overall errors for the trials, the vertical lines divide the data between each of the twelve separate orientations. The next figure, Figure 2-39, is a zoomed in version where only a single orientation with 216 data points is presented; here the vertical lines divide the measurement by their respective levels on the calibration fixture. The final figure in this series, Figure 2-40, is zoomed in even further and shows a single level's 36 measurements from a single sensor; here the vertical lines divide the data by their rows on the platform of the fixture. Points further to the right of the plot are getting further from the field generator; points close to the vertical lines are closer to the horizontal edge of the measurement volume. The purpose of these plots is to compare the consistency of the measurement between separate measurement trials of the field. It seems that the measurement are quite repeatable with the exception of a few very large outliers; if a set of measurements do not have three '+' signs in one vertical line, the reported measurement exceeded the maximum value of the y-axis. Measurements that lie exactly on the abscissa ( $y=0$ ) represent null / invalid readings for that particular point. Note that these collections were taken at different times, in slightly different environments, so the results should not be expected to be identical.



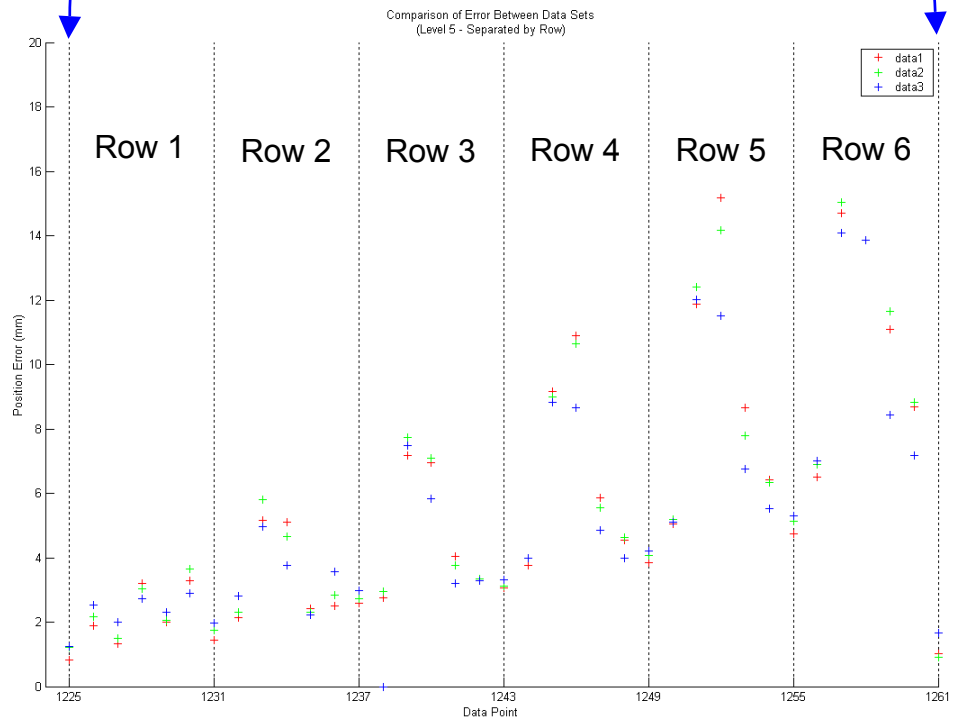
Data collection from three trials using SA fixture in Lab  
(Separated by sensor orientation)

**Figure 2-38**





**Figure 2-39**

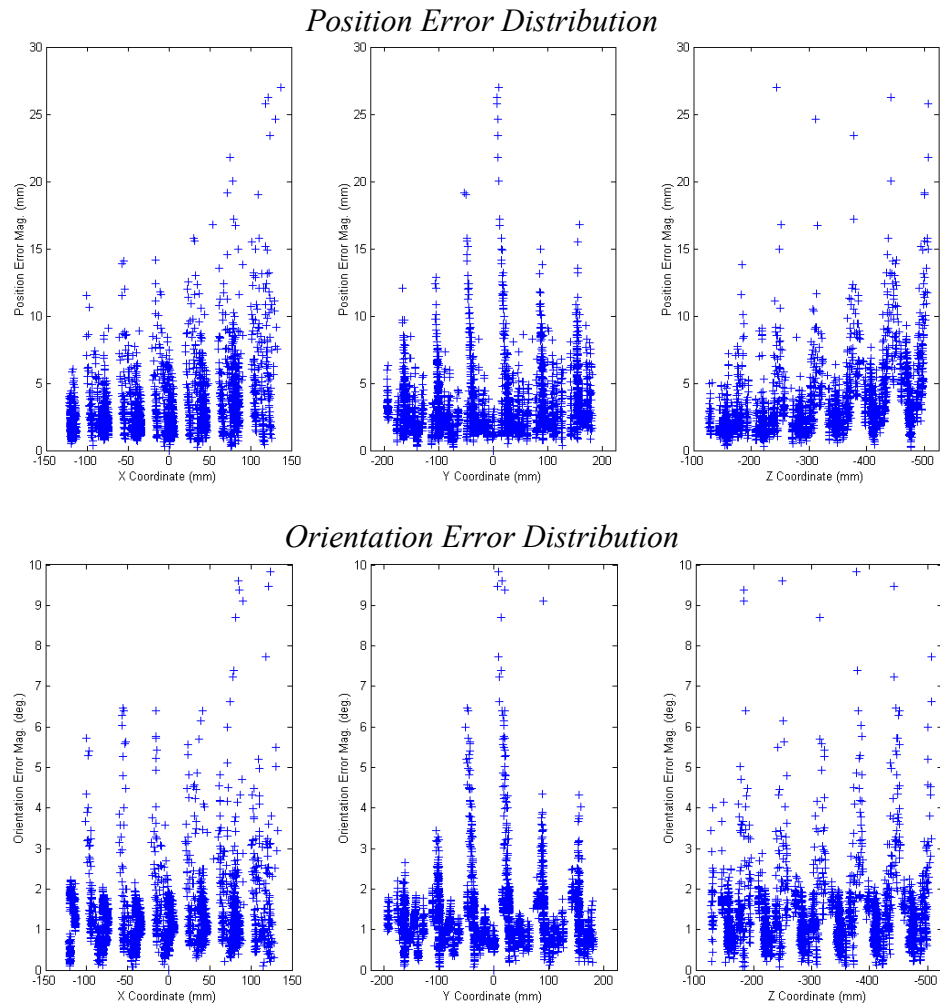


**Figure 2-40**

## Analysis of Error Distribution

Here we study the distribution of error along each of the X,Y,Z components. The following scatter plots in Figure 2-41 show how the position and orientation error vary as a function of each of the components of the sensor position for the Stand-alone fixture in the Lab. This is important for determining the optimal portion of the Aurora's working volume for doing measurements.

By examining the trends, the error seems to increase as the magnitude of Z increases (away from FG), and the error also seems to get worse as X (the height) increases. More interestingly, the error in Y (horizontal component) seems to be at a maximum in the center of the volume. These trends can be compared to those obtained when collecting data in the OR as shown in Figure 2-48. It is clear that the trends are similar, but less pronounced in X&Y, and much more pronounced in Z.



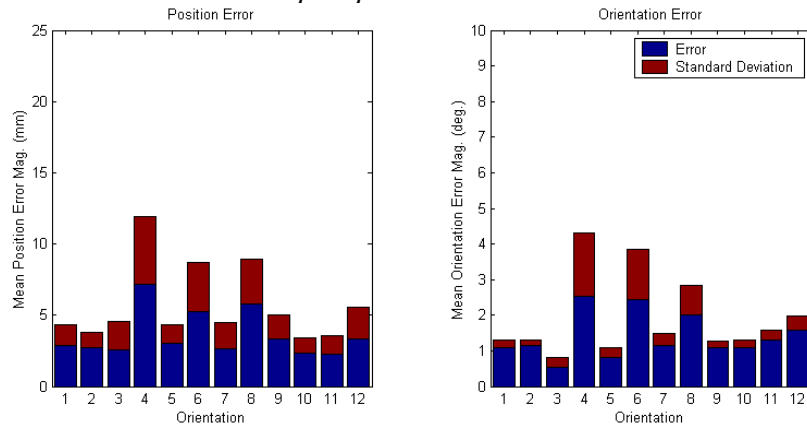
Distribution of error magnitude by X,Y,Z component of the sensor position  
(As measured in the Lab with no additional distortion using the Stand-alone fixture)

**Figure 2-41**

## Stand-alone Fixture Mapping (Disturbed Environment)

Here we show results obtained using the Stand-alone fixture in an artificially disturbed environment. Distorting the field with instruments is tricky to do realistically because the tool must remain still while the SA fixture is moved into all of its configurations. For the first results, shown in Figure 2-42, a stainless steel hip implant was placed between the Aurora FG unit and the measurement volume. Comparing these results to those of the undisturbed lab environment, there is very little difference. If the tool were in the middle of the field, the distortion would probably be much more dramatic. For the second set of results, a 2mm thick, 450x450mm Aluminum sheet was placed horizontally under the data collection region in the SA fixture at a height of 200mm below the Aurora unit. Notice that these results, shown in Figure 2-43, appear to correlate quite well with the highly disturbed NDI Set #3 shown in Figure 2-26 (Note that the orientations do not directly correspond between the two sets).

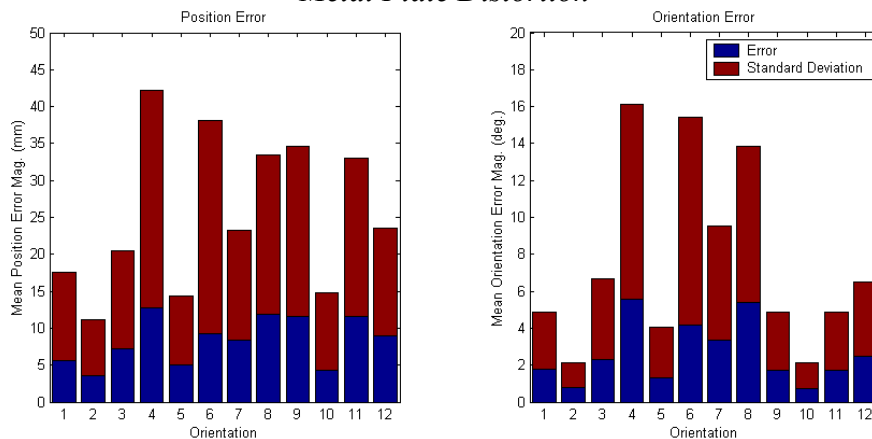
### *Hip Implant Distortion*



Error magnitude for all orientations measured using SA fixture with distortion

**Figure 2-42**

### *Metal Plate Distortion*



Error magnitude for all orientations measured using SA fixture with severe distortion

**Figure 2-43**

## Stand-alone Fixture Mapping (Operating Room)

This data represents the results from data collection in the operating room using the calibrated Stand-alone fixture. This is the same OR used for many ENT procedures at Bayview Medical Center by Dr. Masaru Ishii. The room configuration is similar that of a surgical procedure, except for the lack of instruments, patient, and endoscope controller. This experiment was set up as shown in Figure 2-44. The following results, as expected, show significantly higher errors than those from similar data collections in the lab.

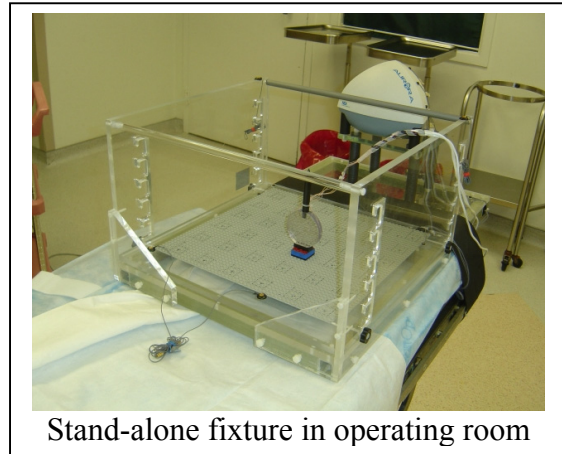
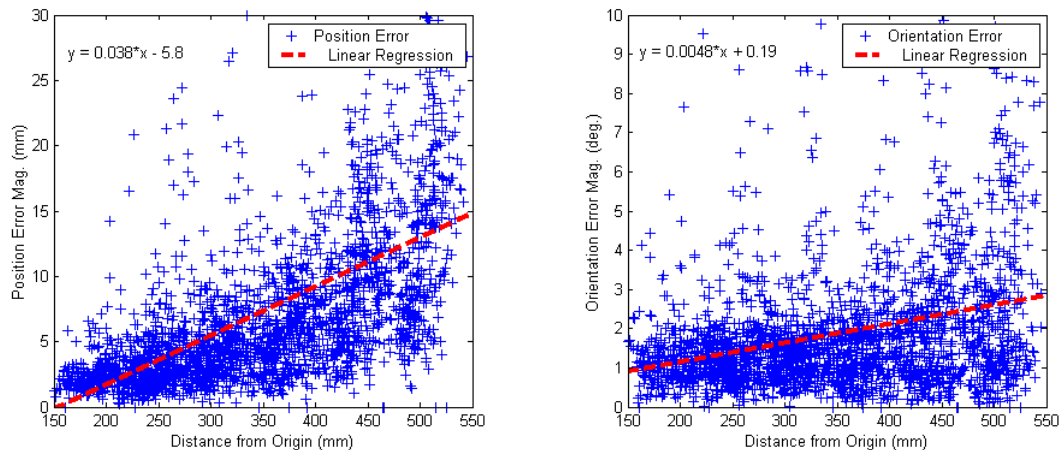


Figure 2-44

### Error as a function of distance from FG

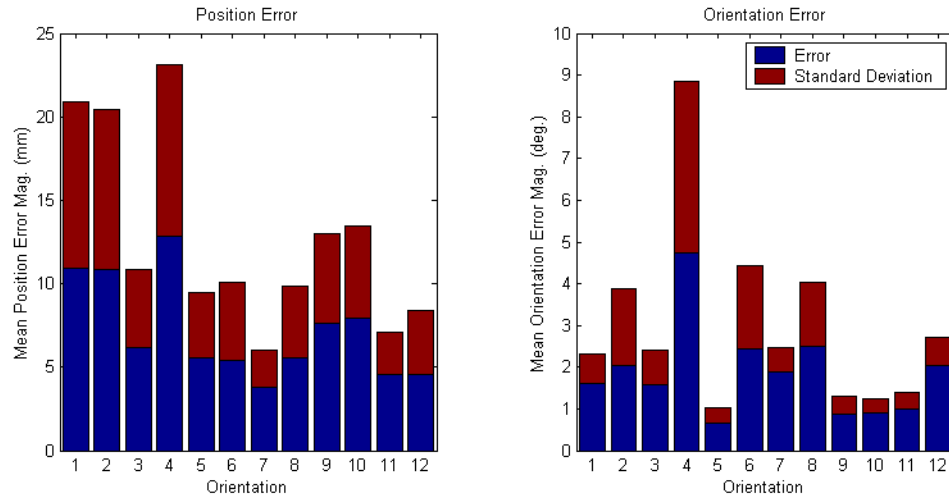
Below in Figure 2-45 is a plot showing the error as a function of distance of the sensor from the FG. The slopes are clearly positive and larger than of the undisturbed data set taken in the Lab. Comparing this to the earlier two data sets for ‘ideal’ and ‘disturbed’ environments, it appears that the slope of the regression lines are about half of that of the ‘worst-case’ disturbed data set. This data set contains the standard 2592 points for the SA fixture less nine points where the Aurora returned null readings.



Measurement error plotted vs. distance from field generator in SA fixture in the OR  
Figure 2-45

## Error as a function of sensor orientation

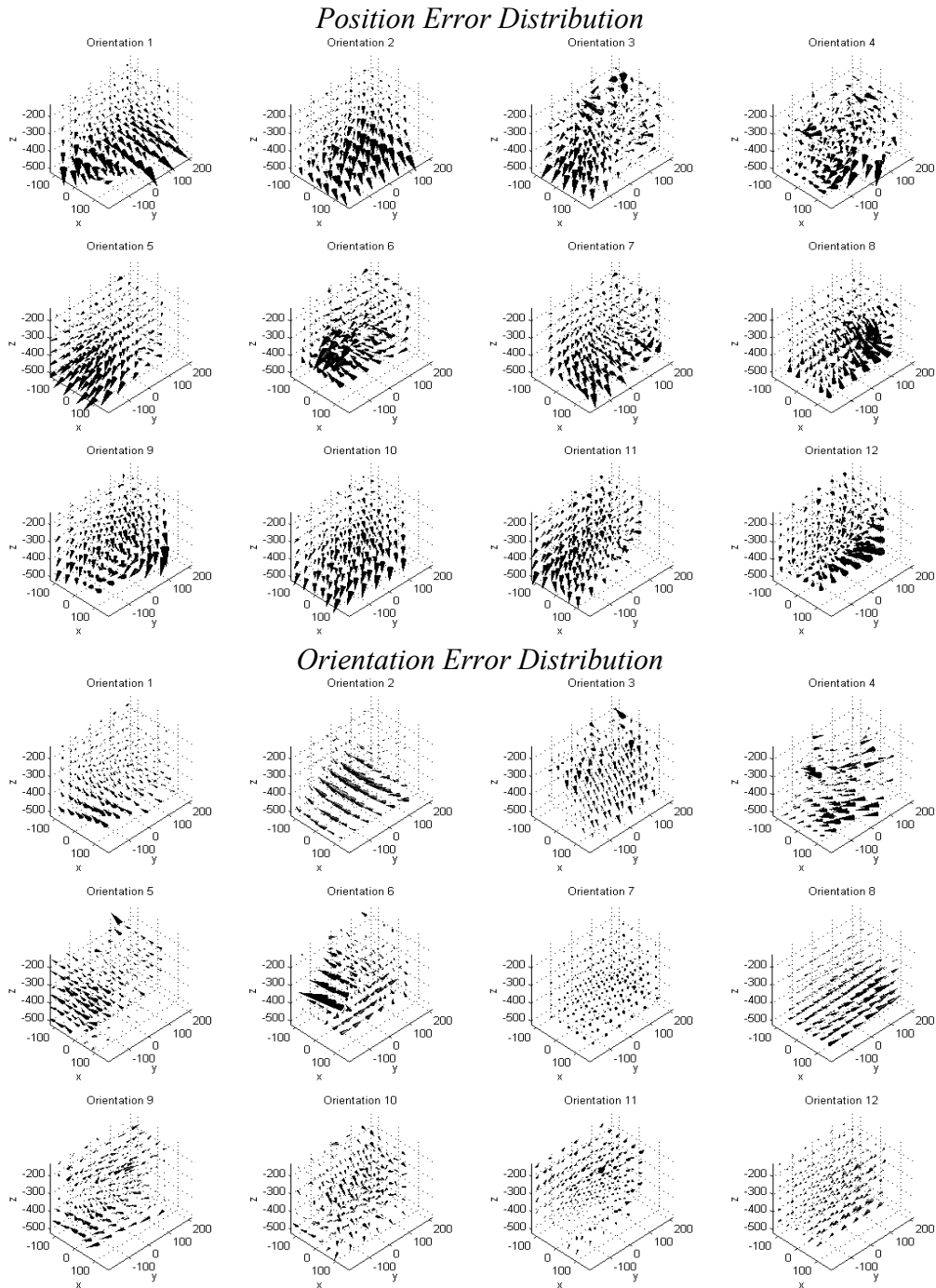
Again, the accuracy is broken down by orientation by the 12 orientations used in the data collection. They are numbered as before for data collection in the undisturbed lab environment. The average errors for each orientation and their standard deviations are presented in the bar graphs in Figure 2-46. Notice how the errors as well as their standard deviations are significantly higher than those taken in the lab. However, they are still somewhat smaller than the errors present in the artificially disturbed ‘worst-case’ data set when compared to Figure 2-25 and Figure 2-26.



Error magnitudes of all orientations measured with SA fixture in **Operating Room**  
**Figure 2-46**

### 3D Visual Representation of Errors (OR Environment)

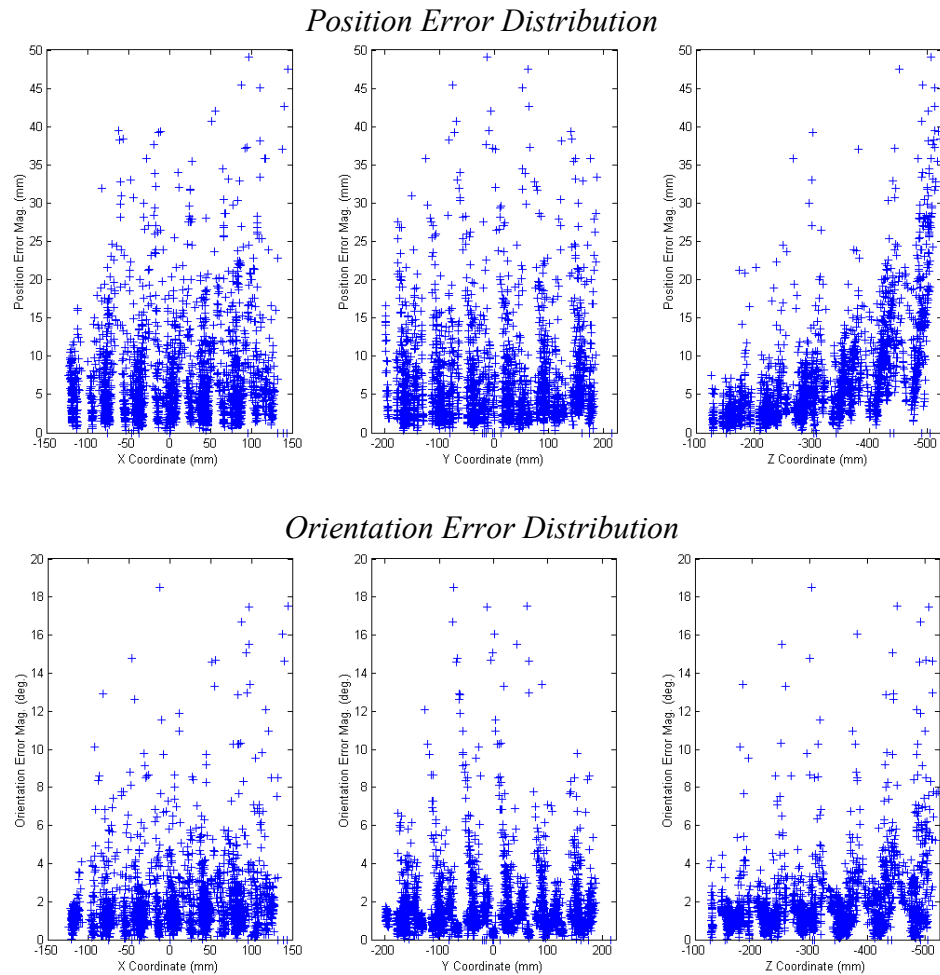
Figure 2-47 displays the distributions of the position and orientation error in the measurement volume. The directions the components of the error; the magnitudes are normalized for each orientation to show the relative error distribution.



Distribution and direction of measurement error separated by orientation  
**Figure 2-47**

## Analysis of Error Distribution

Shown in Figure 2-48 are the errors measured in the operating room as a function of distance along each of the three principal axes of the measurement system. In this distorted environment, the trends in X (height) and Y (horizontal) seem less pronounced than in the undisturbed Lab results shown in Figure 2-41. However, the increase in error as a function of distance along Z (distance away from the FG) is even more pronounced.



Distribution of error magnitude by X,Y,Z component of the sensor position  
(As measured in the OR with significant field distortion using the Stand-alone fixture)

**Figure 2-48**

## Stand-alone Fixture Mapping (Simulated OR)

In order to quantify the effects of certain elements in an OR during a typical ENT IGS application, we decided to perform a distortion mapping with an endoscope cart in the appropriate location but no other intentional distortion present. From the sinus surgery that I observed, the configuration with the cart beside the patient's head and nearby the Aurora unit as shown in Figure 6-2 seems appropriate. The components of the cart are all active (CRT, Camera, Light source) and the experimental setup is as shown in Figure 2-49. This experiment gives a good idea of the field distortion characteristic of such a cart in the OR, without the other effects of the OR present.

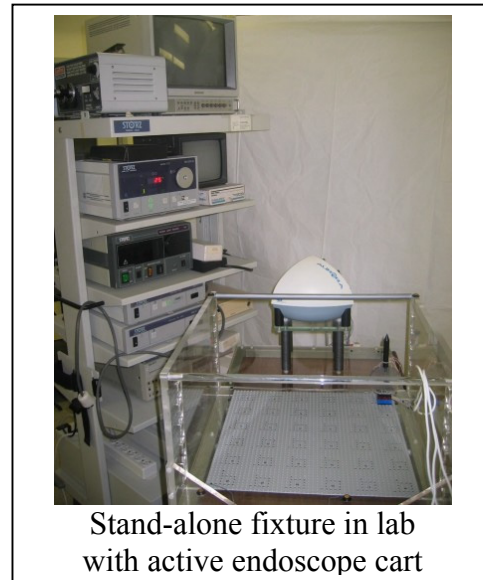
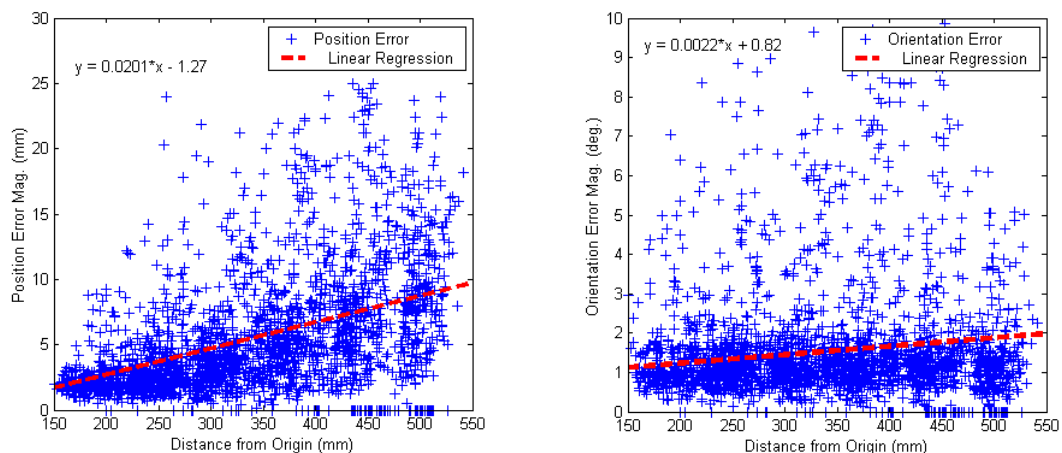


Figure 2-49

### Error as a function of distance from FG

Below in Figure 2-50 is a plot showing the error as a function of distance of the sensor from the FG. The slopes are clearly positive and larger than of the undisturbed data set taken in the Lab. Comparing this to the earlier two data sets for 'ideal' and 'disturbed' environments, it appears that the slope of the regression lines are about half of that of the 'worst-case' disturbed data set.



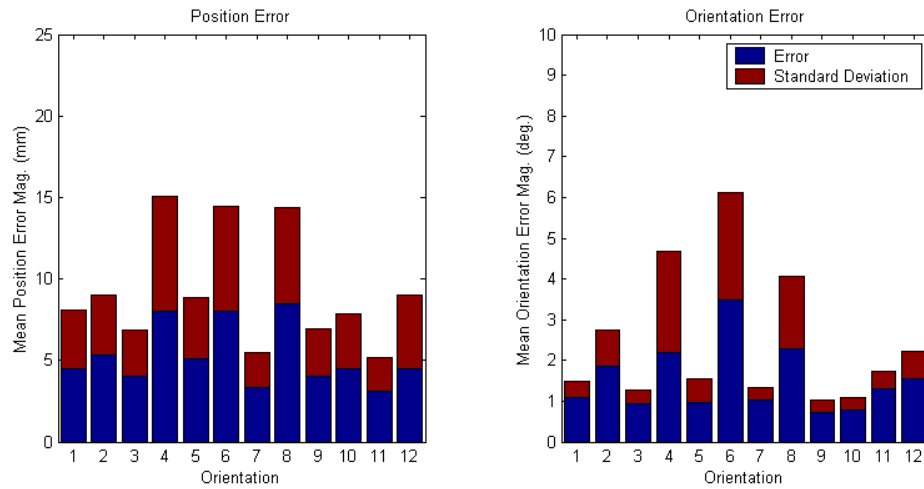
Measurement error plotted vs. distance from field generator in SA fixture in simulated OR scenario with endoscope cart nearby and active

Figure 2-50



## Error as a function of sensor orientation

Tracking accuracy is broken down by orientation into the twelve orientations used in the data collection. They are numbered as before for the data collection in the undisturbed lab environment. The average errors for each orientation and their standard deviations are presented in the bar graphs in Figure 2-51. As would be expected, note that the error here is higher than that of the measurements in the lab with no additional disturbances, but is still not as large as that of the actual OR; this is because in that environment, the OR table dominates the distortion.



Error magnitude for all orientations measured using SA fixture in **Simulated OR** environment with endoscope equipment

**Figure 2-51**

## ***Analysis of Distortion***

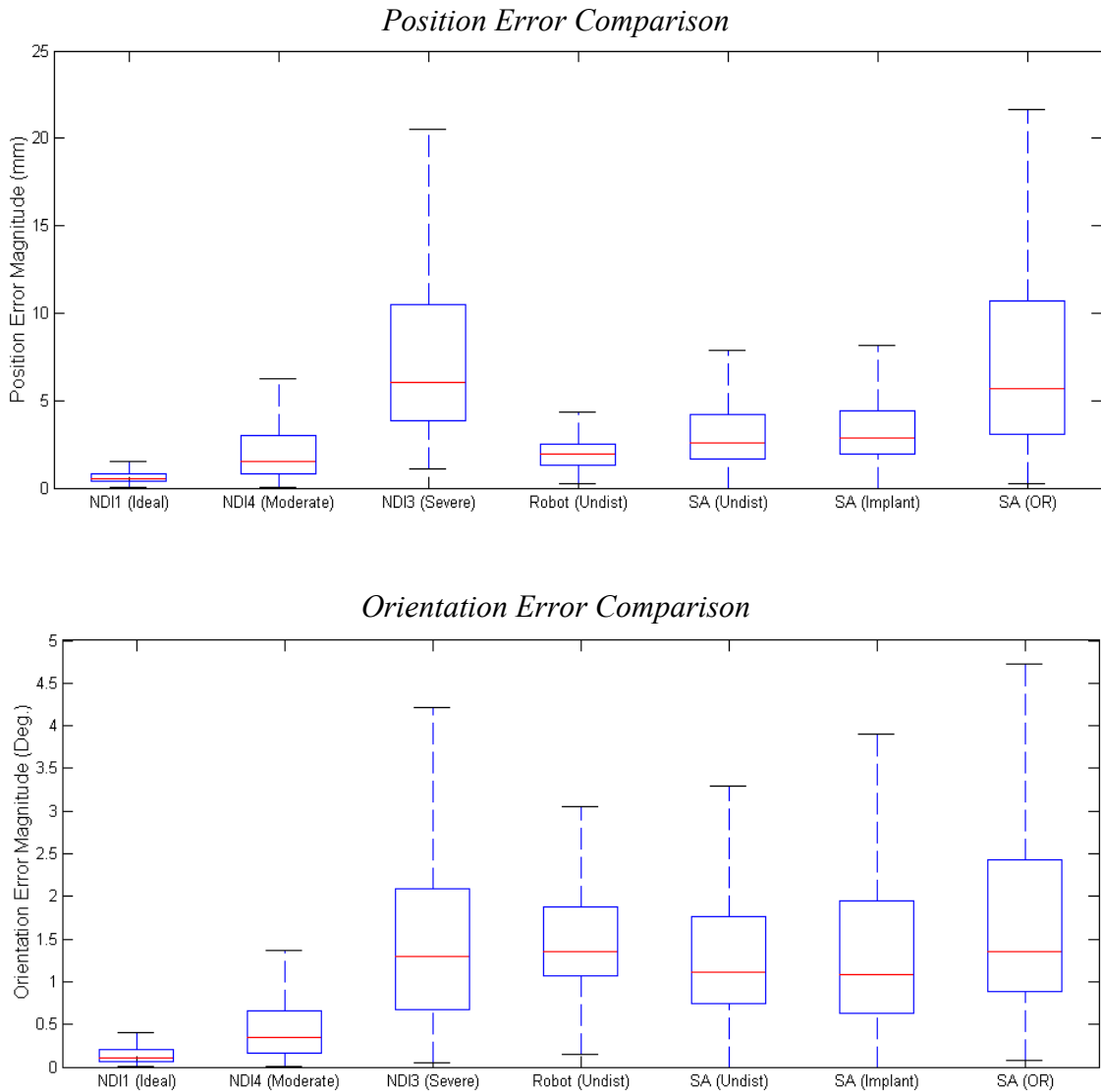
### **Comparisons of Environments**

This section compares the distortions measured with the Aurora system in various environments so that we can get a feel for what kind of errors can be reasonably expected to be present in a given environment. The environments compared here are listed in Table 2-1. The first three data sets were provided by NDI Europe; they represent ‘ideal’ undisturbed, reasonably distorted, and ‘worst-case’ extremely distorted environments respectively. The ‘ideal’ and ‘worst-case’ are the same as presented earlier in this chapter. Next is the results obtained using the hybrid EMTS-OTS tracked robotic calibration with no intentional field distortions present in the lab. The last three data sets were collected with the Stand-alone fixture. The first two were gathered in the lab; one being an environment free of intentional disturbances, and the other having a stainless steel hip implant placed near the measurement volume. The third SA data set was collected in the ENT operating room. For all of the data sets here, 1000 random points from each set were used. See Table 2-1 for a detailed look at the statistics for each of these environments.

	<b><i>Position Error (mm)</i></b>			<b><i>Orientation Error (deg.)</i></b>		
	<b>Mean</b>	<b>Std. Dev.</b>	<b>RMS</b>	<b>Mean</b>	<b>Std. Dev.</b>	<b>RMS</b>
NDI1 (Ideal)	0.684	0.563	0.886	0.182	0.246	0.036
NDI4 (Moderate)	2.285	2.154	3.139	0.461	0.389	0.603
NDI3 (Severe)	8.090	6.350	10.282	1.649	1.597	2.295
Robot (Undist)	2.013	0.958	2.229	1.611	0.765	1.783
SA (Undist)	3.452	2.868	4.487	1.522	1.296	1.999
SA (Implant)	3.748	3.118	4.875	1.512	1.372	2.041
SA (OR)	8.792	10.149	13.424	2.386	3.531	4.260
Comparison of error statistics for measurements from various environments						

**Table 2-1**

The following figure provides a graphical interpretation of the data in Table 2-1 by using box plots to represent the distribution of the magnitudes of the errors. The top portion of Figure 2-52 gives us information about position error distribution. Similarly, the bottom gives information about the orientation error distribution. These results seem to follow the trends that we would expect, with the median and variance of error increasing with the NDI data sets and for the Stand-alone sets as the additional intentional distortion is increased.



Comparison of error distribution for various environments and collection techniques  
 (Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)  
**Figure 2-52**

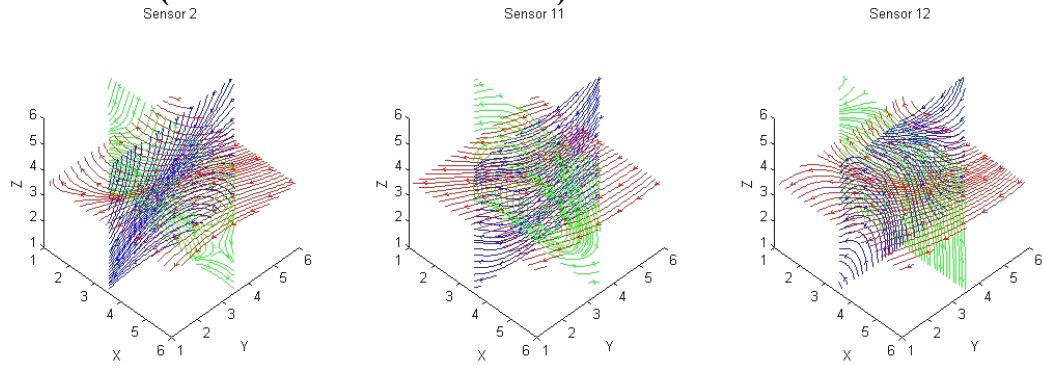
## Curvature of Field Distortion

Magnitude of error is not the only important issue to be concerned with; the rate of change of error in orientation and position space is also crucial. This is because the registration accuracy of a tool frame being fit to a set of sensors is dependent upon the relative position and orientations of those sensors. A way to more realistically predict the relative errors between two sensors may be to base it on the error field curvature, not just the global error bounds. In addition, the transformation between a tool and a patient-fixed reference frame may be a function of field curvature.

The following plots represent slices through the vector field representing the position and orientation error as a function of position in the field. Three distinct sensors (i.e. distinct orientations) are presented for each set. The error is shown for comparison between the curvature of the field in a relatively undisturbed scenario in the lab, a disturbed scenario in the lab, and a disturbed scenario in an operating room on top of the OR table. Position is normalized into a 6x6x6 grid to represent the 216 evenly spaced data points collected for each orientation using the stand-alone fixture.

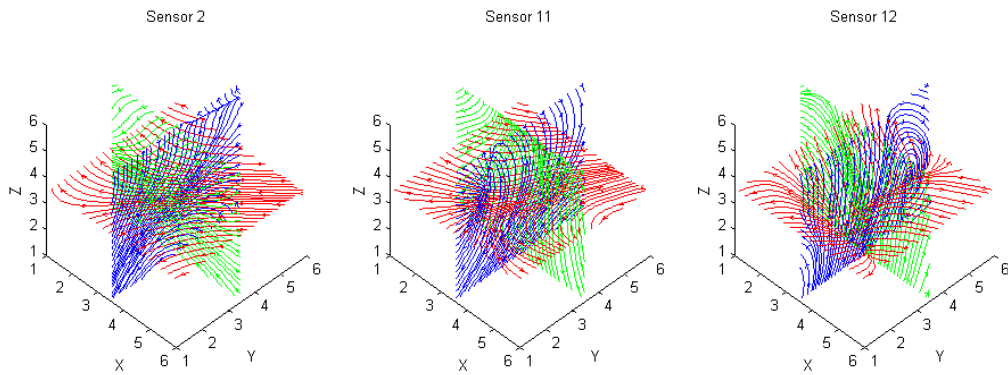
From looking at the figures, it is clear that there is in fact a clear change in the curvature of the distortion field as the disturbance to the field is increased. What you should see in these plots is how the flow fields become more and more warped as moving from Figure 2-53 to Figure 2-54 to Figure 2-55 for position error. Similarly, see how the same happens for orientation error as going from Figure 2-56 to Figure 2-57 to Figure 2-58. However, not that the fields appear much smoother for orientation error, this is because the orientation error does not vary as fast with respect to sensor position as position error does.

**Position Error (Lab with minimal distortion):**



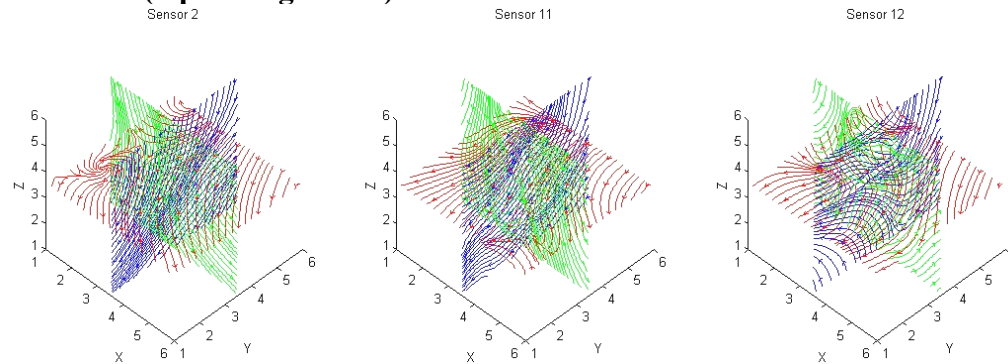
**Figure 2-53**

**Position Error (Lab with Endoscope Cart):**



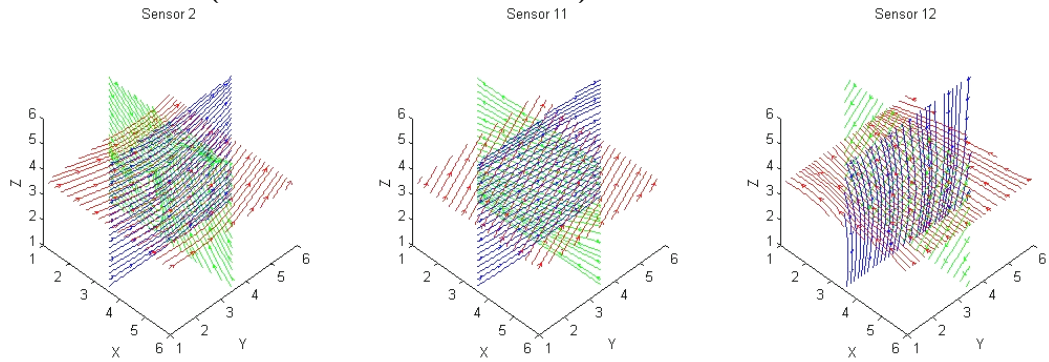
**Figure 2-54**

**Position Error (Operating Room):**



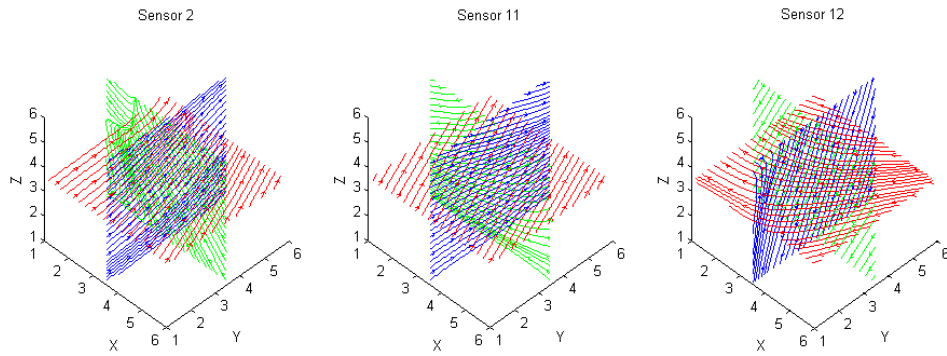
**Figure 2-55**

**Orientation Error (Lab w/ minimal distortion):**



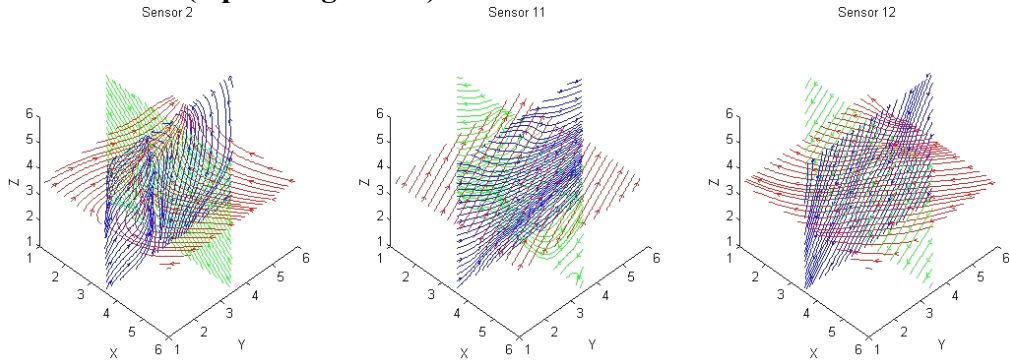
**Figure 2-56**

**Orientation Error (Lab with Endoscope Cart):**



**Figure 2-57**

**Orientation Error (Operating Room):**



**Figure 2-58**

## ***Contributions***

This chapter discussed the methods for, and results of data collection for analyzing the field distortion of an EM tracker. The calibration procedure was based upon that of [Wu, 2003c], but was significantly adapted to allow for data collection using the robot and the Stand-alone fixture. The further contributions were construction of a new calibration wand for robotics data collection, and even more importantly, a digitized Stand-alone calibration fixture that allows data collection in any environment without the need for robotics of external tracking. Using these data collection techniques and hardware, results indicating the levels and shapes of the distortion of various environments were reported. The primary goal of this section was to set up the next chapter where we discuss using this data to generate models of the field distortion.

## **Chapter 3: Error Characterization and Compensation**

### ***Overview***

With a detailed map of the error in the working volume of a tracker, the logical next step is to see how this information can be used to improve measurement accuracy. This section describes the methods for compensating for systematic distortion of the tracking unit's measurements. The calibration framework involves several steps: 1) collecting representative data for the field, 2) determining the error for each of these points, and 3) fitting a polynomial model to this error mapping. In order to do this, a clear definition of the error measure is necessary; the first part describes the error measure.

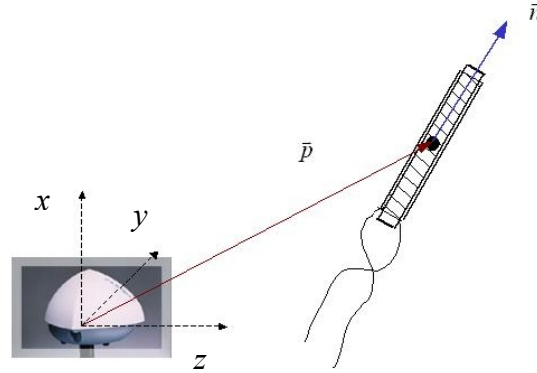
Following error definition is a description of Bernstein polynomials; this is the polynomial basis used for modeling the field distortion error. This description is extended step-by-step to formulate the polynomial model we will use for this application. Finally, the polynomial model is applied to data and the results are presented. This data is used to determine the optimal parameters to use for the model, which take into account the tradeoff between practicality and complexity.



## Methods for Characterizing Measurement Error

### Definition of Error

It is important to have a useful and consistent representation of the sensor error in order to do any analysis. In this first section, the definition of sensor measurement error is defined. The sensor coordinate system is shown here in Figure 3-1.



Sensor measurement configuration  
(Image courtesy of [Wu, 2003a])

**Figure 3-1**

Where:

$\bar{p}$  represents the position of the sensor's origin in the Aurora base frame

$\bar{n}$  represents the unit vector along the axis of the sensor. This is the  $\bar{R}_z$  vector ( $z$  axis) of the sensor orientation in the Aurora base frame

### Position Error

Position error is rather straightforward; the only issue is to maintain the sign convention, i.e. error is additive to the ideal measurement.

Position error is defined as follow:

$$\bar{e}_{pos} = \bar{p}_{measured} - \bar{p}_{ideal}$$

$$\bar{e}_{pos} = \begin{pmatrix} x_{measured} \\ y_{measured} \\ z_{measured} \end{pmatrix} - \begin{pmatrix} x_{reference} \\ y_{reference} \\ z_{reference} \end{pmatrix}$$

## Orientation Error

Orientation error could take on many different meanings. We decided to use the Rodriguez vector to parameterize rotation errors.

Orientation Error is defined as:

$$\bar{e}_{ori} = \bar{\omega}\theta \frac{180}{\pi}$$

Where:  $\bar{\omega} \in \mathbb{R}^3$  is the unit vector pointing along the required axis of rotation to align the reference and measured vectors  $\bar{n}$

$\theta$  is the angle of rotation about  $\bar{\omega}$  between the reference and measured vectors  $\bar{n}$  measured in *radians*

This is derived from:

$$\begin{aligned} \bar{e}_{ori} &= \frac{\bar{n}_{ideal} \times \bar{n}_{measured}}{\|\bar{n}_{ideal} \times \bar{n}_{measured}\|} \arcsin \left\| \bar{n}_{ideal} \times \bar{n}_{measured} \right\| * \frac{180}{\pi} \\ \bar{e}_{ori} &= \frac{\begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix}}{\left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \right\|} \arcsin \left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \right\| * \frac{180}{\pi} \\ \bar{e}_{ori} &= \frac{\begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix}}{\left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \right\|} \left( \theta \frac{180}{\pi} \right) \\ \bar{e}_{ori} &= \bar{\omega}\theta \frac{180}{\pi} \end{aligned}$$

The  $\bar{R}_z$  vector is used to define the sensor orientation because this defines the vector aligned with the axis of rotation of the sensor coils. Since these are 5 DoF sensors, this is enough information to quantify the two rotational DoF. Other parameterizations of the rotation could be used (i.e. Euler angles and quaternions), but for the case of 2 DoF rotational motion, this proves to be a very effective method.

## Bernstein Polynomials

Bernstein polynomials are the basis used in this work for modeling the distortion field. They allow for a global representation of the error, and as with any polynomial, by modifying the order the sensitivity to variation can be adapted. These polynomials have many nice features that make them ideal for this type of application. Two particularly useful ones are 1) by construction the coefficients correspond to the error at the respective control points on the grid, and 2) the derivatives are represented by another Bernstein polynomial that is generated from the differences in adjacent coefficients.

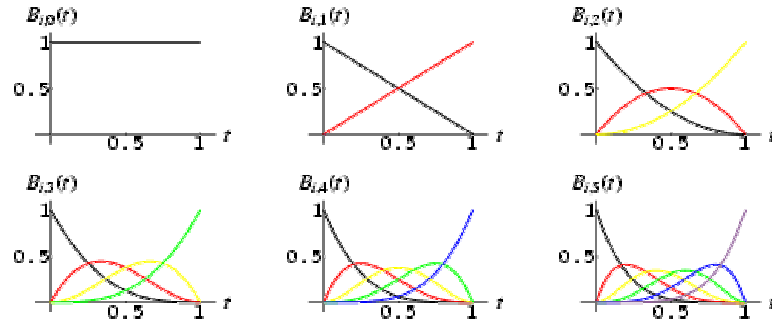
Bernstein polynomials are defined in general by:

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

Where the binomial coefficient is:

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{if } 0 \leq i \leq n \\ 0 & \text{else} \end{cases}$$

Figure 3-2 shows the basis functions of Bernstein polynomials for up to 5<sup>th</sup> order.



Bernstein polynomial basis functions  
(Image courtesy of [Weisstein, 2005])

**Figure 3-2**

To model error with a Bernstein polynomial in a simple 1D case:

$$e(x) = \sum_{i=0}^n c_i B_i^n(x)$$

Where:

$e(x)$  is the error at a position  $x$ , where  $x$  is normalized between 0 and 1

$c_i$  is the  $i^{\text{th}}$  coefficient of the polynomial

$B_i^n(x)$  is the  $i^{\text{th}}$  basis term of an  $n^{\text{th}}$  order polynomial

For the case of a 2<sup>nd</sup> order polynomial ( $n=2$ ), the error is modeled as follows:

$$e(x) = c_0(1-x)^2 + 2c_1(1-x)x + c_2x^2$$

Notice that there is always going to be one more control point/coefficient and basis function than the order of the polynomial. Therefore, we need at least  $n+1$  data points to be able to solve for the coefficients.

Evaluating at the endpoints, two of the three coefficients can be calculated directly if the values are known there:

$$e(0) = c_0$$

$$e(1) = c_2$$

Extending the problem to a 2D space, but still interpolating for a scalar error value:

$$e(x, y) = \sum_{i=0}^n \sum_{j=0}^n c_{i,j} B_i^n(x) B_j^n(y)$$

For a simple 1<sup>st</sup> order example:

$$\begin{aligned} e(x, y) &= \sum_{i=0}^1 \sum_{j=0}^1 c_{i,j} B_i^1(x) B_j^1(y) \\ &= c_{0,0}(1-x)(1-y) + c_{0,1}(1-x)y + c_{1,0}(1-y)x + c_{1,1}xy \end{aligned}$$

Solving at the corners of the unit cube, assuming the values are known, there is sufficient information to solve for all of the coefficients in this case:

$$e(0, 0) = c_{0,0} \quad e(0, 1) = c_{0,1}$$

$$e(1, 0) = c_{1,0} \quad e(1, 1) = c_{1,1}$$

Notice that  $(n+1)^d$  data points are sufficient to solve the problem, where  $d$  is the dimensionality of the space being interpolated in (i.e.  $x, y : d=2$ ). This will always be the minimum number of points necessary to solve for the coefficients for each.

For an arbitrary point:

$$e(x, y) = e(0, 0)(1-x)(1-y) + e(0, 1)(1-x)y + e(1, 0)(1-y)x + e(1, 1)xy$$

Which can be solved for using the four known coefficients,  $e(i, j) = c_{i,j}$ .

For a point directly in the middle, the form is as follows:

$$e\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{4}(c_{0,0} + c_{0,1} + c_{1,0} + c_{1,1}) = \frac{(e(0, 0) + e(0, 1) + e(1, 0) + e(1, 1))}{4}$$

The result, as expected, is the average of the errors at the corners

For convenience, it is useful to write the equations in matrix form:

$$e(x, y) = \sum_{i=0}^n \sum_{j=0}^n c_{i,j} B_i^n(x) B_j^n(y)$$

For the corner points, the matrix equation looks like this:

$$\begin{aligned} \begin{bmatrix} e(0,0) \\ e(0,1) \\ e(1,0) \\ e(1,1) \end{bmatrix} &= \begin{bmatrix} B_0^1(0)B_0^1(0) & B_0^1(0)B_1^1(0) & B_1^1(0)B_0^1(0) & B_1^1(0)B_1^1(0) \\ B_0^1(0)B_0^1(1) & B_0^1(0)B_1^1(1) & B_1^1(0)B_0^1(1) & B_1^1(0)B_1^1(1) \\ B_0^1(1)B_0^1(0) & B_0^1(1)B_1^1(0) & B_1^1(1)B_0^1(0) & B_1^1(1)B_1^1(0) \\ B_0^1(1)B_0^1(1) & B_0^1(1)B_1^1(1) & B_1^1(1)B_0^1(1) & B_1^1(1)B_1^1(1) \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{bmatrix} \end{aligned}$$

In more general terms for an  $n^{\text{th}}$  order polynomial in a 2D space:

$$\begin{bmatrix} e(x, y) \\ \vdots \end{bmatrix}_{i,1} = \begin{bmatrix} B_0^n(x)B_0^n(y) & \cdots & B_n^n(x)B_n^n(y) \\ \vdots & & \vdots \end{bmatrix}_{i,(n+1)^2} \begin{bmatrix} c_{0,0} \\ \vdots \\ c_{n,n} \end{bmatrix}_{(n+1)^2,1}$$

(Matrix dimensions are listed at the bottom right of a matrix for clarity)

Extending from  $\mathbb{R}^2$  to  $l$  dimensional space,  $\mathbb{R}^l$ :

$$\begin{bmatrix} e(x_1, \dots, x_l) \\ \vdots \end{bmatrix}_{i,1} = \begin{bmatrix} B_0^n(x_1) \dots B_0^n(x_l) & \cdots & B_n^n(x_1) \dots B_n^n(x_l) \\ \vdots & & \vdots \end{bmatrix}_{i,(n+1)^l} \begin{bmatrix} c_{0,\dots,0} \\ \vdots \\ c_{n,\dots,n} \end{bmatrix}_{(n+1)^l,1}$$

The previous equation represents a polynomial of degree  $n$  in a space of  $l$  dimensions. However, there is still only a scalar value being interpolated at any point. Below, this is extended to the case where the measure being interpolated has  $m$  dimensions:

$$\begin{bmatrix} e_1(x_1, \dots, x_l) & \cdots & e_m(x_1, \dots, x_l) \\ \vdots & & \vdots \end{bmatrix}_{i,m} = \begin{bmatrix} B_0^n(x_1) \dots B_0^n(x_l) & \cdots & B_n^n(x_1) \dots B_n^n(x_l) \\ \vdots & & \vdots \end{bmatrix}_{i,(n+1)^l} \begin{bmatrix} c_{0,\dots,0}^1 & \cdots & c_{0,\dots,0}^1 \\ \vdots & & \vdots \\ c_{n,\dots,n}^1 & \cdots & c_{n,\dots,n}^1 \end{bmatrix}_{(n+1)^l, m}$$

This framework allows for an  $n$  degree polynomial to be used to interpolate  $m$  measures inside of an  $l$  dimensional space that is normalized such that  $0 \leq x_i \leq 1$   $\forall i \in [0 \dots l]$ .

Note that it is **not** necessary for  $l$ , the dimensionality of the space, and  $m$ , the dimension of the measure being interpolated, to be equal. For example, 3 DoF position error ( $l=3$ ) could be estimated based on a 6 DoF space including position and orientation ( $m=6$ ). Or, as is demonstrated in the following section, 6 DoF measurement error ( $l=6$ ) is estimated in a 3 DoF space including only position ( $m=3$ ). Sensor orientation is accounted for by other means, as will be shown later.

## Position Related Error

Below is the representation used to interpolate six parameters in a 3D space  $\in \mathbb{R}^3$ . This is what is used to determine the three translational and three rotational errors of a sensor for a given 3D location  $(x,y,z)$ .

$$\begin{bmatrix} e_x^1 & e_y^1 & e_z^1 & e_{Rx}^1 & e_{Ry}^1 & e_{Rz}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_x^i & e_y^i & e_z^i & e_{Rx}^i & e_{Ry}^i & e_{Rz}^i \end{bmatrix}_{i,6} = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3} * \begin{bmatrix} c_{x_{0,0,0}} & c_{y_{0,0,0}} & c_{z_{0,0,0}} & c_{Rx_{0,0,0}} & c_{Ry_{0,0,0}} & c_{Rz_{0,0,0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{x_{n,n,n}} & c_{y_{n,n,n}} & c_{z_{n,n,n}} & c_{Rx_{n,n,n}} & c_{Ry_{n,n,n}} & c_{Rz_{n,n,n}} \end{bmatrix}_{(n+1)^3,6}$$

To solve for the  $6*(n+1)^3$  coefficients, a minimum of  $(n+1)^3$  data points are required since each data point has an element that contributes to each of the six polynomial elements.

For clarity and simplification of solving, the problem can be separated into separate equations for each of the dimensions being interpolated, giving six separate problems.

$$\begin{bmatrix} e_x^1 \\ \vdots \\ e_x^i \end{bmatrix} = \beta \begin{bmatrix} c_{x_{0,0,0}} \\ \vdots \\ c_{x_{n,n,n}} \end{bmatrix}, \dots, \begin{bmatrix} e_{Rz}^1 \\ \vdots \\ e_{Rz}^i \end{bmatrix} = \beta \begin{bmatrix} c_{Rz_{0,0,0}} \\ \vdots \\ c_{Rz_{n,n,n}} \end{bmatrix}$$

Where:

$$\beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3}$$

(This is constant for each of the 6 problems, so it need not be recalculated)

These sets of equations are of the form ' $Ax=b$ ', with  $x$  being the coefficients  $c$ ,  $A$  being the Bernstein polynomial basis terms  $B_i^n$ , and  $b$  being the measured error  $e^i$  of the  $i^{\text{th}}$  data point. The coefficients can easily be solved for in six separate least squares problems as long as  $i \geq (n+1)^3$ .

## Solution to Least Squares Problem

It is desired to find the values for the coefficients,  $c$ , in this problem:

$$\begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3} \begin{bmatrix} c_{x_{0,0,0}} & c_{y_{0,0,0}} & c_{z_{0,0,0}} & c_{Rx_{0,0,0}} & c_{Ry_{0,0,0}} & c_{Rz_{0,0,0}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{x_{n,n,n}} & c_{y_{n,n,n}} & c_{z_{n,n,n}} & c_{Rx_{n,n,n}} & c_{Ry_{n,n,n}} & c_{Rz_{n,n,n}} \end{bmatrix}_{(n+1)^3,6} \\ \cong \begin{bmatrix} e_x^1 & e_y^1 & e_z^1 & e_{Rx}^1 & e_{Ry}^1 & e_{Rz}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_x^i & e_y^i & e_z^i & e_{Rx}^i & e_{Ry}^i & e_{Rz}^i \end{bmatrix}_{i,6}$$

Writing out for one dimension being interpolated along in a 3D space:

$$\beta \begin{bmatrix} c_{x_{0,0,0}} \\ \vdots \\ c_{x_{n,n,n}} \end{bmatrix} \cong \begin{bmatrix} e_x^1 \\ \vdots \\ e_x^i \end{bmatrix}$$

Where:

$$\beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3}$$

This equation is in the traditional form of  $A\bar{x} \cong \bar{b}$ . The least squares (LS) solution to this problem for the unknown coefficients,  $c$ , can be solved using singular value decomposition (SVD).

The SVD of  $A \triangleq \beta$  is:

$$A_{i,(n+1)^3} = U_{i,i} \begin{bmatrix} S_{(n+1)^3,(n+1)^3} \\ 0_{i-(n+1)^3,(n+1)^3} \end{bmatrix}_{i,(n+1)^3} V^T_{(n+1)^3,(n+1)^3}$$

(The matrix dimensions are added for clarity)

The  $A\bar{x} \cong \bar{b}$  problem can be re-written as:

$$\begin{bmatrix} S \\ 0 \end{bmatrix} p \cong g \quad \text{Where: } g = U^T b$$

The solution for  $c \triangleq x$  is:

$$x = Vp$$



## Sensitivity Analysis

It is important to know how changes in the values of the collected data affect the resulting polynomial coefficients. This will aid in choosing the proper number and distribution of data points, as well as the optimal order of the polynomial. A similar analysis can be found in [Deif, 1986].

Rewriting the  $A\bar{x} = \bar{b}$  problem with perturbations included:

$$(A + \Delta A)(\bar{x} + \Delta\bar{x}) = (\bar{b} + \Delta\bar{b})$$

Multiplying out the LHS:

$$A\bar{x} + \Delta A\bar{x} + A\Delta\bar{x} + \Delta A\Delta\bar{x} = \bar{b} + \Delta\bar{b}$$

Canceling out the  $A\bar{x} = \bar{b}$  terms:

$$\Delta A\bar{x} + A\Delta\bar{x} + \Delta A\Delta\bar{x} = \Delta\bar{b}$$

Assuming that  $\Delta A\Delta\bar{x} \approx 0$ :

$$\Delta A\bar{x} + A\Delta\bar{x} \approx \Delta\bar{b}$$

Solving for the error in the coefficients:

$$\Delta\bar{x} \approx A^{-1}(-\Delta A\bar{x} + \Delta\bar{b})$$

Taking the norm of both sides:

$$\|\Delta\bar{x}\| \approx \|A^{-1}(-\Delta A\bar{x} + \Delta\bar{b})\|$$

For any consistent matrix and vector norms, we get the following bound:

$$\|\Delta\bar{x}\| \leq \|A^{-1}\| \|\Delta A\bar{x} + \Delta\bar{b}\|$$

The actual definition of the norms used here are described below.

Using the Triangle Inequality:

$$\|\Delta\bar{x}\| \leq \|A^{-1}\| (\|\Delta A\bar{x}\| + \|\Delta\bar{b}\|)$$

Again using the property of consistent norms:

$$\|\Delta\bar{x}\| \leq \|A^{-1}\| (\|\Delta A\| \|\bar{x}\| + \|\Delta\bar{b}\|)$$

Getting the ratio of the error in the coefficients relative to the magnitude:

$$\frac{\|\Delta\bar{x}\|}{\|\bar{x}\|} \leq \|A^{-1}\| \left( \|\Delta A\| + \frac{\|\Delta\bar{b}\|}{\|\bar{x}\|} \right)$$

Now we want to get rid of the  $\|\bar{x}\|$  on the RHS using the norm of the original  $A\bar{x} = \bar{b}$  :

$$A\bar{x} = \bar{b} \Rightarrow \|A\| \cdot \|\bar{x}\| \geq \|\bar{b}\| \Rightarrow \|\bar{x}\| \geq \frac{\|\bar{b}\|}{\|A\|}$$

Plugging this into the previous equation:

$$\frac{\|\Delta\bar{x}\|}{\|\bar{x}\|} \leq \|A^{-1}\| \left( \|\Delta A\| + \frac{\|\Delta\bar{b}\|}{\frac{\|\bar{b}\|}{\|A\|}} \right)$$

Pulling out  $\|A\|$  we can get a more useful expression:

$$\frac{\|\Delta\bar{x}\|}{\|\bar{x}\|} \leq \|A^{-1}\| \cdot \|A\| \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta\bar{b}\|}{\|\bar{b}\|} \right)$$

We can now make the substitution for the condition number of the matrix  $A$ :

$$\frac{\|\Delta\bar{x}\|}{\|\bar{x}\|} \leq \kappa(A) \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta\bar{b}\|}{\|\bar{b}\|} \right)$$

Where:  $(\|A^{-1}\| \cdot \|A\|) = \kappa(A)$  is the condition number of the  $A$  matrix

For this problem,  $A$  represents the Bernstein basis values,  $\beta$ . There is no perturbation present in this matrix. So,  $A$  and  $\Delta A$  are as follows:

$$A = \beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3}$$

$$\Delta A = 0$$

In terms of our problem at hand, the equations simplify to:

$$\boxed{\frac{\|\Delta \bar{x}\|}{\|\bar{x}\|} \leq \kappa(\beta) \left( \frac{\|\Delta \bar{b}\|}{\|\bar{b}\|} \right)}$$

For the case of a unique solution, when  $i = (n+1)^3$ :

$$\kappa(A) = \|A\| \|A^{-1}\|$$

For the case of an over constrained solution that is solved using LS SVD:

$$\kappa(A) = \|A\| \|A^+\|$$

Where:  $A^+$  is the Moore-Penrose matrix pseudo inverse

Using the SVD of  $A$ , this matrix is described by:

$$A^+ = V \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^*$$

When solving this equation, the norms of the  $x$  and  $b$  vectors, and their associated errors, is straightforward; this can be the standard Euclidean norm ( $L^2$ -norm). In order for the above relationships to hold, then the matrix norm used must be consistent with the vector norm. Therefore, the matrix norm described is the  $p$ -norm where  $p=2$ ; this is also known as the spectral norm. The spectral norm is defined as follow.

The matrix norm is described by:

$$\|A\|_2 = \sqrt{\rho(A^*A)}$$

Where:  $A^*$  is the Hermitian of the matrix  $A$  (conjugate transpose)

Since  $A=\beta$  is real, then  $A^*=A^T$

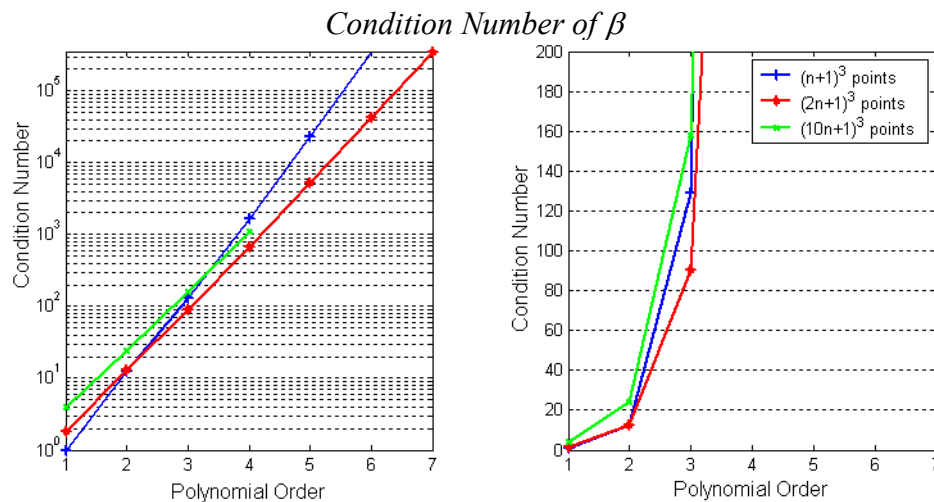
$\rho(A)$  denotes the magnitude of largest eigenvalue of the matrix  $A$

To help test the sensitivity of the coefficients to changes in the errors, we can generate a synthetic matrix of the following form. This matrix is dependent on the points and the polynomial order, but not on the error distribution.

$$A = \beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3}$$

Since we must choose positions corresponding to each row, we can choose the control points. Therefore, we will get a square matrix with evenly distributed data points, e.g. for a 3<sup>rd</sup> order polynomial, we will need the corresponding 4x4x4 grid of points from a unit cube. Figure 3-3 shows the relationship between the matrix's condition number,  $\kappa(\beta) = \|\beta\| \|\beta^{-1}\|$ , and the order of the polynomial. In the following plot, the largest singular value of A is used as the matrix norm. The results are plotted for several numbers of evenly dispersed data points, with  $(n+1)^3$  representing a uniquely determined system where  $\beta$  is square.

For the overdetermined system when there are more data points than necessary (greater than  $(n+1)^3$  points), the pseudo-inverse,  $\beta^+$ , is used to calculate the condition number instead of  $\beta^{-1}$ . As would be expected, as the order gets higher, the model becomes more and more sensitive to noise in the error measurements. Up to a 3<sup>rd</sup> order model, the condition number stays in a good range (~100); after that it gets rather large very fast. This trend is the same regardless of the number of data points used.

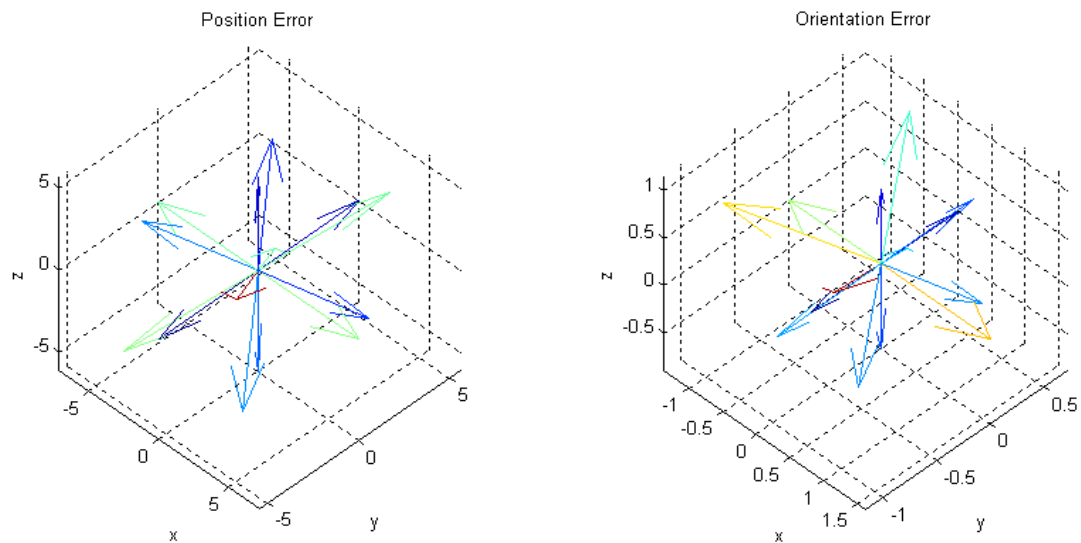


Condition number of  $\beta$  as a function of polynomial order plotted on a semi-log axis (left), and close up (right)

**Figure 3-3**

## Orientation Related Error

Next is the issue of how the sensor orientation plays a role in measurement distortion. It is clear that a sensor's measurement error is not only a function of its position, but also a function of its orientation. Unfortunately, this adds significant complexity to the problem and increases the required number of data points in order to get an adequate mapping of the field. Figure 3-4 shows the relative errors produced by the Aurora system when the field is subjected to a disturbance (NDI Set #3). The data separated into 14 sets such that each includes the data points whose orientations fall within  $20^\circ$  of their respective base vectors. Blue lines represent minimal error and red lines represent the maximum error. It is clear that the error is in fact orientation dependent and that there is not even a direct correlation between opposing orientations.



Base vector distribution with corresponding relative errors (Max = Red, Min = Blue)  
**Figure 3-4**

One method of incorporating orientation-induced errors would be to add three more coefficients to the  $\beta$  matrix from above to get a matrix of the following form:

$$\beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1)B_0^n(Rx_1)B_0^n(Ry_1)B_0^n(Rz_1) & \cdots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1)B_n^n(Rx_1)B_n^n(Ry_1)B_n^n(Rz_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i)B_0^n(Rx_i)B_0^n(Ry_i)B_0^n(Rz_i) & \cdots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i)B_n^n(Rx_i)B_n^n(Ry_i)B_n^n(Rz_i) \end{bmatrix}_{i,(n+1)^6}$$

However, this is a six dimensional problem which is rather unpractical because we must now solve for  $(n+1)^6$  coefficients. For a 3<sup>rd</sup> order polynomial, that translates to an increase from 64 to 4096 data points required at a minimum.

A more practical method, based upon the work presented in [Wu, 2003a], is interpolating the sensor's orientation between a set of base orientations and generating a 3D polynomial for each of these. So, if there are  $b$  distinct base orientations, then the number of coefficients for each interpolated dimension, and therefore the minimum number of data points required, is  $b*(n+1)^3$ . Therefore, rather than an  $O(n^3)$  increase, we have an  $O(n)$  increase in calculations. For the previous 3<sup>rd</sup> order polynomial example, this would mean 896 data points if there are 14 base orientations. For 5<sup>th</sup> order, the difference is even more dramatic between the two methods, 46656 vs. 3024 coefficients.

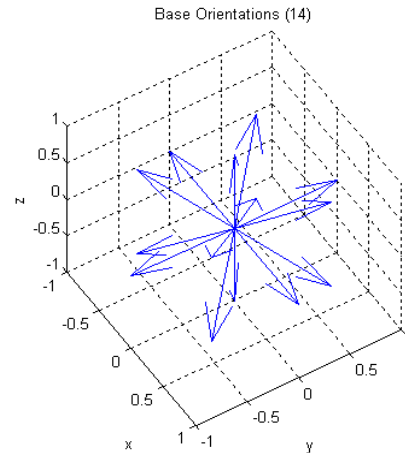
**Algorithm**

- Choose the set of base orientations / basis vectors for which polynomials are generated. They should be evenly distributed as shown in Figure 3-5, but the actual number depends on the particular field. For much of this work, the vectors described in Table 3-1 appear to work quite well.

For 14 base orientations, we have:

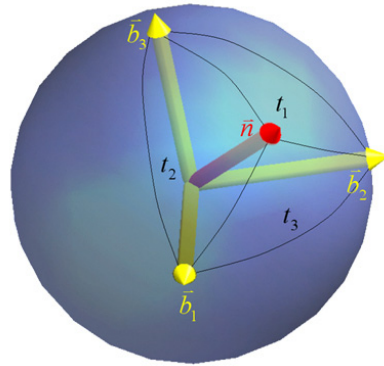
Base vectors:	14		
1	0	0	
0	1	0	
0	0	1	
0.57735	0.57735	0.57735	
-0.57735	0.57735	0.57735	
0.57735	-0.57735	0.57735	
-0.57735	-0.57735	0.57735	
-1	0	0	
0	-1	0	
0	0	-1	
-0.57735	-0.57735	-0.57735	
0.57735	-0.57735	-0.57735	
-0.57735	0.57735	-0.57735	
0.57735	0.57735	-0.57735	

**Table 3-1**



**Figure 3-5**

- For each data point, determine the closest three base orientation vectors that enclose the  $z$ -axis of the measured sensor reading,  $\vec{n}$ , inside of a spherical triangle created by the tips of the unit vectors as shown in Figure 3-6.
- Determine the corresponding area of each of the three spherical triangles generated when subdividing the surface using Barycentric coordinates. The weighting of a particular base vector's contribution to the error is directly proportional to the area of the corresponding, opposing spherical triangle.



- $\bar{n}$  Measured orientation (z axis of sensor)
- $\bar{b}_1, \bar{b}_2, \bar{b}_3$  Enclosing three base orientation vectors
- $t_1, t_2, t_3$  Areas of corresponding spherical triangles

Definition of the spherical triangles used for spherical interpolation

Figure 3-6

- Normalize the areas ( $t_1, t_2, t_3$ ) of each of the three spherical triangles such that they sum to one; these numbers are the three non-zero entries of the weighting coefficient matrix to be used when generating the polynomial for each of the orientations. If an orientation lies directly on a base vector, then it's weight is set to one, and all others are set to zero. This can be useful for data that is collected in a consistent set of orientations.
- The results are put into a matrix describing the orientation weights corresponding to each data point. Table 3-2 contains a printout of the first 25 data points of this matrix for an example data set of 14 base orientation vectors.

Calculating Coefficient matrix: 16437															
0	-0.425351	-0.459193	-0.779888	0	0	0	0	0	0	0.0224916	0.260434	0.717074	0	0	0
1	-0.79435	0.08261	-0.601814	0	0	0	0	0	0	0.51439	0	0.353973	0	0	0.131636
2	-0.273788	-0.882264	-0.382933	0	0	0	0	0	0	0.468667	0.0718821	0.459451	0	0	0
3	0.593382	-0.435309	0.677056	0.107591	0	0.16884	0	0	0	0.72357	0	0	0	0	0
4	-0.085319	0.882936	-0.461695	0	0.608828	0	0	0	0	0	0	0.252237	0	0	0.138934
5	0.782203	-0.605976	-0.144779	0.455952	0	0	0	0	0	0	0	0.312768	0	0	0.23128
6	-0.981157	-0.191926	-0.02212	0	0	0	0	0	0	0.84588	0.11524	0	0.03888	0	0
7	0.115046	-0.993325	0.006586	0.0744028	0	0	0	0	0.0119318	0	0	0.913665	0	0	0
8	-0.254755	0.961597	0.102126	0	0.722149	0	0	0.175882	0	0	0.101969	0	0	0	0
9	-0.964233	-0.029132	0.263456	0	0	0.158075	0	0	0	0.0499485	0.791977	0	0	0	0
10	0.875635	0.476322	-0.080079	0.603997	0.266001	0	0	0	0	0	0	0	0	0	0.130002
11	-0.153444	0.982094	0.103259	0	0.77502	0	0	0.195255	0	0	0.0297243	0	0	0	0
12	0.921548	-0.385331	-0.047534	0.694512	0	0	0	0	0	0	0.226792	0	0	0.0786956	0
13	0.008888	0.996148	-0.087201	0	0.929807	0	0	0	0	0	0	0.0538978	0	0	0.0162956
14	0.16039	-0.974834	0.154854	0.00370294	0	0	0	0.276535	0	0	0.719762	0	0	0	0
15	-0.486063	0.060304	0.87184	0	0.615559	0.097669	0	0.286772	0	0	0	0	0	0	0
16	-0.301799	-0.94996	0.080534	0	0	0	0	0.136554	0.147949	0.715497	0	0	0	0	0
17	-0.14581	-0.932703	0.329839	0	0	0.122084	0	0	0.245823	0	0.632093	0	0	0	0
18	-0.454741	0.875727	0.162192	0	0.54013	0	0	0.265411	0	0.19446	0	0	0	0	0
19	-0.579128	0.572398	-0.580516	0	0	0	0	0	0.00470034	0.00567201	0	0	0	0.989628	0
20	0.660806	0.099302	0.743962	0.386889	0.45547	0.157641	0	0	0	0	0	0	0	0	0
21	-0.902941	0.421685	0.083011	0	0.226544	0	0	0.136342	0	0.637115	0	0	0	0	0
22	-0.968685	0.218639	-0.117729	0	0.0674388	0	0	0	0.72717	0	0	0	0.205391	0	0
23	-0.847689	-0.138239	-0.512173	0	0	0	0	0	0.526205	0.250295	0.2235	0	0	0	0
24	-0.81512	-0.546705	0.191593	0	0	0	0	0.30918	0.453045	0.237775	0	0	0	0	0
25	-0.504326	0.815521	-0.283692	0	0.388661	0	0	0	0	0.146537	0	0	0	0.464802	0

Example data from orientation weighting matrix for 14 base vectors

Table 3-2

- Determine the extents of the sensor positions in the data set to use as bounds. These bounds are used such that all of the measured locations can be scaled into a unit cube for use with Bernstein polynomial approximation. The reason for scaling the data is to improve the numerical stability of the Bernstein approximation.

- Calculate the Bernstein coefficients. For each base orientation, there are six sets of coefficients to solve for:  $X, Y, Z, Rx, Ry, Rz$ .
  - Normalize the data point locations within the bounds
  - Build the set of equations to be used for SVD Least Squares for each basis vector and each dimension to be interpolated:

$$\begin{bmatrix} w_{1,b}e_x^1 \\ \vdots \\ w_{i,b}e_x^i \end{bmatrix} = \beta \begin{bmatrix} w_{1,b}C_{x_{0,0,0}} \\ \vdots \\ w_{i,b}C_{x_{n,n,n}} \end{bmatrix}, \dots, \begin{bmatrix} w_{1,b}e_{Rz}^1 \\ \vdots \\ w_{i,b}e_{Rz}^i \end{bmatrix} = \beta \begin{bmatrix} w_{1,b}C_{Rz_{0,0,0}} \\ \vdots \\ w_{i,b}C_{Rz_{n,n,n}} \end{bmatrix}$$

Where:

$w_{i,b}$  is the weight of the  $i^{\text{th}}$  data point for the  $b^{\text{th}}$  basis vector

$$\beta = \begin{bmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \dots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \dots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{bmatrix}_{i,(n+1)^3}$$

( $\beta$  is constant for each of the  $6b$  least squares problems)

- Solve for the  $6b*(n+1)^3$  coefficients in the Least Squares sense using Singular Value Decomposition



## Measurement Compensation

The following equation allows for compensation along each dimension being interpolated ( $X, Y, Z, Rx, Ry, Rz$ ) in a space  $\in \mathbb{R}^3$ :

$$e(x, y, z) = \sum_{b=1}^3 w_b \left( \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n c_{i,j,k}^b B_i^n(x) B_j^n(y) B_k^n(z) \right)$$

Where:

$b$  is the index of each of the three surrounding base orientations

$c_{i,j,k}^b$  is the corresponding set of Bernstein coefficients for each of the base orientations used

$x, y, z$  are the normalized positions in the unit cube as determined by the bounds of the data used to generate the polynomial.

In vector form, the equation looks like this:

$$\begin{bmatrix} \bar{e}_{pos} \\ \dots \\ \bar{e}_{ori} \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_z \\ e_{Rx} \\ e_{Ry} \\ e_{Rz} \end{bmatrix} = \sum_{b=1}^3 w_b \left( \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \begin{bmatrix} c_{i,j,k_x}^b \\ c_{i,j,k_y}^b \\ c_{i,j,k_z}^b \\ c_{i,j,k_{Rx}}^b \\ c_{i,j,k_{Ry}}^b \\ c_{i,j,k_{Rz}}^b \end{bmatrix} B_i^n(x) B_j^n(y) B_k^n(z) \right)$$

## ***Optimization of Characterization Parameters***

### **Optimizing the Polynomial Approximation**

Using Bernstein polynomials as described above to model the measurement distortion leaves several parameters that must be chosen wisely. Ideally, we would be able to collect an enormous amount of data and fit a high order polynomial model with a large number of base orientations. However, there is a tradeoff between a practical data collection for a given application and its accuracy. Therefore, we are trying to find the optimal solution for the following parameters for a given scenario:

1. Order of the polynomial
2. Number and Distribution of Base Orientations
3. Distribution of collected data points (positions & orientation)

The optimal parameters are not fixed; they depend on the purpose and the conditions that the system will be operating in. For instance, to compensate for systematic distortion of the Aurora unit itself, which is the dominating factor in a low distortion area, it is necessary to use a high order polynomial and a large number of base orientations; therefore, the data collection must be very dense. If we are trying to compensate for errors in a distorted environment, we can generally get away with a lower order polynomial and fewer base orientations since the induced field distortion is the dominating factor and is generally relatively smooth. This allows for fewer data points to generate the polynomial; thus making it much more practical for use in an OR setting. The following will discuss how the optimal parameters are chosen.

The optimal order and number of base vectors can be determined analytically and experimentally. By looking at the curvature of the field (as appears in a high order model), we can determine what lower order polynomial will retain as much accuracy as possible.

To verify the solution, first a high order polynomial with a large number of base vectors is generated from a dense data set (in both position and orientation space). Then the order is decimated and the numbers of base orientations are dropped sequentially. By comparing the results and the trends generated, it is possible to see where the optimal order and number of base vectors is for a given environment. With this information, it is possible to determine the number and distribution of data points necessary to generate such a polynomial. The combinations that are tested are:

Number of Base Orientation Vectors : {6,14,26}

Polynomial Model Order : {1,2,3,4,5,6,7}

## Choosing the Appropriate Parameters

Knowing the effects of adding basis vectors and increasing the order on the results is essential to make a practical choice for these parameters. It is necessary to make a choice that allows for a practically sized data set, especially if the particular application is intra-operative compensation of measurements.

Assuming a perfectly distributed data set, the plot in Figure 3-7 shows the absolute minimum number of data points that would be necessary to generate polynomials of a given order and number of base vectors. By ‘perfectly distributed’, each sensor orientation would have to fall evenly between three base vectors, and each base vector receives three sensors that contribute to it at each of the grid points of a  $(n+1) \times (n+1) \times (n+1)$  grid, where  $n$  is the order of the polynomial.



Representation of the minimum required number of measurements necessary as a function of polynomial order for several base orientation sets

**Figure 3-7**

The **minimum number of data points** is defined by:

$$N_{\min} = \frac{(n+1)^3 * b}{3}$$

Where:

$n$  is the polynomial order

$b$  is the number of base orientations

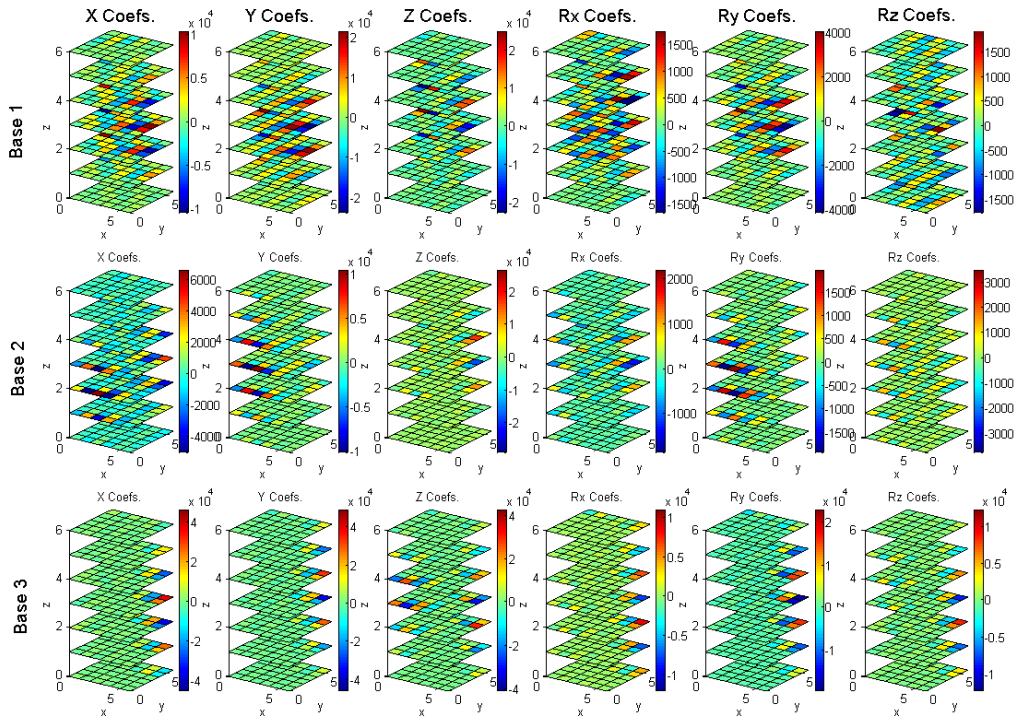
## Curvature of the Distortion Field

The goal here is to gain some insight into how the error field changes with respect to sensor position and orientation. Since this is very difficult to do with the actual measurement data, I will assume that a 6<sup>th</sup> order Bernstein polynomial provides an accurate interpretation of the error field. The following plots in Figure 3-8 show the relative magnitudes of the coefficients for this polynomial. Since we have six sets of coefficients (three for position error and three for orientation error), there are six columns in the plots. The rows represent the data associated with each of three distinct base orientation vectors; the three orthogonal vectors the  $x$ ,  $y$ , and  $z$  axes. The values are normalized into a  $7 \times 7 \times 7$  grid.

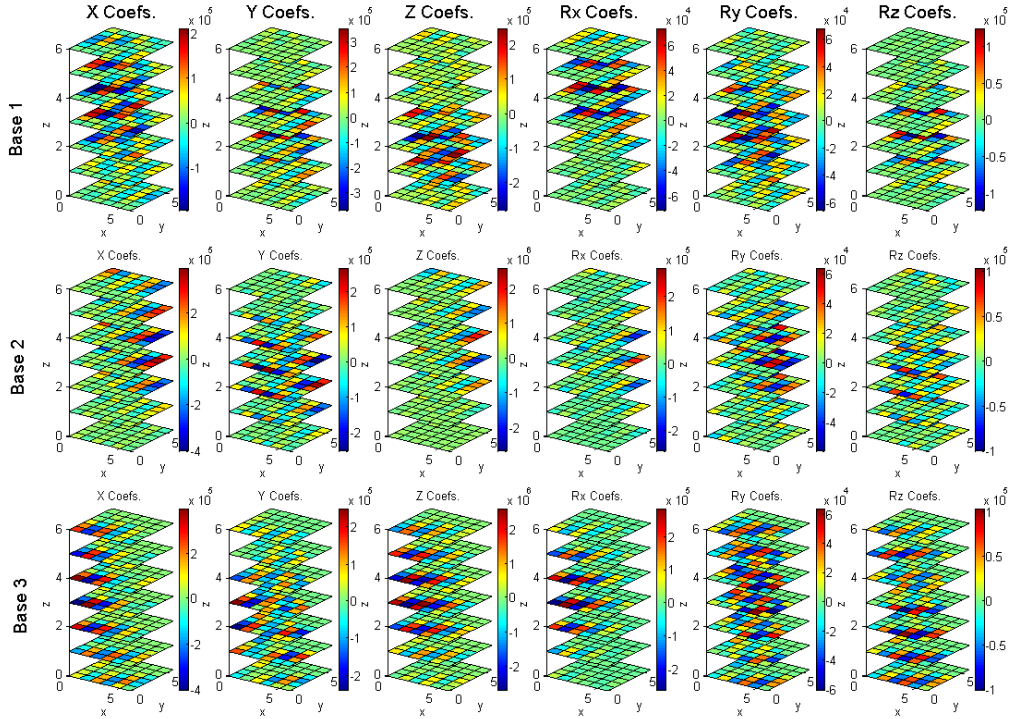
The two separate plots represent an ‘ideal,’ undisturbed data set (NDI Set #1) and a moderately disturbed data set (NDI Set #4). When comparing the distortion maps, the key item to notice is where there is a large spatial gradient of the field. This can be found by looking where the color of the map changes from red at one extreme to blue at the other extreme in a very short interval. Notice that this is not terribly common in the undisturbed case, but in the disturbed case the curvature has a tendency to be high at quite a few places.

The other key concept that can be deduced from these plots is the variation in the error map with respect to the sensor orientation. Recall from the previous chapter that the errors are very different for dissimilar orientations. By examining the following plots for three different base orientations ( $x$ ,  $y$ , and  $z$  axes are shown), we see again that the error is very orientation dependent. Notice that the patterns of the coefficients of the distortion map are drastically different for the three different base orientation vectors. This further enforces this concept and shows why we must be careful to subdivide the orientation space appropriately.

*Visual Representation of Coefficients for Undisturbed Data Set*



*Visual Representation of Coefficients for Disturbed Data Set*



Bernstein coefficients for 6<sup>th</sup> order model of NDI Set #1 & #4  
(Normalized in 7x7 grid, for three orthogonal base orientations)

Figure 3-8

## Error Analysis of Bernstein Approximation

This section details a method for determining the accuracy of a polynomial for representing an error field. It is assumed that a high order can be used to represent the actual error field, and a low order polynomial represents the best-fit approximation of the field. The difference between these polynomials gives valuable insight into the accuracy of our representation of error with a polynomial.

### Determining error from polynomial estimation

Assume the true error represented by high order polynomial ( $n$  order):

$$e(x, y) = \sum_{i=0}^n \sum_{j=0}^n e_{i,j} B_i^n(x) B_j^n(y) = C^n(x, y) \cdot \bar{e}$$

Where:  $\bar{e} \in \mathbb{R}^{(n+1)^2}$  is the actual error at the control points of the high order polynomial

A low order polynomial ( $l$  order) is fit to the data:

$$f(x, y) = \sum_{i=0}^n \sum_{j=0}^n f_{i,j} B_i^l(x) B_j^l(y) = D^l(x, y) \cdot \bar{f}$$

Where:  $\bar{f} \in \mathbb{R}^{(l+1)^2}$  is the actual error at the control points of  $\bar{e}$ , the high order polynomial

If the low order polynomial model is first order ( $l=1$ ):

Solving for four sample points:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix} = \begin{bmatrix} \cdots & D_1^1 & \cdots \\ \cdots & D_2^1 & \cdots \\ \cdots & D_3^1 & \cdots \\ \cdots & D_4^1 & \cdots \end{bmatrix} \begin{bmatrix} \bar{f} \end{bmatrix} \quad \bar{\varepsilon} = \underline{D} \bar{f}$$

To solve for the coefficients,  $f$ , of the low order approximation:

$$\bar{f} = \underline{D}^{(-1)} \bar{\varepsilon} \quad (\text{May be pseudo inverse, actually solve as 'Ax=b' problem})$$

Therefore:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix} = \begin{bmatrix} D_{1,1}^1 & D_{1,2}^1 & D_{1,3}^1 & D_{1,4}^1 \\ D_{2,1}^1 & D_{2,2}^1 & D_{2,3}^1 & D_{2,4}^1 \\ D_{3,1}^1 & D_{3,2}^1 & D_{3,3}^1 & D_{3,4}^1 \\ D_{4,1}^1 & D_{4,2}^1 & D_{4,3}^1 & D_{4,4}^1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \Rightarrow \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} D_{1,1}^1 & D_{1,2}^1 & D_{1,3}^1 & D_{1,4}^1 \\ D_{2,1}^1 & D_{2,2}^1 & D_{2,3}^1 & D_{2,4}^1 \\ D_{3,1}^1 & D_{3,2}^1 & D_{3,3}^1 & D_{3,4}^1 \\ D_{4,1}^1 & D_{4,2}^1 & D_{4,3}^1 & D_{4,4}^1 \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix}$$

Where:  $\varepsilon_i = e(x_i, y_i)$  from high order (accurate) polynomial

The error between high order (accurate) and low order (approximate) solution for an arbitrary point  $(x,y)$  is:

$$g(x, y) = e(x, y) - f(x, y)$$

Plugging in the above equations:

$$g(x, y) = C^n(x, y) \cdot \bar{e} - D^l(x, y) \cdot \bar{f}$$

Where:  $\bar{f}$  is a function of  $e(x,y)$  and  $[D^l(x, y)]^{(-1)}$ .

Plugging into the error equation:

$$g(x, y) = C^n(x, y) \cdot \bar{e} - D^l(x, y) \cdot (D^{(-1)} \bar{e})$$

Where:  $D^l \in \mathbb{R}^{(l+1)^2}$  corresponds to the Bernstein basis of order  $l$  for the current position

$\underline{D} \in \mathbb{R}^{(l+1)^2 \times (l+1)^2}$  corresponds to the Bernstein basis of order  $l$  for the  $(l+1)^2$  control points of the lower degree polynomial

Therefore, given the coefficients  $\bar{e}$  of a high order polynomial, we can determine the error in the low order approximation at a given point  $(x,y)$ .

For an example, we can work out the 1D problem. Higher dimensional problems follow similarly, but are very difficult to interpret due to their complexity.

In 1D, we have the following equation for the error between the high order and the low order polynomial:

$$g(x) = e(x) - f(x)$$

For a 2<sup>nd</sup> order  $e(x)$ :

$$e(x) = e_0(1-x)^2 + e_1 2x(1-x) + e_2 x^2$$

For a 1<sup>st</sup> order  $f(x)$ :

$$f(x) = f_0(1-x) + f_1 x$$

Since the control points of  $f(x)$  are a subset of those of  $e(x)$ :

$$f_0 = e_0$$

$$f_1 = e_2$$

Solving for the error,  $g(x)$ :

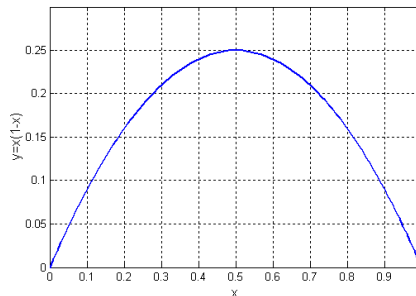
$$g(x) = [e_0(1-x)^2 + e_1 2x(1-x) + e_2 x^2] - [e_0(1-x) + e_1 x]$$

$$g(x) = e_0(1-x)[(1-x)-1] + e_1 2x(1-x) + e_2 x[x-1]$$

$$g(x) = -e_0 x(1-x) + e_1 2x(1-x) - e_2 x(1-x)$$

$$g(x) = x(1-x)[2e_1 - (e_0 + e_2)]$$

Since  $x(1-x)$  was able to be pulled out, we can put a bound on it to help simplify the problem. See Figure 3-9 for a plot of this function.



**Figure 3-9**

We can see that the following bound holds for  $0 \leq x \leq 1$ :

$$x(1-x) \leq 0.25$$

Putting a bound on the error,  $g(x)$ , we get the following:

$$|g(x)| \leq \left| \frac{1}{4} [2e_1 - (e_0 + e_2)] \right|$$



Simplifying the bound, we end up with this inequality describing the error between a 2<sup>nd</sup> order Bernstein polynomial and a 1<sup>st</sup> order Bernstein polynomial.

$$|g(x)| \leq \frac{1}{2} \left| e_1 - \frac{(e_0 + e_2)}{2} \right|$$

Therefore, for this case we can show that the maximum error between the two polynomials is the difference between the error in the middle of the field and the average of the errors at the edges.

We can assume the worst case scenario is where there is no distortion at the edges and maximal distortion at the midpoint, and therefore the low order model has no effect. In this case,  $e_0 = e_n = 0$ .

Therefore, we can get the following simplified bound:

$$|g(x)| \leq \frac{1}{2} e_1$$

For higher order polynomials, we can extend this to get the following bound:

$$|g(x)| \leq \frac{n}{4} \left| \frac{\sum_{i=1}^{n-1} e_i}{n-1} \right|$$

Where, the term in the absolute values indicates the average error at the control points of the high order polynomial excluding the edges.

This can be extended analytically to high order polynomials working in a six dimensional space (such as in the case we are concerned with). However, it is more practical in such a case to determine the error empirically since the equations will get prohibitively complex to be useful.

## Experimental Verification

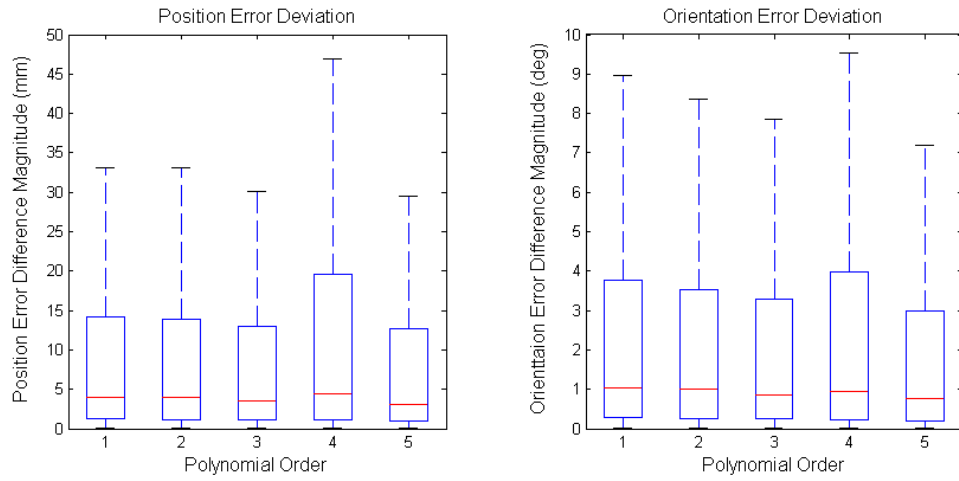
In order to verify this experimentally, we take a high order polynomial (6<sup>th</sup> order), and assume this is the ground truth for the error. Then 1000 random measurements (position and orientation) within the bounds of that polynomial are generated. We use the polynomial model to determine the baseline for the error,  $e(x,y,z,Rx,Ry,Rz)$ . We then take the same data set and plug it into lower order polynomials generated from the same environment. This gives us the lower order model's prediction of the error,  $f(x,y,z,Rx,Ry,Rz)$ . We then use the following to determine the deviation between the two.

Deviation between high and low order error approximation:

$$\gamma(x, y, z, Rx, Ry, Rz) = |g(x, y, z, Rx, Ry, Rz) - f(x, y, z, Rx, Ry, Rz)|$$

The value for  $\gamma$  is determined for each of the 1000 points for each polynomial order with respect to the 6<sup>th</sup> order model. The following plots show these results for two separate environments. The first, shown in Figure 3-10, represents an experiment where data was collected in a highly distorted environment (large magnitude of errors), but the distortion pattern is very smooth (low curvature). This is why there seems to be no distinct trend in the residual error since a lower order polynomial (2<sup>nd</sup> or 3<sup>rd</sup> order) represents this data just as well as a higher order one does. The later results, shown in Figure 3-11, represent an almost ideal data set with low distortion (very small error magnitudes). However, the distortion that does exist has very high curvature (i.e. changes very rapidly, almost randomly) and requires a high order polynomial to represent it. Therefore, there is an upward trend in both the mean and the variance values for the deviation from the ideal model to the lower order models as model order is decreased. Both of the experiments use polynomials generated for 14 base orientations.

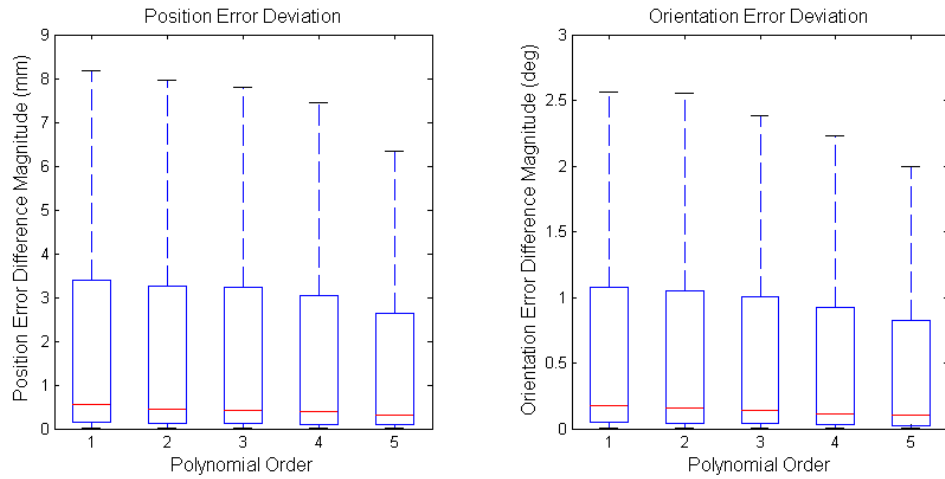
**High Distortion Magnitude, Low Curvature Model**



Deviation in modeled error between low order models and high order 'true' model  
(Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)

**Figure 3-10**

**Low Distortion Magnitude, High Curvature Model**



Deviation in modeled error between low order models and high order 'true' model  
(Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)

**Figure 3-11**

## **Characterization Results**

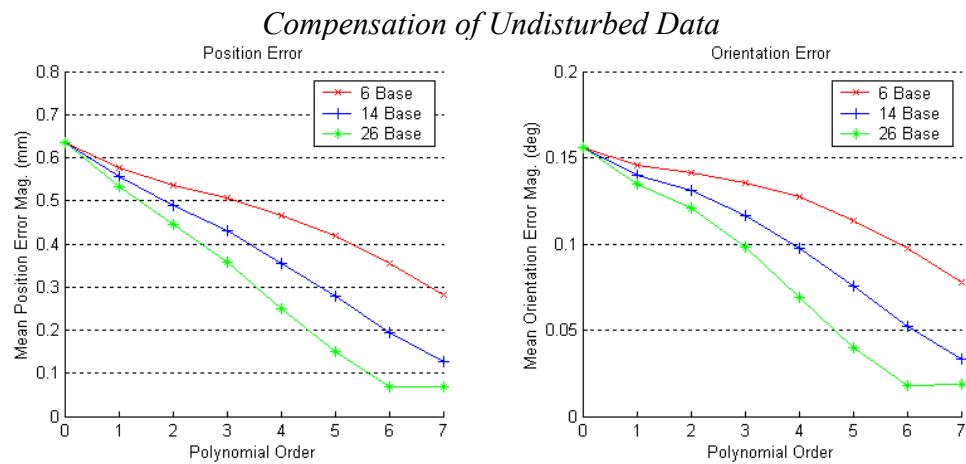
Polynomial models are created and verified using the collected datasets. From these models, plots like those following in this section are generated. By finding the minimums in terms of position and angular error, the best model parameters can be determined. However, minimal error is not the only concern; the number of data points collected must be kept small such that performing such compensation still remains practical. So, to determine the optimal configuration, we look for where there are steep changes in the accuracy of the polynomial and choose parameters accordingly. It is important to note that the error measure is not only that of the original data set used to generate the model because that will be biased; for all data sets, an independent set of points from the same collection is compensated with the model, and its residual errors are used to gage the adequacy of the model. Without doing this, it is clear that the model residual error will always get lower with higher order polynomial fits of the original set, but this is due to over-fitting of the data. An independent set allows us to see if the model is truly representative in of the general case.

The following are results obtained by performing characterizations of the measurement field distortion for various environments. For all of these sets, data was collected and analyzed to generate a best-fit set of Bernstein polynomials to model the error. For these datasets, there are two key items are plotted: the residual error of the original data set after fitting a polynomial, and the error remaining after applying the polynomial to compensate readings for an independent data set. This independent set, in the some cases, is separated into two entities: one for an independent set of the same sensor, and one for a different sensor in the same field. For each of these, there are plots representing the data in two ways, either as a function of *Polynomial Order* for various *Base Vectors*, or vice versa. They are both useful for determining the optimal. In each plot, both the position and orientation error are displayed.

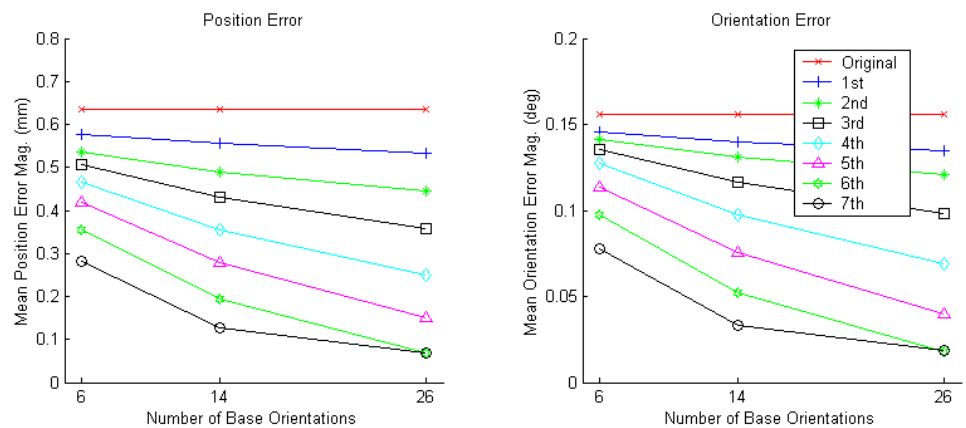
## Ideal Undisturbed Set (NDI Set #1)

The following plots show the results obtained from a distortion model generated from a random set of 2750 data points from a single sensor in the 'ideal,' undisturbed environment (NDI Set #1). The errors shown are the average magnitudes of the residual errors when the model is applied to an independent set of 2750 different random measurements of the same sensor in the same environment. These results represent the effects of compensation on the systematic distortion of the Aurora system itself in an undisturbed environment. There are plots that show the residual error after compensation as a function of both the polynomial order (Figure 3-12) and the number of base orientation vectors used (Figure 3-13).

### Mean Error After Compensation (Same Data Set)



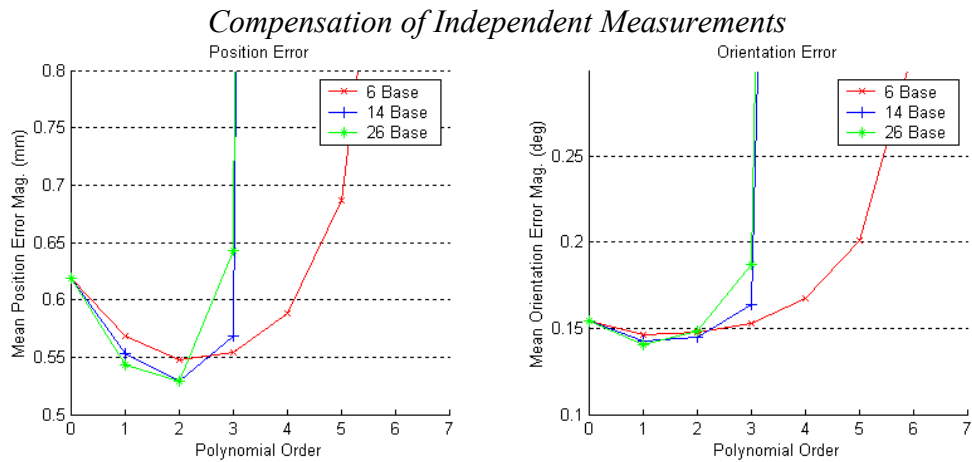
Residual error of modeling undisturbed data set as a function of polynomial order  
**Figure 3-12**



Residual error of modeling undisturbed data set as a function of base orientations  
**Figure 3-13**

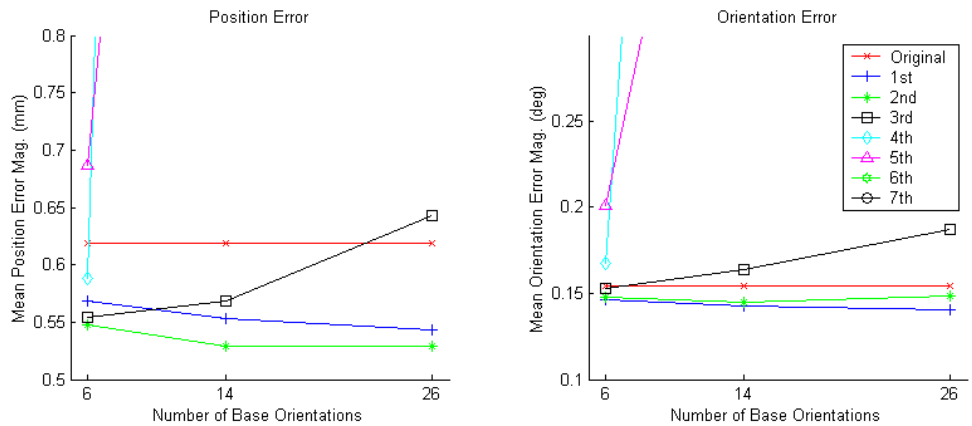
## Mean Error After Compensation (Independent Data Set, Same sensor)

The above results show that we can in fact model sensor measurements accurately, but what is much more important is to be able to accurately represent the distortion map of the measurement system. Therefore, here we take the model obtained before and apply it to a completely independent set of data collected at the same time in the same environment. Figure 3-14 shows the shape of that we would expect; the error improves as the model is applied and as the order increases, the error decreases. As the order of the polynomial gets too high, it starts to become very specific to the original data set and becomes less general; therefore, the error begins to increase. The parameters used to generate the polynomial that correspond to the minima of the plots are optimal for this configuration. Figure 3-15 shows the same results, but is presented another way which makes it very clear which order polynomial we should pick; in this case it looks like a 2<sup>nd</sup> order polynomial with 14 base orientations would work best.



Residual error of modeling undisturbed data set as a function of polynomial order

**Figure 3-14**

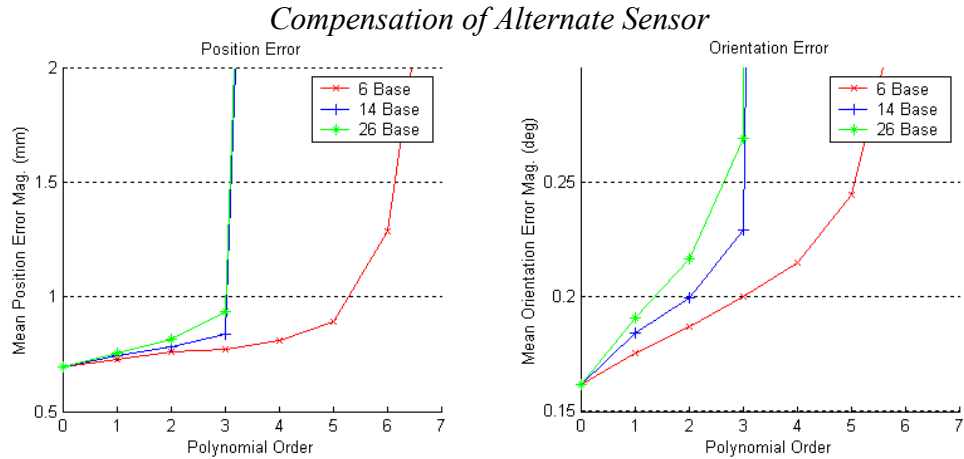


Residual error of modeling undisturbed data set as a function of base orientations

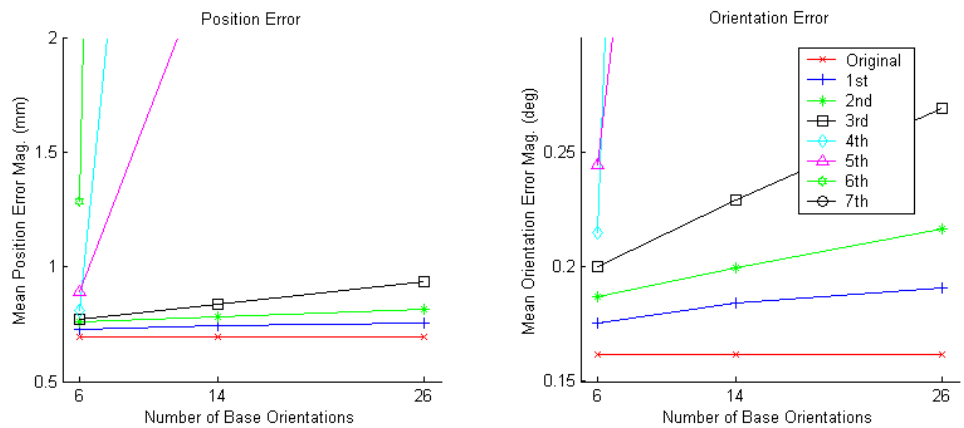
**Figure 3-15**

## Mean Error After Compensation (Independent Data Set, Different sensor)

These results present what happens when we apply our model to a different sensor. Unfortunately, these results in Figure 3-16 are not what we would hope to see because the error no better than the original (*Polynomial Order* = 0) in any case. This can most likely be attributed to one of two issues: 1) the sensor measurements were not distributed in similar orientations to the other sensor (which would not be a problem if the original data set was distributed evenly, which it unfortunately was not), or 2) the sensor coils themselves are inconsistent (this effect would most likely cancel out if a large number of different sensors were used in the original data collection to generate the model).



Residual error of modeling undisturbed data set as a function of polynomial order  
**Figure 3-16**

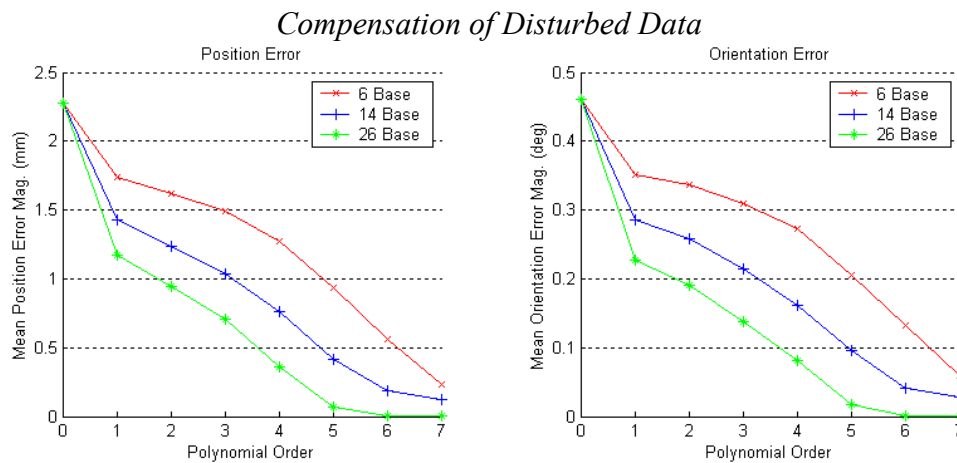


Residual error of modeling undisturbed data set as a function of base orientations  
**Figure 3-17**

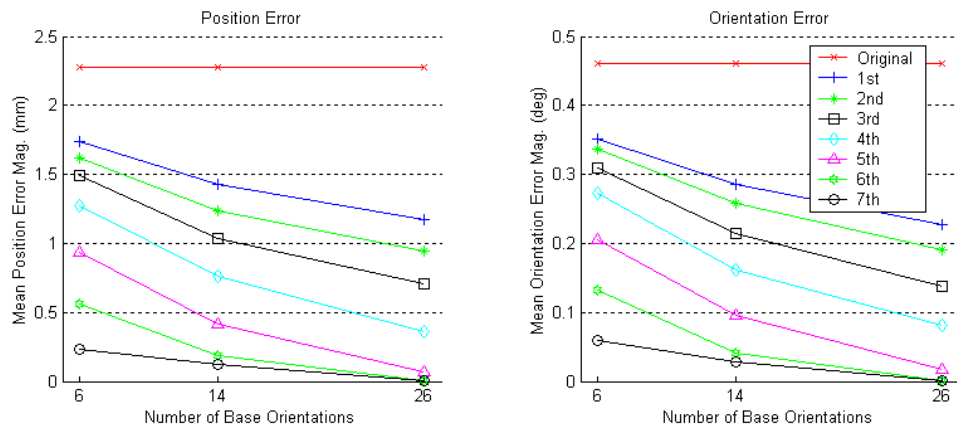
## Moderately Disturbed Set (NDI Set #4)

These results represent the effects of measurement compensation on the distortion of the Aurora system in a reasonably disturbed environment; the data is from (NDI Set #4). The plots show the effects on the mean error as a function of polynomial order (Figure 3-18) and the number of base orientations used (Figure 3-19). Here, the results are obtained from a distortion model generated from a random set of 1025 data points from a single sensor. The errors shown are the average magnitude of the residual errors when the model is applied to an independent set of 1025 different random points for the same sensor in the same environment.

### Mean Error After Compensation (Same Data Set)



Residual error of modeling disturbed data set as a function of polynomial order  
**Figure 3-18**



Residual error of modeling disturbed data set as a function of base orientations  
**Figure 3-19**



## Statistical Analysis of Mean Error After Compensation

To further understand the effects of the model parameters on the residual error we can model this residual error as a function of the factors *Order* and *NumBase*. The following statistical analysis can be used to help determine the contribution of the number of base orientation vectors and the order of the polynomial model. The first model is presented in Table 3-3, and represents the position error. The second model is shown in Table 3-4, and is for orientation error. These models were arrived at using backward stepwise elimination regression. Using this technique, it was determined that the second and higher order terms for *NumBase*, the third order and higher terms *Order*, and the interaction terms are insignificant for  $\alpha=0.05$ .

<i>Position Error Model</i>					
*** Linear Model ***					
Call: lm(formula = <b>PosErr ~ Order + NumBase + Order^2</b> , data = NDI4OrigSet, na.action = na.exclude)					
Residuals:					
	Min	1Q	Median	3Q	Max
	-0.303804	-0.1042885	-0.01927109	0.06238903	0.4607773
Coefficients:					
	Value	Std. Error	t value	Pr(> t )	
(Intercept)	2.5416030	0.1181005	21.5206846	0.0000000	
<b>Order</b>	-0.4421744	0.0620358	-7.1277319	<b>0.0000007</b>	
<b>NumBase</b>	-0.0280454	0.0047510	-5.9030727	<b>0.0000090</b>	
<b>I(Order^2)</b>	0.0228109	0.0085213	2.6769382	<b>0.0144902</b>	
Residual standard error: 0.191302 on 20 degrees of freedom					
Multiple R-Squared: 0.9406137					
F-statistic: 105.5926 on 3 and 20 degrees of freedom, the p-value is 1.96354e-012					
Analysis of Variance Table					
Response: PosErr					
Terms added sequentially (first to last)					
	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Order	1	10.05544625	10.05544625	274.7654937	0.00000000000
NumBase	1	1.27525027	1.27525027	34.8462676	0.00000896594
I(Order^2)	1	0.26225022	0.26225022	7.1659983	0.01449021059
Residuals	20	0.73192933	0.03659647		
S-plus analysis of model for residual position error					

**Table 3-3**

*Orientation Error Model*

```

*** Linear Model ***

Call: lm(formula = OriErr ~ Order + NumBase + Order^2, data =
NDI4OrigSet, na.action
= na.exclude)
Residuals:
      Min       1Q   Median       3Q      Max
-0.06611168 -0.02113476 -0.002801003  0.01413854  0.1020159

Coefficients:
              Value Std. Error  t value Pr(>|t|)
(Intercept)  0.5191741  0.0253748  20.4602553  0.0000000
Order      -0.0853538  0.0133288  -6.4036873  0.0000030
NumBase   -0.0061881  0.0010208  -6.0620826  0.0000063
I(Order^2)  0.0042244  0.0018309   2.3073361  0.0318590

Residual standard error: 0.04110266 on 20 degrees of freedom
Multiple R-Squared: 0.9320076
F-statistic: 91.38345 on 3 and 20 degrees of freedom, the p-value is
7.567724e-012

Analysis of Variance Table

Response: OriErr

Terms added sequentially (first to last)
      Df  Sum of Sq  Mean Sq  F Value  Pr(F)
Order  1  0.3920787367  0.3920787367  232.0777182  0.000000000000
NumBase 1  0.0620845505  0.0620845505  36.7488453  0.00000632246
I(Order^2) 1  0.0089941802  0.0089941802  5.3238001  0.03185900435
Residuals 20  0.0337885722  0.0016894286

```

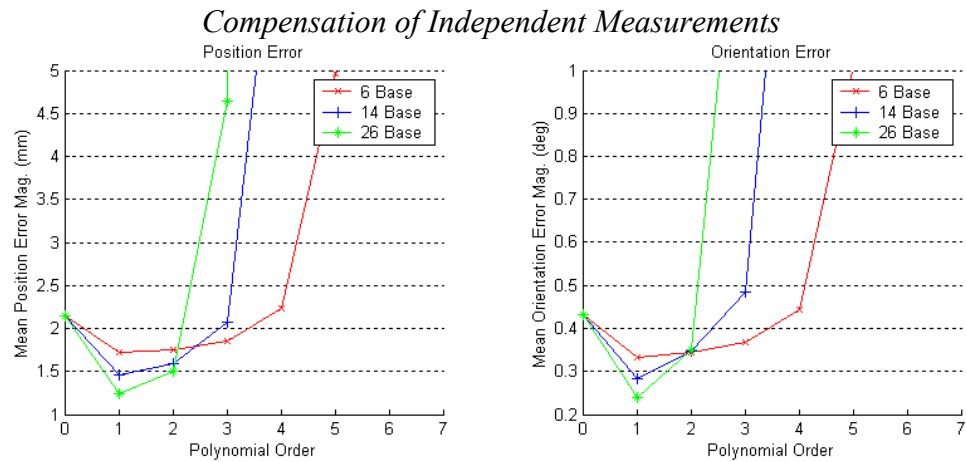
S-plus analysis of model for residual orientation error

**Table 3-4**

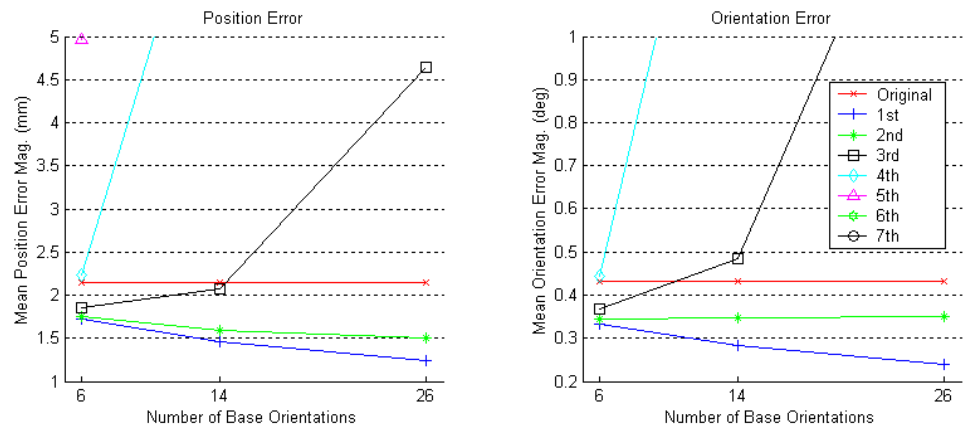
## Mean Error After Compensation (Independent Data Set, Same sensor)

As before, we again apply this model to a completely independent set of data collected at the same time in the same environment. Figure 3-20 shows the shape of the same trend as in Figure 3-14; the error improves as the model is applied and as the order increases, the error decreases. As the order of the polynomial gets too high, it starts to become very specific to the original data set, and is less general and therefore the error begins to increase. Figure 3-21 shows the same results, but is presented another way which makes it very clear which order polynomial we should pick.

In this case it looks like a 1<sup>st</sup> order polynomial with 26 base orientations would work best. This is an interesting result because it indicates that the order of the polynomial with respect to position in the field only needs to be 1<sup>st</sup> order, but that we need to have a very fine subdivision of the orientation space to accurately represent the error in the distorted case.



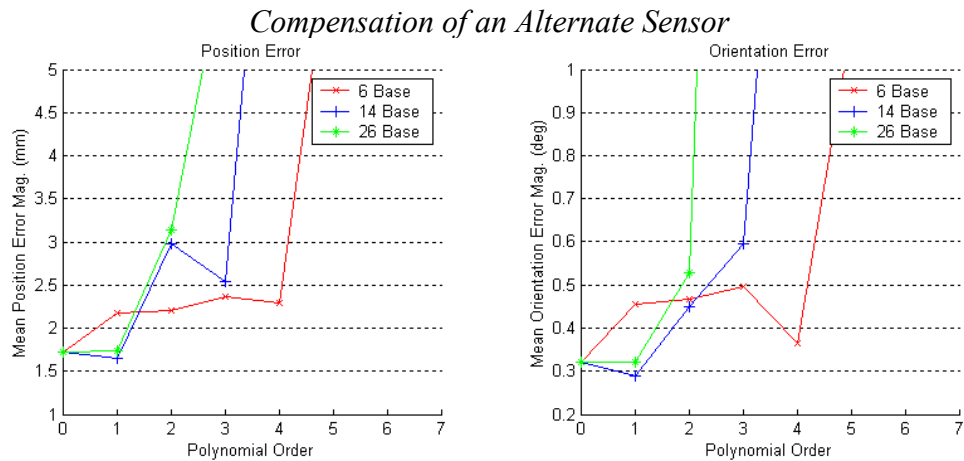
Residual error of modeling disturbed data set as a function of polynomial order  
**Figure 3-20**



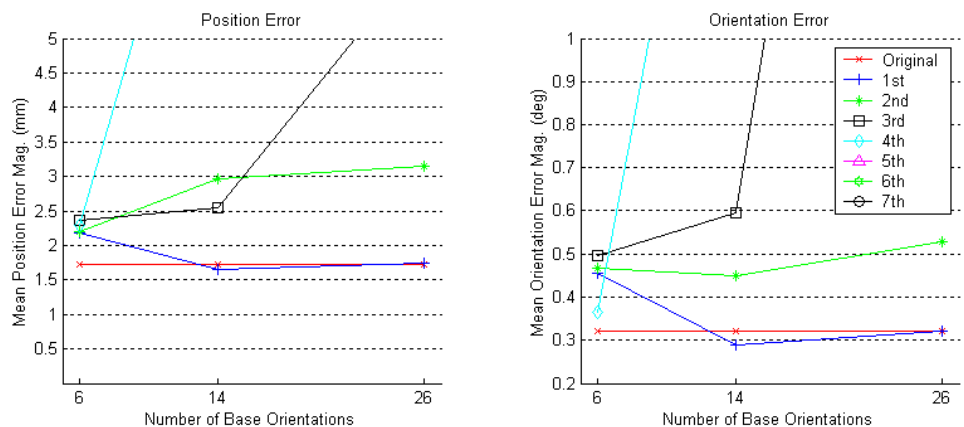
Residual error of modeling disturbed data set as a function of base orientations  
**Figure 3-21**

## Mean Error After Compensation (Independent Data Set, Different sensor)

Again, these results represent what happens when we apply our model to a different sensor that the one used to generate the model. In the case where there is significant field distortion, we do in fact do see an improvement in the error of a separate sensor as shown in Figure 3-22. Although the increase in accuracy after applying the model is not as drastic as for the same sensor, and it falls off rapidly at higher orders, there is a definite increase in accuracy for the model we decided to use (1<sup>st</sup> order with 26 base vectors). This is despite the same issues as before, and would most likely be better if the original sensor distribution were more even in the orientation space and if multiple sensors were used in the collection. By looking at these results, we may wish to refine our choice of parameters to a 1<sup>st</sup> order model with 14 base orientations vectors as opposed to 26 base vectors.



Residual error of modeling disturbed data set as a function of polynomial order  
**Figure 3-22**

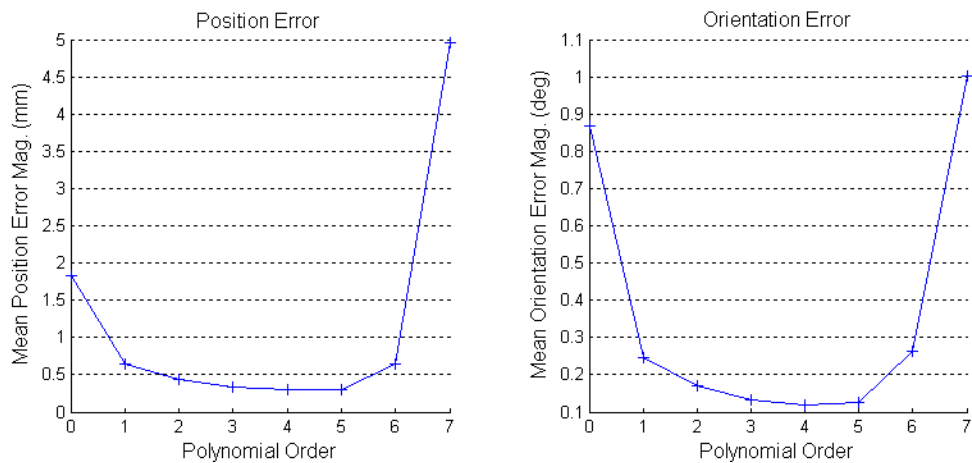


Residual error of modeling disturbed data set as a function of base orientations  
**Figure 3-23**

## Robotic Data Collection in Lab

The following results represent the modeling accuracy of the polynomial for a dense data set collected in the lab with no additional disturbance added. This data set includes 2000 measurements from each of 12 distinct orientations. Figure 3-24 shows the accuracy of the polynomial model when being fit to this dense data set. The errors shown represent the mean magnitude of the residual errors after applying the polynomial to all of the data points in the set. These results are promising in that it appears the accuracy can be improved by increasing the polynomial order; however, after a second or third order model, the improvement is very small. Therefore, we can probably get away with a 2<sup>nd</sup> or 3<sup>rd</sup> order model to accurately represent this environment.

### *Compensation of Robot Lab Data*

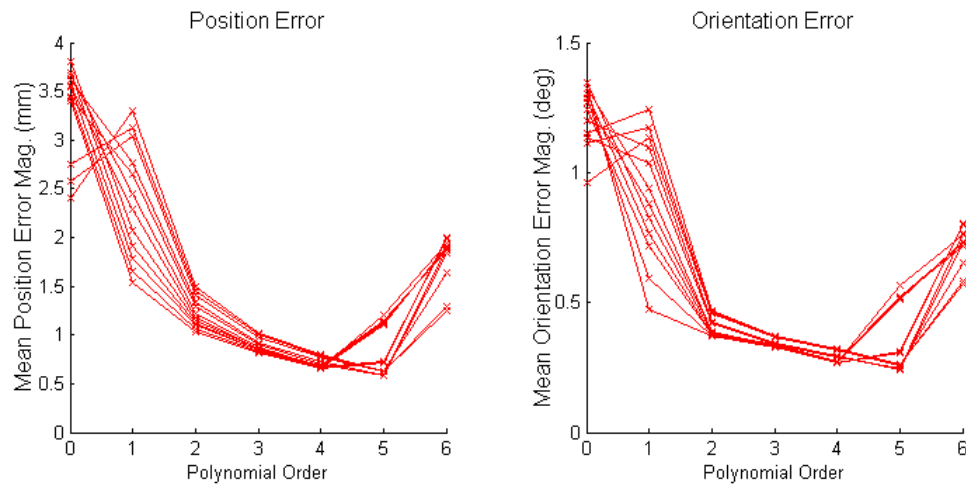


Accuracy of modeling measurement distortion field for robotically collected data in lab  
**Figure 3-24**

## Stand-alone Fixture in Lab

Here, we investigate modeling the distortion field from the data collected with the Stand-alone fixture. Higher order models are meaningless in this case because of the limited number of data points available. Assuming all measurements are valid, a single collection with this fixture would have the bare minimum number of measurements to fit a 5<sup>th</sup> order polynomial. As can be seen in Figure 3-25, we can in reality reliably fit up to a 4<sup>th</sup> order polynomial, with 3<sup>rd</sup> order polynomials appearing to be the most practical. Compare these results to those of the disturbed OR environment in Figure 3-28; the trends seem similar, but the errors are much lower. However, an important feature to notice is how the change in residual error starts to level out more after 2<sup>nd</sup> order in the disturbed environment; this is expected because we have already seen that a distorted environment can usually be modeled with a lower order polynomial than that of an undisturbed one.

### *Compensation of SA Lab Data*

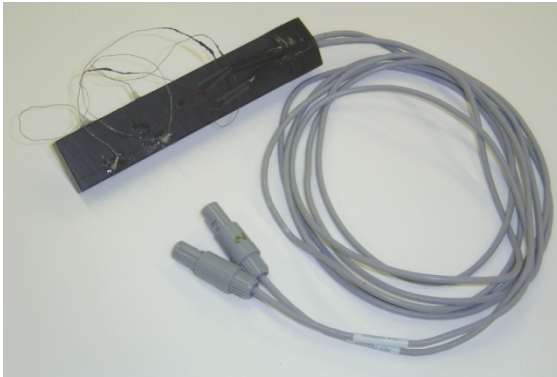


Residual error magnitude of modeling disturbed data set as a function of *Polynomial Order* for each of 12 orientations

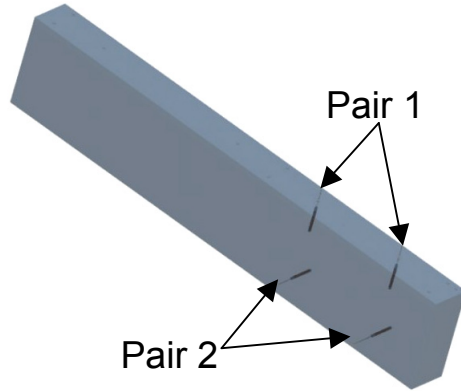
**Figure 3-25**

## Compensation Results

In order to verify that the compensation is in fact helping, I performed an experiment that will help verify this. In this experiment, the plastic tool shown in Figure 3-26 was accurately machined such that we know the relative position and orientation of four sensors on it. By picking a pair of sensors on this tool, we know that the relative translation between the two sensors is precisely  $30\text{mm} \pm 0.005\text{mm}$  and the relative orientation is  $0^\circ$ .

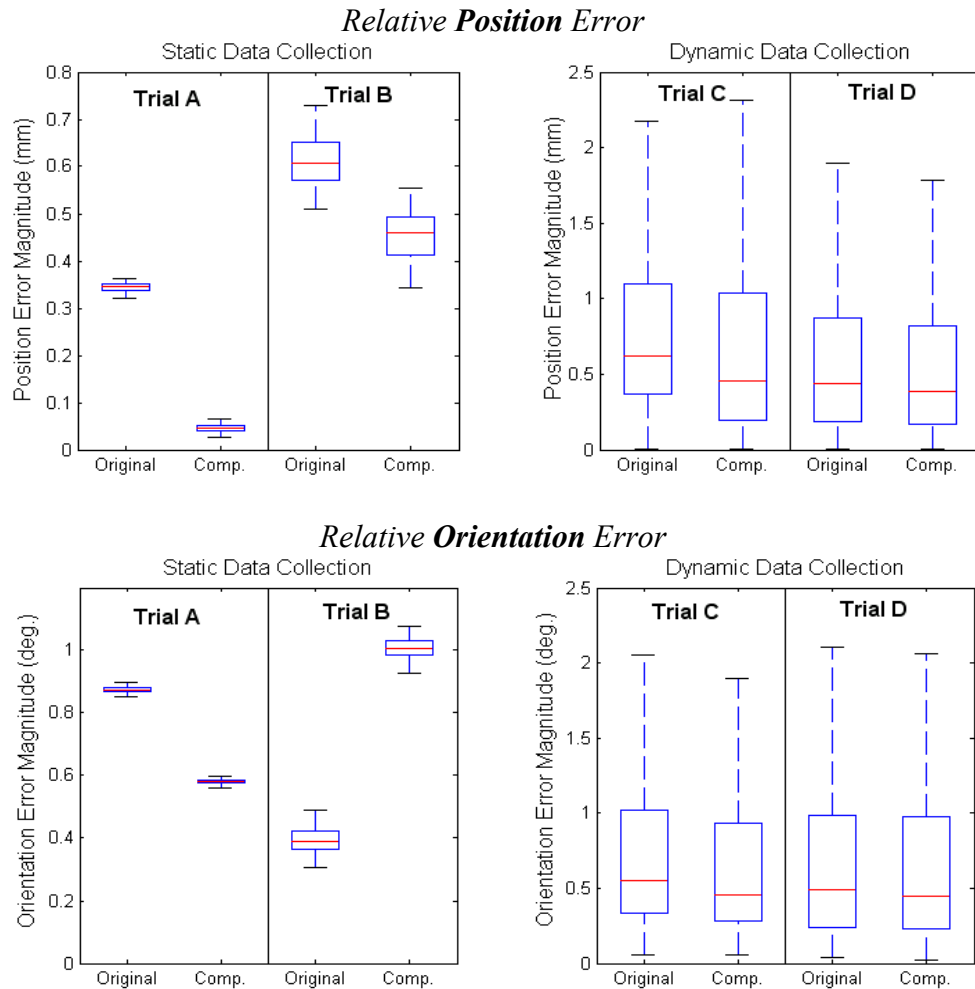


Test tool used for validation experiments  
**Figure 3-26**



CAD model of test tool design

Four trials were performed using this tool, and in each trial the measurements were compensated in real-time using the *Aurora Tools* software I have written that is discussed later. Measurements were compensated using the previously mentioned 3<sup>rd</sup> order polynomial model for the lab environment that was generated using the SA fixture. Of the four trials, two were collected statically with 100 measurements for each sensor in each of ten tool poses. The other two were collected dynamically by slowly moving the tool through the working volume of the Aurora and collecting 1000 measurements of each sensor in the tool. By examining the results in Figure 3-27, the compensation does have a quite drastic improvement on tracking results in all but one case; this is especially true for data collected quasi-statically since this was the data collection method used for generation of the polynomial model. For these results, Pair #1 from the right of Figure 3-26 was used; results are similar for the other pairs.



Relative position and orientation error between two sensors  
(Compensation using 3<sup>rd</sup> order polynomial model of lab environment)

**Figure 3-27**

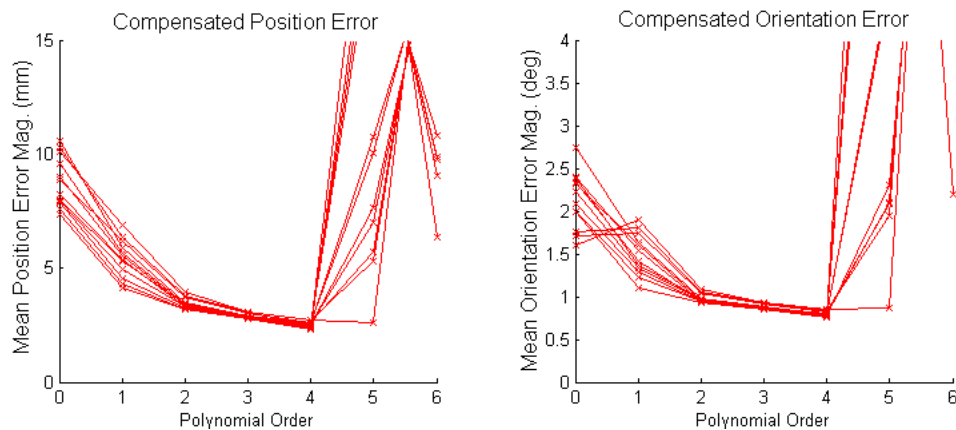
These results in Figure 3-27 show that there is in fact significant potential for real-time compensation of sensor measurements using a field characterization. The distortion model used here was generated from a rather sparse set of data collected using the Stand-alone fixture; the results would undoubtedly be even better if a more dense data collection were performed. In this experiment, only one trial seems to have spurious results; this is the orientation error of Trial B. It appears that the orientation error got worse even though the position error got better. This can most likely be attributed to one of two actors: 1) the orientation error was already the lowest of the group, or 2) the orientations collected were not similar to those used to generate the model. Either way, in a more dense data collection this would probably not be a problem.



## Stand-alone Fixture in Operating Room

The results presented here are based upon a data set collected in the operating room (OR) using the digitized Stand-alone fixture. This data set contains 204 evenly dispersed measurements for each of 12 orientations; ideally there would be 216 measurements, but invalid readings were discarded along with their counterparts of the other orientations. Figure 3-28, displays the average magnitudes of the residual errors for each of the 12 orientations after compensation; they are plotted as a function of the polynomial order. Notice that even though the sensors behave differently and have varying initial errors (shown as  $Order=0$ ), they all converge to a consistent minimum when modeled with a 4<sup>th</sup> order polynomial. This suggests that we can reliably fit up to 4<sup>th</sup> order models to data sets collected in this configuration. Further, notice that the relative change in residual error after 2<sup>nd</sup> order levels out even more than for the previous set as we would expect because we are in an even more distorted environment.

### Compensation of OR Data



Residual error magnitude of modeling disturbed data set as a function of Polynomial Order for each of 12 orientations

**Figure 3-28**

Although we can reliably fit 4<sup>th</sup> order models to the data, in reality a lower order model may be more useful. This is because we are not trying to get a perfect model for these measurements, we want a model that is representative of the distortion field and can be applied to real-time measurement compensation in the OR. Therefore, probably a 2<sup>nd</sup> or 3<sup>rd</sup> order model would be used since it is more general, and still retains a low residual error with the original data set.

## Compensating for a Tool in the OR

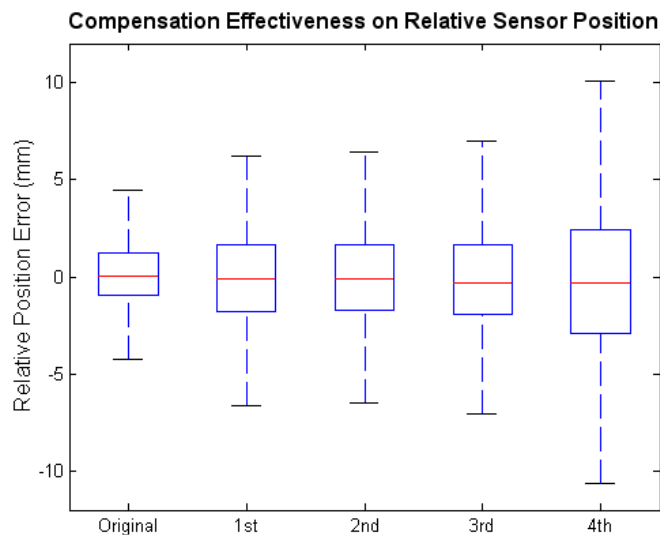
Here we try out the compensation methods on a tissue shaver bit in the operating room using the tool shown in Figure 3-29. The following results are obtained by determining the errors in the relative position of two sensors on the tool. The analysis is performed for the original measurements and again after performing measurement compensation on the sensor readings using several different polynomial models as shown in Figure 3-30.



Tracked tissue shaver tool

**Figure 3-29**

Unfortunately, this result is not what we would hope to see. Applying the distortion compensation to this tool in fact made the uncertainty in the relative position between the two sensors even greater. This is most likely due to additional distortion caused by the metal tool and accuracy loss due to dynamic data collection. Unfortunately, we did not have a plastic tool to include in this trial at the time of the OR experiments. One thing to note is that the variance of the trials is very similar for 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> order models, and the 2<sup>nd</sup> order model seems best. These results seem reasonable when referring back to Figure 3-28.



Relative position error between two sensors on tissue shaver in the OR with comparison between effects of compensation model order (Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)

**Figure 3-30**

## ***Contributions***

This chapter took the data collection techniques from Chapter 2 and exploited the results for generation of polynomial models that describe the distortion. The basis of the modeling technique is again from [Wu, 2003c]. However, much time and effort was put into optimization of this software in terms of its robustness, speed, and reliability. With the distortion modeling software functional, the error mappings from Chapter 1 were transformed into distortion models. These models were analyzed to determine the extent of the distortion, the shape of the distortion, and the consistency. Verification of these models was performed by applying them to independent data sets and charting tracking accuracy improvements. The key results of the section are twofold: 1) generation of models describing several measurement environments, and 2) optimization of the model in terms of polynomial order, orientation resolution (number of basis orientations), and number of data points required. In Chapter 5 we use these models to generate a simulator of the distortion field that allows for tool design without physically building the tools.

## Chapter 4: Optimal Tool Design

### *Overview*

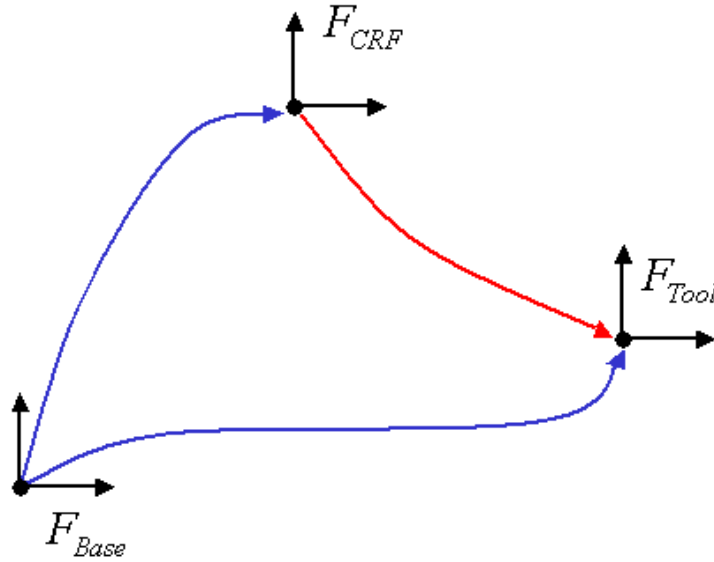
Design of an ‘optimal’ tool means designing a tracked tool and reference frame such that they function with the best possible accuracy for a given scenario while still maintaining practicality/feasibility of construction. Designing tools with the best possible accuracy falls into two possible categories: 1) analysis of error bounds and uncertainty based on the tool configuration, and 2) simulation of the tool design either in a virtual simulator, as described later, or by building the tools. However, just because a tool is ‘optimal’ in the sense of accuracy, it may not be practical; therefore this is an important factor in the design. For example, a tool with the best possible accuracy would have a very large number of sensors distributed evenly over a relatively large volume, with the target directly in the center; obviously there is a practical limit on number of sensors that can be used, tool size, and location of the tool tip.

This section details possible steps for design of a tool. Some of the steps are applicable to all tracked tools including OTS, while others are very specific to EMTS applications, and some even more specialized to the Aurora’s 5 DoF sensors. First, bounds are placed on both position and orientation error of the tool–reference registration. Following the placement of bounds is an uncertainty analysis that gives a more reasonable approximation of the distribution of errors. This section is rather generic to just about any IGS application.

Following the generic tracking scenario is a detailed description of the methods for fitting tool frames to the sensor measurements. The idea of fitting a rigid body frame to individual sensor measurements is rather generic, but the methods here are extended from the 3 DoF points obtained from an OTS to the 5 DoF measurements obtained from the Aurora EMTS. Following is an analysis of the error bounds and the error uncertainty of these fits. These values are then plugged into the previous section to determine the overall tool registration error.

## Tool Tracking Overview

Here, we investigate error propagation in the tool tracking scenario. For this situation, independent of the tracker type, there is generally a coordinate reference frame (CRF) that is used to track the target, i.e. the patient anatomy, with respect to a base frame (the tracking unit). Then, there is the tool that is also tracked with respect to this same base frame. The goal is to determine the relative transformation (position and orientation) of the tool with respect to the CRF. The general tracking scenario is shown in Figure 4-1.



Frame transformations in the standard tool tracking scenario  
**Figure 4-1**

In this case, the transformation representing the tool in the reference frame is:

$$F_{CRF}^{Tool} = \left( F_{Base}^{CRF} \right)^{-1} F_{Base}^{Tool}$$

Where:

$$F_{CRF} \triangleq F_{Base}^{CRF}$$

$$F_{Tool} \triangleq F_{Base}^{Tool}$$

These frames,  $F_i$ , are homogeneous transformations represented as follows:

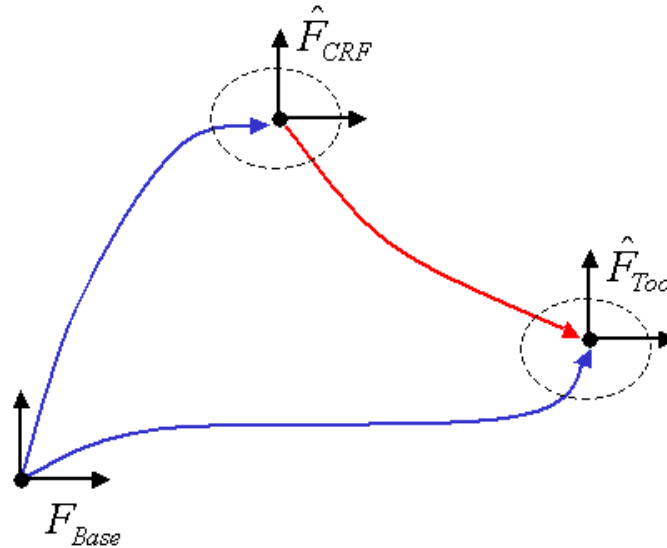
$$F_i = \begin{bmatrix} R_i & \bar{p}_i \\ 0 & 1 \end{bmatrix} \in SE(3)$$

Where:  $R_i \in SO(3)$  is the rotation element

$\bar{p}_i \in \mathbb{R}^3$  is the translation element

## Tool Tracking Error Bounds

The measured transformation just mentioned is the one we must use for performing calculations, but for a real tracking system it is not as simple as it appears due to inaccuracy of measurements. It must be assumed that the frames representing the CRF and the tool in the base frame are only approximations of the true frame, and errors in these approximations can be bounded based on knowledge of the tracking system's accuracy. Figure 4-2 displays this problem, where the dashed lines represent bounds that the error stays within.



Measured frame transformations with error bounds

**Figure 4-2**

Here,  $\hat{F}_{CRF}$  and  $\hat{F}_{Tool}$  represent the measurements of the CRF and tool frames respectively with respect to the base frame.

Each of these measured frames has an error associated with it. Using the *right error* representation, the measurement is as follows:

$$\hat{F}_{CRF} = F_{CRF} \Delta F_{CRF}$$

$$\hat{F}_{Tool} = F_{Tool} \Delta F_{Tool}$$

Where:  $F_{CRF}$  and  $F_{Tool}$  represent the nominal/true transformations

$\Delta F_{CRF}$  and  $\Delta F_{Tool}$  represent the additive measurement error transformation

The error could also be represented as a *left error*, where the error acts in the base frame and the actual transformation is applied to it.

This is a less common representation:

$$\hat{F}_i = \Delta F_i F_i$$

For this work, the *right error* representation is used. The error is represented as a homogeneous transformation that applies a rotation and translation after the nominal measurement. This error transformation is defined as follows:

$$\Delta F_i = \begin{bmatrix} \Delta R_i & | & \delta \bar{p}_i \\ \hline 0 & | & 1 \end{bmatrix} \in SE(3)$$

Where:  $\delta \bar{p}_i \in \mathbb{R}^3$  is the translation element of the error

$\Delta R_i = \exp(\hat{\alpha}_i) \in SO(3)$  is the rotation element of the error

Where:  $\hat{\alpha}_i = skew(\bar{\alpha}_i) = skew(\delta \theta_i \bar{\omega}_i) \in so(3)$

$\bar{\omega}_i \in \mathbb{R}^3$  is the unit vector along the axis of rotation

$\delta \theta_i \in [0, 2\pi]$  is the non-negative angle about  $\bar{\omega}_i$  (radians)

The estimated transformation from the CRF to the Tool in this notation is:

$$\hat{F}_{CRF}^{Tool} = \left[ F_{Base}^{CRF} \Delta F_{Base}^{CRF} \right]^{-1} \left[ F_{Base}^{Tool} \Delta F_{Base}^{Tool} \right]$$

$$F_{CRF}^{Tool} \Delta F_{CRF}^{Tool} = \left[ F_{Base}^{CRF} \Delta F_{Base}^{CRF} \right]^{-1} \left[ F_{Base}^{Tool} \Delta F_{Base}^{Tool} \right]$$

Where:  $F_{CRF}^{Tool}$  is the nominal / actual frame

$\Delta F_{CRF}^{Tool}$  is the error in the estimated / calculated frame

To determine the error in the relative transformation  $\hat{F}_{CRF}^{Tool}$ , we isolate the associated error:

$$\Delta F_{CRF}^{Tool} = \left( F_{CRF}^{Tool} \right)^{-1} \left[ F_{Base}^{CRF} \Delta F_{Base}^{CRF} \right]^{-1} \left[ F_{Base}^{Tool} \Delta F_{Base}^{Tool} \right]$$

$$\Delta F_{CRF}^{Tool} = \left( F_{CRF}^{Tool} \right)^{-1} \left( \Delta F_{Base}^{CRF} \right)^{-1} \left( F_{Base}^{CRF} \right)^{-1} F_{Base}^{Tool} \Delta F_{Base}^{Tool}$$

$$\Delta F_{CRF}^{Tool} = \left( F_{CRF}^{Tool} \right)^{-1} \left( \Delta F_{Base}^{CRF} \right)^{-1} F_{CRF}^{Tool} \Delta F_{Base}^{Tool}$$

Putting into homogeneous representation:

$$\begin{bmatrix} \Delta R_{CRF}^{Tool} & | & \delta \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix} = \begin{bmatrix} R_{CRF}^{Tool} & | & \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix}^{-1} \begin{bmatrix} \Delta R_{Base}^{CRF} & | & \delta \bar{p}_{Base}^{CRF} \\ \hline 0 & | & 1 \end{bmatrix}^{-1} \begin{bmatrix} R_{CRF}^{Tool} & | & \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix} \begin{bmatrix} \Delta R_{Base}^{Tool} & | & \delta \bar{p}_{Base}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix}$$

Passing the inverses through into the frames we get an expression for the error in the estimated frame:

$$\begin{bmatrix} \Delta R_{CRF}^{Tool} & | & \delta \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix} = \begin{bmatrix} \left( R_{CRF}^{Tool} \right)^{-1} & | & -\left( R_{CRF}^{Tool} \right)^{-1} \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix} * \\ \begin{bmatrix} \left( \Delta R_{Base}^{CRF} \right)^{-1} & | & -\left( \Delta R_{Base}^{CRF} \right)^{-1} \delta \bar{p}_{Base}^{CRF} \\ \hline 0 & | & 1 \end{bmatrix} \begin{bmatrix} R_{CRF}^{Tool} & | & \bar{p}_{CRF}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix} \begin{bmatrix} \Delta R_{Base}^{Tool} & | & \delta \bar{p}_{Base}^{Tool} \\ \hline 0 & | & 1 \end{bmatrix}$$

In order to analyze the error, the next step is to break the error apart into rotational and translational components.

### Position Error:

$$\begin{aligned}\delta \bar{p}_{CRF}^{Tool} &= (R_{CRF}^{Tool})^{-1} \left[ (\Delta R_{Base}^{CRF})^{-1} (R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} + \bar{p}_{CRF}^{Tool}) - (\Delta R_{Base}^{CRF})^{-1} \delta \bar{p}_{CRF}^{Base} \right] - (R_{CRF}^{Tool})^{-1} \bar{p}_{CRF}^{Tool} \\ \delta \bar{p}_{CRF}^{Tool} &= (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} (R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} + \bar{p}_{CRF}^{Tool}) - (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} \delta \bar{p}_{CRF}^{Base} - (R_{CRF}^{Tool})^{-1} \bar{p}_{CRF}^{Tool} \\ \delta \bar{p}_{CRF}^{Tool} &= (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} + (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} \bar{p}_{CRF}^{Tool} - (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} \delta \bar{p}_{CRF}^{Base} - (R_{CRF}^{Tool})^{-1} \bar{p}_{CRF}^{Tool}\end{aligned}$$

$$\delta \bar{p}_{CRF}^{Tool} = (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} - (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} \delta \bar{p}_{CRF}^{Base} + (R_{CRF}^{Tool})^{-1} \left( (\Delta R_{Base}^{CRF})^{-1} - I_3 \right) \bar{p}_{CRF}^{Tool}$$

### Rotation Error:

$$\Delta R_{CRF}^{Tool} = (R_{CRF}^{Tool})^{-1} (\Delta R_{Base}^{CRF})^{-1} R_{CRF}^{Tool} \Delta R_{Base}^{Tool}$$

For further analysis, we must define the representation of rotation errors and look for ways to simplify them. This is the same as the representation shown earlier, but now it is expand and linearized.

### Representation of Rotations:

$$\Delta R = \Delta R(\vec{\alpha}) = \exp(\hat{\alpha}) \in SO(3)$$

Where:  $\vec{\alpha} \in \mathbb{R}^3$  is a vector pointing along the axis of rotation

$\|\vec{\alpha}\|$  is the angle of rotation (radians)

$$\hat{\alpha} = skew(\vec{\alpha}) \in so(3)$$

Normally rotations are represented as:  $R = \exp(\hat{\omega}\theta)$ , where  $\vec{\omega}$  is the unit vector representing the axis of the rotation and  $\theta$  is the magnitude of rotation. However, since  $\vec{\alpha} = \vec{\omega}\theta$  is the Rodriguez vector, only  $\hat{\alpha}$  is necessary in the expansion.

Using the infinite series expansion of a matrix exponential to expand the rotation:

$$\Delta R = \exp(\hat{\alpha}) = \sum_{j=0}^{\infty} \frac{1}{j!} \hat{\alpha}^j = I_3 + \hat{\alpha} + \frac{1}{2!} \hat{\alpha}^2 + \frac{1}{3!} \hat{\alpha}^3 + \dots$$

Since the magnitude of the rotation error is small, second order and higher terms will be negligible, and can be neglected. The result is the following linearization:

$$\Delta R = \exp(\hat{\alpha}) \approx I_3 + \hat{\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix}$$



Now that the rotation error is approximated linearly, the next step is to plug this into the equations for position error and orientation error that were presented above. After plugging in and expanding the equations out, we get the following equations that are linear in terms of the error.

### Linearizing the Position Error:

$$\delta \bar{p}_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)^{-1} R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} - \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)^{-1} \delta \bar{p}_{Base}^{CRF} + \left(R_{CRF}^{Tool}\right)^{-1} \left(\left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)^{-1} - I_3\right) \bar{p}_{CRF}^{Tool}$$

$$\delta \bar{p}_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 - \hat{\alpha}_{CRF}^{Base}\right) R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} - \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 - \hat{\alpha}_{CRF}^{Base}\right) \delta \bar{p}_{Base}^{CRF} + \left(R_{CRF}^{Tool}\right)^{-1} \left(-\hat{\alpha}_{CRF}^{Base}\right) \bar{p}_{CRF}^{Tool}$$

$$\delta \bar{p}_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} - \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base} R_{CRF}^{Tool} \delta \bar{p}_{Base}^{Tool} - \left(R_{CRF}^{Tool}\right)^{-1} \delta \bar{p}_{Base}^{CRF} + \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base} \delta \bar{p}_{Base}^{CRF} - \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base} \bar{p}_{CRF}^{Tool}$$

$$\delta \bar{p}_{CRF}^{Tool} \approx \delta \bar{p}_{Base}^{Tool} + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \delta \bar{p}_{Base}^{Tool} + \left(R_{CRF}^{Tool}\right)^T \left(-\delta \bar{p}_{Base}^{CRF} + \hat{\alpha}_{CRF}^{Base} \delta \bar{p}_{Base}^{CRF} - \hat{\alpha}_{CRF}^{Base} \bar{p}_{CRF}^{Tool}\right)$$

### Linearizing the Orientation Error:

$$\Delta R_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)^{-1} R_{CRF}^{Tool} \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)$$

$$\Delta R_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} \left(I_3 - \hat{\alpha}_{CRF}^{Base}\right) R_{CRF}^{Tool} \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)$$

$$\Delta R_{CRF}^{Tool} \approx \left(\left(R_{CRF}^{Tool}\right)^{-1} - \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base}\right) \left(R_{CRF}^{Tool} + R_{CRF}^{Tool} \hat{\alpha}_{CRF}^{Base}\right)$$

$$\Delta R_{CRF}^{Tool} \approx \left(R_{CRF}^{Tool}\right)^{-1} R_{CRF}^{Tool} + \left(R_{CRF}^{Tool}\right)^{-1} R_{CRF}^{Tool} \hat{\alpha}_{CRF}^{Base} - \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base} R_{CRF}^{Tool} - \left(R_{CRF}^{Tool}\right)^{-1} \hat{\alpha}_{CRF}^{Base} R_{CRF}^{Tool} \hat{\alpha}_{CRF}^{Base}$$

Using the identity:  $R \hat{\alpha} R^T = skew(R \bar{\alpha})$

$$\Delta R_{CRF}^{Tool} \approx I_3 + \hat{\alpha}_{CRF}^{Base} + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \hat{\alpha}_{CRF}^{Base}$$

$$\Delta R_{CRF}^{Tool} \approx \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right) + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)$$

$$\Delta R_{CRF}^{Tool} \approx \left(I_3 + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right)\right) \left(I_3 + \hat{\alpha}_{CRF}^{Base}\right)$$

Where both factors are linearized approximations of rotation matrices of the form:

$$\Delta R \approx I_3 + skew(\bar{\alpha})$$

Now that we have equations for the error tool registration error that are linear in terms of the individual frame errors, the logical next step is to place bounds on the magnitudes of the position and orientation registration error.

**Bounds on Position Error:**  $\delta \bar{p}_{CRF}^{Tool}$

Taking the norm of both sides of the equation:

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| = \left\| \delta \bar{p}_{Base}^{Tool} + skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \delta \bar{p}_{Base}^{Tool} + \left(R_{CRF}^{Tool}\right)^T \left(-\delta \bar{p}_{Base}^{CRF} + \hat{\alpha}_{CRF}^{Base} \delta \bar{p}_{Base}^{CRF} - \hat{\alpha}_{CRF}^{Base} \bar{p}_{CRF}^{Tool}\right) \right\|$$

Using the triangle inequality to put a bound on the error magnitude:

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \left\| \delta \bar{p}_{Base}^{Tool} \right\| + \left\| skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \delta \bar{p}_{Base}^{Tool} \right\| + \left\| \left(R_{CRF}^{Tool}\right)^T \left(-\delta \bar{p}_{Base}^{CRF} + \hat{\alpha}_{CRF}^{Base} \delta \bar{p}_{Base}^{CRF} - \hat{\alpha}_{CRF}^{Base} \bar{p}_{CRF}^{Tool}\right) \right\|$$

Since length is preserved in rigid body rotation, the rotation can be removed:

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \left\| \delta \bar{p}_{Base}^{Tool} \right\| + \left\| skew\left(R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base}\right) \delta \bar{p}_{Base}^{Tool} \right\| + \left\| \left(-\delta \bar{p}_{Base}^{CRF} + \hat{\alpha}_{CRF}^{Base} \delta \bar{p}_{Base}^{CRF} - \hat{\alpha}_{CRF}^{Base} \bar{p}_{CRF}^{Tool}\right) \right\|$$

Since multiplying a small displacement by a skew matrix generated from the approximation of a small rotation produces almost negligible values, these components can be removed as well:

$$\boxed{\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \left\| \delta \bar{p}_{Base}^{Tool} \right\| + \left\| \delta \bar{p}_{Base}^{CRF} \right\|}$$

Therefore, we end up with this bound on the translational error of the transformation between the CRF and tool frames. Notice that the bound is a function only of the errors of the measured frames themselves. This is a worst-case scenario, and it may result in a severe overestimation of the error.

**Bounds on Orientation Error:**  $\delta\theta_{CRF}^{Tool} = \left\| \bar{\alpha}_{CRF}^{Tool} \right\|$

From before we have:

$$\Delta R_{CRF}^{Tool} \approx \left( I_3 + skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) \right) \left( I_3 + \hat{\alpha}_{Base}^{Tool} \right)$$

Isolating the error term:

$$I_3 + \hat{\alpha}_{CRF}^{Tool} \approx I_3 + skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) + \hat{\alpha}_{Base}^{Tool} + skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) \hat{\alpha}_{Base}^{Tool}$$

$$\hat{\alpha}_{CRF}^{Tool} \approx skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) + \hat{\alpha}_{Base}^{Tool} + skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) \hat{\alpha}_{Base}^{Tool}$$

Removing higher order error terms as before:

$$\hat{\alpha}_{CRF}^{Tool} \approx skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) + \hat{\alpha}_{Base}^{Tool}$$

Converting the skew symmetric matrices into the associated Rodriguez vectors:

$$\left[ \hat{\alpha}_{CRF}^{Tool} \right]^\vee \approx \left[ skew\left( R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right) + \hat{\alpha}_{Base}^{Tool} \right]^\vee$$

$$\bar{\alpha}_{CRF}^{Tool} \approx R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} + \bar{\alpha}_{Base}^{Tool}$$

Note: the ‘ $\vee$ ’ operator turns a skew-symmetric matrix  $\in so(3)$  into its corresponding vector  $\in \mathbb{R}^3$ . This is the same as the  $vec(\cdot)$  operator in other notation.

Taking the norm of both sides:

$$\left\| \bar{\alpha}_{CRF}^{Tool} \right\| \approx \left\| R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} + \bar{\alpha}_{Base}^{Tool} \right\|$$

Using the triangle inequality to get a bound on the error:

$$\left\| \bar{\alpha}_{CRF}^{Tool} \right\| \leq \left\| R_{CRF}^{Tool} \bar{\alpha}_{CRF}^{Base} \right\| + \left\| \bar{\alpha}_{Base}^{Tool} \right\|$$

Since rigid body rotations preserve length, the rotation matrix can be removed without affecting the magnitude:

$$\left\| \bar{\alpha}_{CRF}^{Tool} \right\| \leq \left\| \bar{\alpha}_{CRF}^{Base} \right\| + \left\| \bar{\alpha}_{Base}^{Tool} \right\|$$

Plugging in the rotation angles associated with the Rodriguez vectors, we get a bound on the angular error:

$$\boxed{\left| \delta\theta_{CRF}^{Tool} \right| \leq \left| \delta\theta_{CRF}^{Base} \right| + \left| \delta\theta_{Base}^{Tool} \right|}$$

Therefore, for a given magnitude of angular error for the tool and CRF frames, we can calculate bounds on the relative transformation’s rotational error. As before, this is a worst-case scenario may result in a severe overestimation of the error.

## Tool Tracking Uncertainty

### Covariance Propagation Overview

The previous analysis puts bounds of the errors, but says nothing about the statistical distribution of error. The following analysis details how error covariance propagates through the frame transformations. Another nice reference for error propagation through frame transformations by another member of our lab is [Boztec, 2005]. First we define the notation to be used and the available operations. Then this will be applied to the specific problem.

#### Representation of Frames:

$\hat{F}_i = (F_i, W_i)$  is the  $i^{\text{th}}$  rigid body frame with associated covariance

$$\text{Where: } F_i = \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix} \in SE(3) \mapsto f_i = \begin{bmatrix} \bar{r}_i \\ \bar{p}_i \end{bmatrix} \in \mathbb{R}^6 = (\bar{r}_i, \bar{p}_i) \text{ is the nominal frame}$$

Where:  $R_i \mapsto \bar{r}_i = \theta_i \bar{n}_i \in \mathbb{R}^3$  is the appropriate Rodriguez vector  
 $\bar{n}_i \in \mathbb{R}^3$  is the unit vector along the axis of rotation of  $R_i$   
 $\theta_i$  is the angle of rotation about that axis (radians)

$W_i$  is the covariance matrix representing the uncertainty of the frame

#### Covariance of a Frame:

The covariance is split into two independent components, one for position and one for orientation. Therefore, it is represented as a block diagonal combination of two separate sub-matrices.

$$W_i = \begin{bmatrix} W_{r_i} & 0 \\ 0 & W_{p_i} \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

Where:  $W_{r_i} = E\left((\hat{r}_i - \bar{r}_i)(\hat{r}_i - \bar{r}_i)^T\right) \in \mathbb{R}^{3 \times 3}$  is the covariance of the orientation

$W_{p_i} = E\left((\hat{p}_i - \bar{p}_i)(\hat{p}_i - \bar{p}_i)^T\right) \in \mathbb{R}^{3 \times 3}$  is the covariance of the position

The standard deviations for each component of a frame are plugged into the covariance equation. The vector of standard deviations is as follows:

$$\bar{\sigma} = \left[ \sigma_{r_x} \quad \sigma_{r_y} \quad \sigma_{r_z} \quad \sigma_{p_x} \quad \sigma_{p_y} \quad \sigma_{p_z} \right]^T \in \mathbb{R}^6$$

This results in a matrix with the appropriate variances along the diagonal of  $W_i$ :

$$tr(W_i) = \bar{\sigma}_i \bullet \bar{\sigma}_i$$

Note that the standard deviations of the elements of  $\bar{p}$  are straightforward, but that the standard deviations of the elements of  $\bar{r}$  are a little more difficult to interpret. They represent the uncertainty in the Rodriguez vector, which is related to the uncertainty in angle as follows:

$$\Delta r = |\theta| \sin(\Delta\theta) \approx |\theta| \Delta\theta$$

Therefore, the standard deviation of the rotation error is similarly related. Both the Rodriguez vector and the angular uncertainty can be broken into components to solve for  $\bar{\sigma}$ . If the error is isotropic, then all of the components for each of the position error and orientation error will be the same.

### Definition of Operations:

The operator ‘ $\oplus$ ’ combines two frames of given uncertainties. The two nominal frames are combined through normal matrix multiplication of two matrices  $F_1, F_2 \in \mathbb{R}^{4 \times 4}$ . This operation also includes to combinations of two covariance matrices  $W_1, W_2 \in \mathbb{R}^{6 \times 6}$ , they are combined using Jacobian methods introduced later.

The operator ‘ $\circ$ ’ combines two Rodriguez vectors,  $\bar{r}_1$  and  $\bar{r}_2$ , by converting them into rotation matrices  $R_1, R_2 \in \mathbb{R}^{3 \times 3}$ , performing matrix multiplication, and converting the product back into a Rodriguez vector  $\bar{r} \in \mathbb{R}^3$ .

The operator ‘ $\bullet$ ’ multiplies a position vector,  $\bar{p}$ , by the rotation matrix corresponding to the given Rodriguez vector,  $\bar{r}$ . The resultant is a position vector  $\bar{p} \in \mathbb{R}^3$ .

### Compounding Frames:

$$\hat{F}_3 = \hat{F}_2 \oplus \hat{F}_1 \mapsto f_3 = f_2 \oplus f_1$$

Nominal Frame:

$$F_3 = F_2 F_1 = \begin{bmatrix} R_2 R_1 & R_2 \bar{p}_1 + \bar{p}_2 \\ 0 & 1 \end{bmatrix} \mapsto f_3 = (\bar{r}_2 \circ \bar{r}_1, \bar{r}_2 \bullet \bar{p}_1 + \bar{p}_2)$$

## Compounding of Covariances:

As is presented in [Penec, 1995], covariances are propagated through frame transformations through the use of Jacobian matrices as follows:

$$W_3 = J_1 W_1 J_1^T + J_2 W_2 J_2^T$$

Where:  $W_i \in \mathbb{R}^{6 \times 6}$  is the covariance matrix for the  $i^{\text{th}}$  transformation

In general, Jacobians are defined as follows:

$$J = \frac{\partial h(x)}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial h_n}{\partial x_1} & \dots & \frac{\partial h_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Working out the Jacobians for this particular case:

$$J_1 = \frac{\partial f_3}{\partial f_1} = \frac{\partial(\bar{r}_2 \circ \bar{r}_1, \bar{r}_2 \bullet \bar{t}_1 + \bar{t}_2)}{\partial(\bar{r}_1, \bar{t}_1)} = \begin{bmatrix} \frac{\partial(\bar{r}_2 \circ \bar{r}_1)}{\partial \bar{r}_1} & 0 \\ 0 & R_2 \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

$$J_2 = \frac{\partial f_3}{\partial f_2} = \frac{\partial(\bar{r}_2 \circ \bar{r}_1, \bar{r}_2 \bullet \bar{t}_1 + \bar{t}_2)}{\partial(\bar{r}_2, \bar{t}_2)} = \begin{bmatrix} \frac{\partial(\bar{r}_2 \circ \bar{r}_1)}{\partial \bar{r}_2} & 0 \\ \frac{\partial(\bar{r}_2 \bullet \bar{t}_1)}{\partial \bar{r}_2} & I_3 \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

## Frame Inversion:

Inverting a frame transformation using this notation requires not only inverting the nominal frame, but also the corresponding covariance matrix as follows.

$$(\hat{F})^{-1} = (F, W)^{-1} = (F^{-1}, J_1 W J_1^T)$$

The nominal frame inversions is rather straightforward:

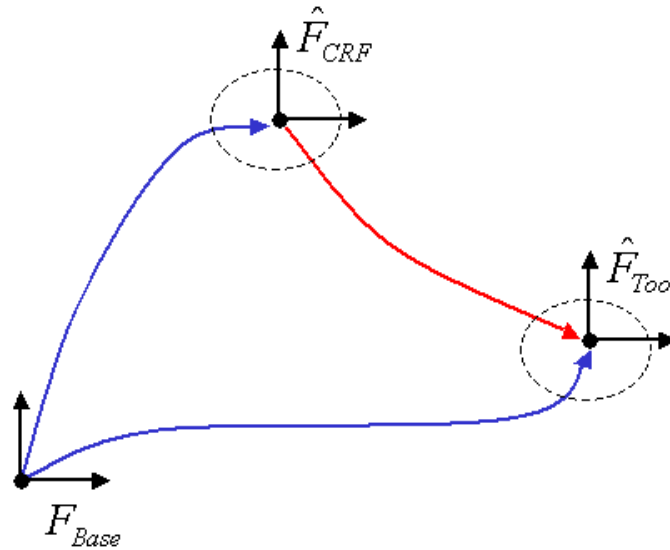
$$F^{-1} \mapsto f^{(-1)} = (\bar{r}, \bar{p})^{(-1)} = (\bar{r}^{(-1)}, \bar{r}^{(-1)} \bullet (-\bar{p})) = (-\bar{r}, -\bar{r} \bullet -\bar{p})$$

As with frame transformations, we use Jacobians to invert the covariance matrix. As presented in [Penec, 1995], the Jacobian to perform this operation is as follows:

$$J_1 = \begin{bmatrix} I_3 & 0 \\ \frac{\partial(\bar{r}^{(-1)} \bullet (-\bar{p}))}{\partial \bar{r}^{(-1)}} & R^{-1} \end{bmatrix} = \begin{bmatrix} I_3 & 0 \\ -\frac{\partial(\bar{r} \bullet \bar{p})}{\partial \bar{r}} & R^T \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

## Tool Tracking Covariance Propagation

The previous description shows in general how uncertainty propagates through frame transformations. We will now apply this to our particular case of tracking a tool with respect to a reference frame where each rigid body has its own covariance matrix,  $W_i$ , representing its measurement uncertainty. Figure 4-3 shows the frames involved, where the dashed lines represent uncertainty in the measurements of  $\hat{F}_{CRF}$  and  $\hat{F}_{Tool}$ , which are the 6 DoF measurements of the CRF and tool frames respectively with respect to the base frame.



Measured frame transformations with given uncertainties  
**Figure 4-3**

Due to the uncertainty in measurements, each transformation has a nominal frame (the actual position and orientation), and an associated covariance that represents its uncertainty.

The following are the frames using the covariance representation:

$$\hat{F}_{CRF} = (F_{CRF}, W_{CRF})$$

$$\hat{F}_{Tool} = (F_{Tool}, W_{Tool})$$

Where:  $F_{CRF}$  and  $F_{Tool}$  represent the nominal/true transformations as before  
 $W_{CRF}$  and  $W_{Tool}$  represent associated covariance matrices (uncertainties)

The representation of the tool with respect to the reference is as follows:

$$\begin{aligned}\hat{F}_{CRF}^{Tool} &= \left( \hat{F}_{Base}^{CRF} \right)^{-1} \hat{F}_{Base}^{Tool} \\ \left( F_{CRF}^{Tool}, W_{CRF}^{Tool} \right) &= \left( F_{CRF}, W_{CRF} \right)^{-1} \left( F_{Tool}, W_{Tool} \right) \\ \left( F_{CRF}^{Tool}, W_{CRF}^{Tool} \right) &= \left( F_{CRF}^{-1}, J_I W_{CRF} J_I^T \right) \left( F_{Tool}, W_{Tool} \right)\end{aligned}$$

Where:

$$J_I = \begin{bmatrix} I_3 & 0 \\ \frac{-\partial(\bar{r}_{CRF} \bullet \bar{p}_{CRF})}{\partial \bar{r}_{CRF}} & R_{CRF}^T \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

Nominal Frame:

$$\begin{aligned}F_{CRF}^{Tool} &= \left( F_{Base}^{CRF} \right)^{-1} F_{Base}^{Tool} \mapsto f_{CRF}^{Tool} = f_{CRF^{(-1)}} f_{Tool} \\ f_{CRF}^{Tool} &= \left( \bar{r}_{CRF}^{Tool}, \bar{p}_{CRF}^{Tool} \right) = \left( \bar{r}_{CRF}^{(-1)}, \bar{r}_{CRF}^{(-1)} \bullet (-\bar{p}_{CRF}) \right) \left( \bar{r}_{Tool}, \bar{p}_{Tool} \right) \\ &= \left( \bar{r}_{CRF}^{(-1)} \circ \bar{r}_{Tool}, \bar{r}_{CRF}^{(-1)} \bullet \bar{p}_{Tool} + \bar{r}_{CRF}^{(-1)} \bullet (-\bar{p}_{CRF}) \right)\end{aligned}$$

Uncertainty Calculation:

$$W_{CRF}^{Tool} = J_{CRF^{(-1)}} \left( J_I W_{CRF} J_I^T \right) J_{CRF^{(-1)}}^T + J_{Tool} W_{Tool} J_{Tool}^T$$

Where:

$$\begin{aligned}J_{Tool} &= \frac{\partial f_{CRF}^{Tool}}{\partial f_{Tool}} = \frac{\partial \left( \bar{r}_{CRF}^{(-1)} \circ \bar{r}_{Tool}, \bar{r}_{CRF}^{(-1)} \bullet \bar{p}_{Tool} + \bar{r}_{CRF}^{(-1)} \bullet (-\bar{p}_{CRF}) \right)}{\partial (\bar{r}_{Tool}, \bar{p}_{Tool})} \\ J_{CRF^{(-1)}} &= \frac{\partial f_{CRF}^{Tool}}{\partial f_{CRF^{(-1)}}} = \frac{\partial \left( \bar{r}_{CRF}^{(-1)} \circ \bar{r}_{Tool}, \bar{r}_{CRF}^{(-1)} \bullet \bar{p}_{Tool} + \bar{r}_{CRF}^{(-1)} \bullet (-\bar{p}_{CRF}) \right)}{\partial \left( \bar{r}_{CRF}^{(-1)}, \bar{r}_{CRF}^{(-1)} \bullet (-\bar{p}_{CRF}) \right)} \\ J_I &= \begin{bmatrix} I_3 & 0 \\ \frac{-\partial(\bar{r}_{CRF} \bullet \bar{p}_{CRF})}{\partial \bar{r}_{CRF}} & R_{CRF}^T \end{bmatrix} \text{ (as before)}\end{aligned}$$



## **Frame Fitting**

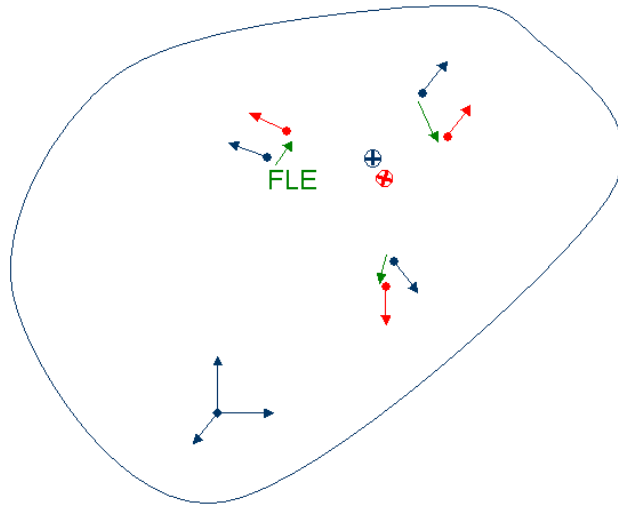
The analysis above describes how error propagates through frame transformations and predicts tool registration error with respect to a reference. However, this result is a function of the error introduced in fitting those frames to the measurements. Therefore, in order to determine the tool registration error, we must investigate the uncertainty in fitting 6 DoF frames to measurement data. This fitting is some type of optimal fit of the rigid body frame to the individual sensors. Often this is performed internally inside of the tracker's system control unit (SCU); other times it is done by custom user software from individual sensor measurements.

In what is referred to as the Procrustes problem, the goal is to align a cloud of points  $\in \mathbb{R}^3$  to a second cloud of points through a rigid transformation  $\in SE(3)$ . In the ideal case, this rigid transformation will align the two sets such that each point is coincident with its corresponding point in the other set. However, both sets of points may be susceptible to errors that can bias the best-fit frame from the nominal frame.

In the general tracking scenario, there is a tool configuration that describes the location of points on the tool with respect to its origin, and this configuration is often known accurately. Measurements of the points / sensors on the tool are taken, and the best rigid transformation that aligns the corresponding points in the tool configuration to the measurements is used to describe the tool frame; this is the registration step. However, due to inaccuracies in the measurement device that results in errors of locating the points on the tool, this alignment will not be perfect; therefore, a best estimate of the transformation must be determined. A common method is to find the rigid transformation that aligns the corresponding points with the minimal error in the least squares sense is presented in [Arun, 1987].

In the case of the Aurora system, the problem is somewhat different. This is because we are receiving 5 DoF information from each sensor, not just 3 DoF position data from markers as detected by an OTS. Instead of only having point correspondences, we also have an equal number of orientation vectors to align. This extra information has no effect on the alignment of the centroids of the two data sets; where it comes into play is in determining the optimal rigid rotation. This rigid rotation is about the centroid/center of gravity (C.G.) of the sensor positions that best aligns the measurements. Here, we treat the orientation vectors as if they were points along the line created by placing the orientation vectors at the C.G. and projecting out a given distance along that line. This distance is the relative weighting factor of the orientation data to the position data; its value is determined based upon the system accuracy as described later in the Error Weight Calculation section. In addition, individual weighting factors could be used on each point if it is known that some measurements can be trusted more than others.

The alignment scenario is demonstrated in Figure 4-4. The original frame is shown in blue; its sensors are perturbed in both position and orientation, shown in red. This perturbation represents measurement error of the sensor and is often termed the fiducial localization error (FLE); however, note that it is not only a translation error in this case.

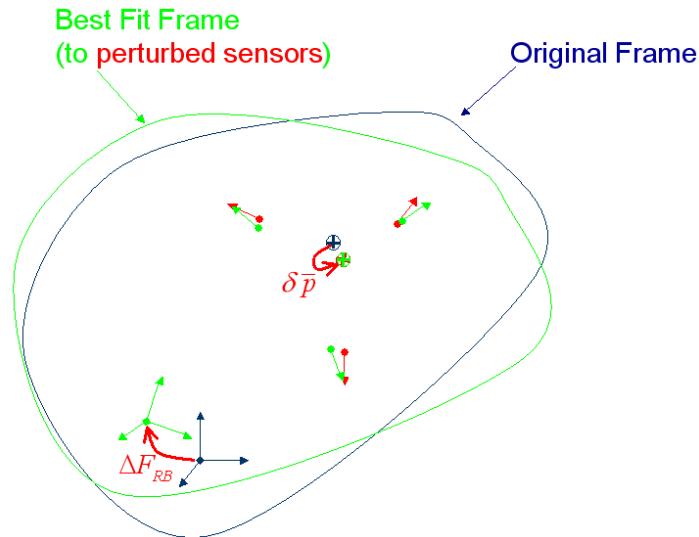


Representation of measurements error

(Blue represents actual sensor configuration, Red represents perturbed measurements)

**Figure 4-4**

The rigid body with its known configuration is then transformed with the best-fit frame that aligns its sensors to the measurements. From Figure 4-5, shown in green, it is first aligned to the centroid of the perturbed points, and then rotated such that it has a fit that is optimal in some sense.



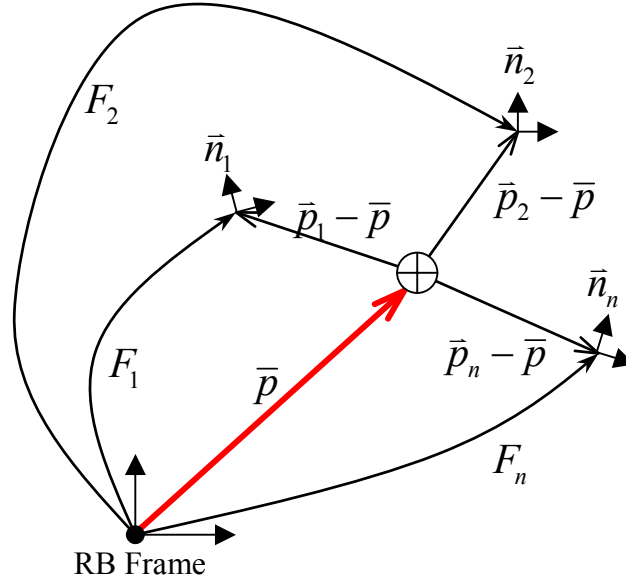
Representation of best fit frame (green) to perturbed measurements (red)

(Original frame is shown in blue,  $\Delta F_{RB}$  is the induced alignment error)

**Figure 4-5**

**Notation:**

The frame of a rigid body (RB) is calculated based on the least squares best fit of the position and orientation of an a-priori known configuration to the measured sensors. Figure 4-6 shows the typical configuration of sensors in an RB's frame.



Arrangement of sensor positions and orientations in the RB

**Figure 4-6**

The frames  $\{F_1, F_2, \dots, F_n\}$  represent the known position and orientation of  $n$  sensors with respect to the given rigid body frame of reference (i.e. a frame centered at a tool tip and aligned with a pointer shaft).

Since the sensor frames are only specified in 5 DoF, they can be represented as:

$$F_i = (\bar{n}_i, \bar{p}_i)$$

Where:  $\bar{p}_i \in \mathbb{R}^3$  is the position of the sensor wrt the RB frame

$\bar{n}_i = \bar{r}_z \in \mathbb{R}^3$  of  $R_i = \begin{bmatrix} \bar{r}_x & \bar{r}_y & \bar{r}_z \end{bmatrix} \in SO(3)$  is the orientation of the sensor with respect to the RB frame

The de-meanned values of the sensor locations are necessary in order to compute the optimal rotational alignment of a rigid body to its measurements. These values are the original values of the sensor positions wrt the RB frame with the location of the centroid (C.G.) in the RB frame subtracted off. Therefore, first the C.G. of the points with respect to the RB frame is computed as follows.

Center of gravity (C.G.) of the sensor positions wrt RB frame:

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n \bar{p}_i \in \mathbb{R}^3$$

## Least Squares Solution

Following are details of the method used for solving for the best rigid point cloud to point cloud rotation problem in the LS sense. The weighted orientations are treated the same way as the demeaned points; the weighting factor,  $w$ , keeps the position and orientation contributions balanced. Methods for solving problems of this type are detailed in great depth in several papers as described in the Literature Review section of Chapter 1. The method used here is a modified version of that presented in [Arun, 1987], with the modification being the addition of orientations as mentioned above.

From before:

$$X = \left[ \begin{array}{c} (\bar{p}_{RB_1} - \bar{p}_{RB})^T \\ \vdots \\ (\bar{p}_{RB_n} - \bar{p}_{RB})^T \\ \hline w \left( \begin{array}{c} \bar{n}_{RB_1}^T \\ \vdots \\ \bar{n}_{RB_n}^T \end{array} \right) \end{array} \right] \quad Y = \left[ \begin{array}{c} (\bar{p}_{Meas_1} - \bar{p}_{Meas})^T \\ \vdots \\ (\bar{p}_{Meas_n} - \bar{p}_{Meas})^T \\ \hline w \left( \begin{array}{c} \bar{n}_{Meas_1}^T \\ \vdots \\ \bar{n}_{Meas_n}^T \end{array} \right) \end{array} \right]$$

Where:  $\bar{p}$  represents the mean position of the given measurement set

$\bar{n}$  represents the unit vector pointing along the  $z$ -axis of the given sensor

$w$  represents the weighting of the orientation measurements with respect to the position measurements. The value for this is determined as shown in the Error Weight Calculation section of Chapter 5.

The subscripts  $RB$  and  $Meas$  represent the sensors in the known rigid body configuration and the actual sensor measurements respectively.

Using the notation in [Arun, 1987]:

$$\begin{aligned} q_i &= X(i, 1:3)^T \\ q'_i &= Y(i, 1:3)^T \end{aligned} \quad i = \{1, \dots, n\}$$

$$H = \sum_{i=1}^n q_i q_i'^T$$

Take the Singular Value Decomposition (SVD) of  $H$ :

$$H = U\Lambda V^T$$

Calculate the optimal rotation,  $R$ , in the Least Squares (LS) sense as follows:

$$R = \tilde{R} = VU^T$$

Verify the solution:

$$\det(\tilde{R}) = \begin{cases} +1 & \text{Set: } R = \tilde{R} \\ -1 & \text{Algorithm fails} \end{cases}$$

A method for dealing with failure of the algorithm is proposed in [Umeyama, 1991].

Using the LS SVD solution for the optimal rotation,  $R$ , the optimal translation vector,  $\bar{v}$ , is calculated by rotating the vector to the C.G. of the RB points,  $\bar{p}_{RB}$ , by the rotation,  $R$ , and subtracting that from the C.G. of the sensor measurements. Note that the centroid is only dependent on the position information of the sensor measurements, the weighted orientations are not included.

The optimal translation that aligns the RB with the measurements is given by:

$$\bar{v} = \bar{p}_{Meas} - R * \bar{p}_{RB}$$

Combining the two components, the 6 DoF frame representing the location of the rigid body in base coordinates is:

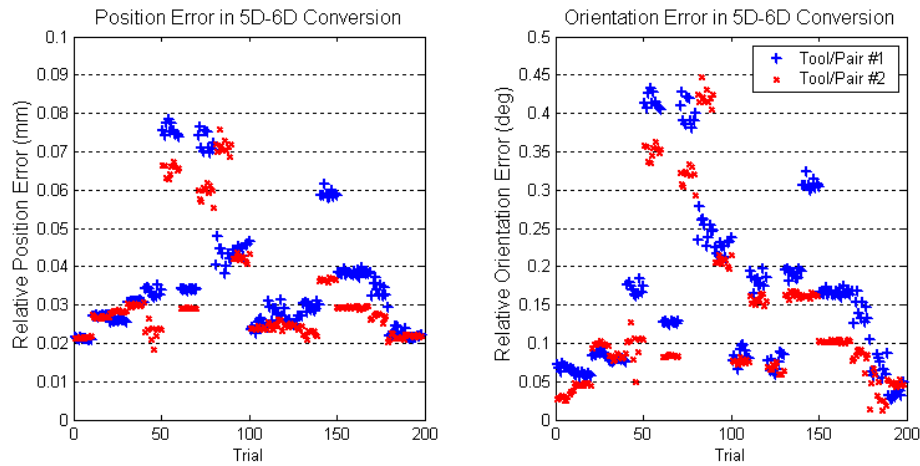
$$F_{RB \rightarrow Meas} = \begin{bmatrix} R & \bar{v} \\ 0 & 1 \end{bmatrix} \in SE(3)$$

Where:  $R$  is the best rigid rotation (from above )

$\bar{v}$  is the translation that aligns the centroids

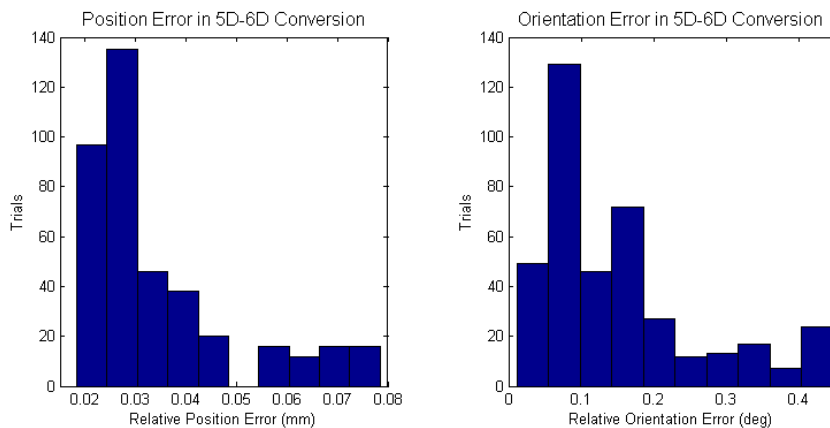
## Validation of 5D to 6D Conversion

Since the accuracy of the frame-fitting conversion is absolutely mission-critical, it is important to verify my algorithm. If we take the Aurora system itself with its internal firmware calculations as the baseline measure, we can compare how well our algorithm performs the 5D to 6D conversion. For this trial I used a plastic test tool with two pairs of sensors in accurately known locations and orientation. The tool was moved to 20 poses where 10 static measurements were made of all four sensors at each pose. The data was collected in the raw format and allowed post processing by both my algorithm and the NDI algorithm that is used in the system firmware. The error is judged as the relative difference in position and orientation of the 6 DoF frame generated from each pair of two 5 DoF sensors. In total, we have a total of 200 frames for each of two 6 DoF tools. Frames were fit using an orientation vs. position weight of  $w=100$ . The results are presented in Figure 4-7. For the distribution of error magnitude, see Figure 4-8.



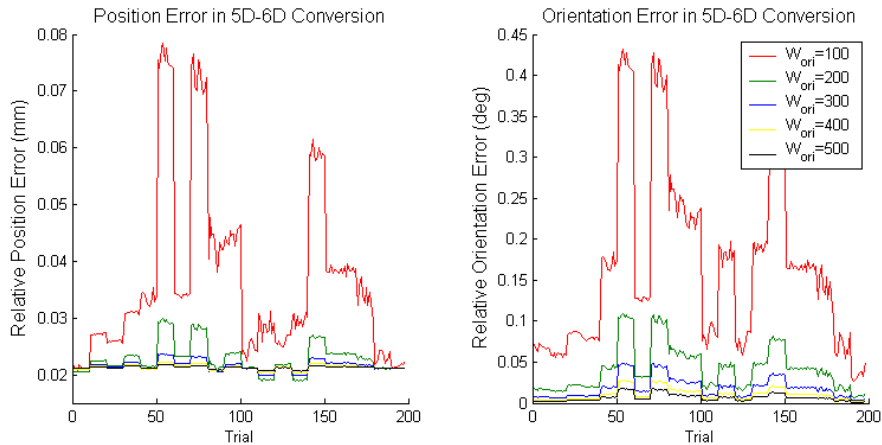
Error between my frame calculation NDI firmware calculation  
(using default orientation weighting, for two sensors)

**Figure 4-7**



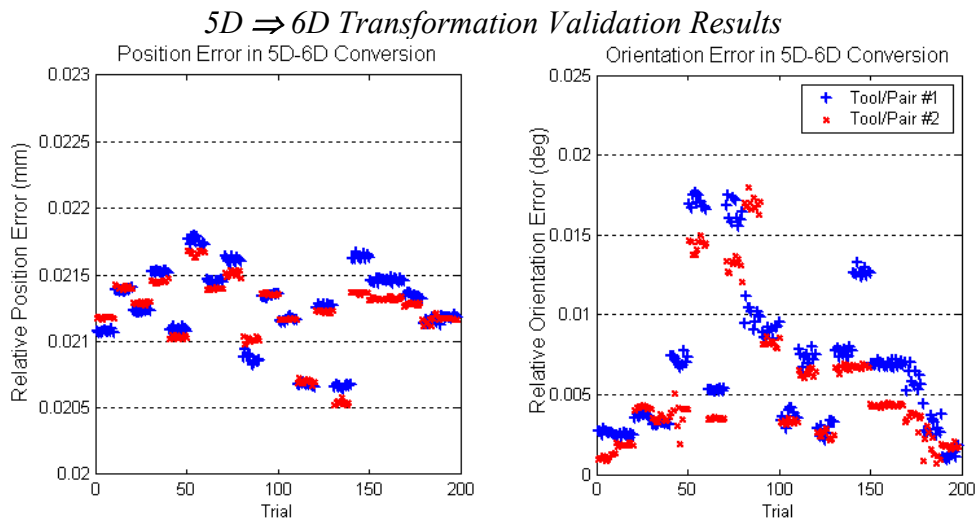
Distribution of difference my frame fitting algorithm and NDI frame fitting algorithm  
**Figure 4-8**

In order to see what the effect of the weighting factor between the orientation and position contributions has, I did a trial where the weight was varied. In this trial the weight of orientation contribution with respect to position in the 5D to 6D conversion was adjusted from 100 to 500. From looking at Figure 4-9, it is clear that the NDI frame fitting algorithm puts a very high weight on the contribution of sensor orientations to the best-fit frame as compared with sensor positions.



Difference in fitted frames (Mine & NDI) vs. the orientation contribution weight  
**Figure 4-9**

The key result in this section is shown in Figure 4-10. Here, I repeat the earlier trial, but with an orientation weighting factor of  $w=500$ . Notice that the axes are different than before in Figure 4-7; the deviation between the frames fit using the two algorithms is now one order of magnitude smaller. Be aware that these results do not indicate which algorithm performs a ‘better’ fit; they represent how close the fitted 6 DoF frames from two 5 DoF frames are for my algorithm and the one running in the NDI Aurora firmware.



Difference between my fitted frames and the NDI fitted frames  
with optimized position vs. orientation weighting

**Figure 4-10**

## Bounds on Frame Fitting

We now have bounds on the registration error of the tool with respect to the coordinate reference frame and a method for fitting a tool frame to the sensor measurements; the next step is determining what the measurement error of these frames is. First, we investigate the bounds on translation error, and then we investigate the bounds on rotation error.

### Translation Error

The translational error of the rigid body frame comes from two components: 1) one associated with the alignment of the centroid of the group of sensors, and 2) one associated with the rotation-induced error. If we make the assumption that the tool configuration is known precisely, then the error in locating the centroid is only associated with the measurements of the sensors associated with the RB.

The translational measurement error for each sensor,  $i$ , is as follows:

$$\hat{p}_i = \bar{p}_i + \delta \bar{p}_i$$

Where:  $\hat{p}$  is the disturbed measurement with respect to the RB frame  
 $\bar{p}$  is the nominal/actual position  
 $\delta \bar{p}$  is the translational measurement error

Returning to the definition of  $\bar{p}$ , now we must determine the value of  $\delta \bar{p}$

$$\hat{\bar{p}} = \bar{p} + \delta \bar{p} = \frac{1}{n} \sum_{i=1}^n (\bar{p}_i + \delta \bar{p}_i)$$

Where:  $\hat{\bar{p}}$  is the location of the centroid (C.G) of the measured sensor positions  
 $\bar{p}$  is the nominal/actual location of the tool's C.G. in its own frame  
 $\delta \bar{p}$  is the error in the position of the centroid of the sensors

With some manipulation, we can get the following equations:

$$\bar{p} + \delta \bar{p} = \frac{1}{n} \left( \sum_{i=1}^n \bar{p}_i + \sum_{i=1}^n \delta \bar{p}_i \right)$$

$$\bar{p} + \delta \bar{p} = \bar{p} + \frac{1}{n} \left( \sum_{i=1}^n \delta \bar{p}_i \right)$$



Solving for the error in centroid location, we notice that is only a function of the translational measurement distortion terms:

$$\delta \bar{p} = \frac{1}{n} \sum_{i=1}^n \delta \bar{p}_i$$

Taking the magnitude of the error:

$$\|\delta \bar{p}\| = \left\| \frac{1}{n} \sum_{i=1}^n \delta \bar{p}_i \right\|$$

The bounds on this components of the position error in locating the rigid body can be found by applying the triangle inequality:

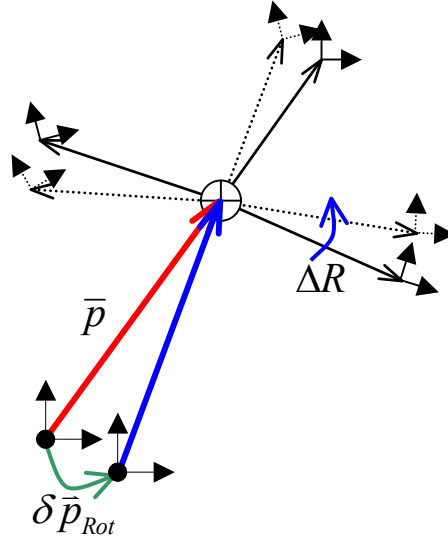
$$\boxed{\|\delta \bar{p}_{Trans}\| \triangleq \|\delta \bar{p}\| \leq \frac{1}{n} \sum_{i=1}^n \|\delta \bar{p}_i\|}$$

Thus, the error is bounded by the average magnitude of the sensor position errors.

The following result is a very large, but easily calculated upper bound on this error. This also applies to the worst case scenario, where there is a systematic translation of all of the points.

$$\|\delta \bar{p}_{Trans}\| = \|\delta \bar{p}\| \leq \max_i \|\delta \bar{p}_i\|$$

The second component of the position error in locating the RB frame comes from the rotation error. If the RB frame's origin lies at the C.G., then this component is negligible. Otherwise the rotation error causes a translational error of the tool's frame. Figure 4-11 shows how the rotation error causes a shift in tool frame position in addition to a rotation.



Rotation-induced position error of RB frame

Figure 4-11

### Representation of Rotations:

The rotation error is expanded as before, but instead of being linearized, the infinite series is replaced with a closed form solution. Also, the axis and angle of the rotation error are no longer combined into a single term for reasons that will be clear later.

$$\Delta R = \Delta R(\bar{\omega}\delta\theta) = \exp(\hat{\omega}\delta\theta) \in SO(3)$$

Where:  $\bar{\omega} \in \mathbb{R}^3$  is the unit vector along the axis of rotation

$\delta\theta$  is the angle of rotation in radians

$$\hat{\omega} = skew(\bar{\omega}) \in so(3), \quad \|\hat{\omega}\| = 1$$

Using the infinite series expansion of a matrix exponential to expand the rotation:

$$\Delta R = \exp(\hat{\omega}\delta\theta) = \sum_{j=0}^{\infty} \frac{1}{j!} \hat{\omega}^j (\delta\theta)^j = I_3 + \hat{\omega}\delta\theta + \frac{1}{2!} \hat{\omega}^2 (\delta\theta)^2 + \frac{1}{3!} \hat{\omega}^3 (\delta\theta)^3 + \dots$$

It can be shown that this infinite series can be rewritten using Rodrigues' formula:

$$\Delta R = \exp(\hat{\omega}\delta\theta) = I_3 + \hat{\omega} \sin(\delta\theta) + \hat{\omega}^2 (1 - \cos(\delta\theta))$$

From the vectors shown in Figure 4-11 we can get the following:

$$-\bar{p} + \delta \bar{p}_{Rot} = -\Delta R \bar{p}$$

Rearranging the equation to isolate the error term:

$$\delta \bar{p}_{Rot} = -\Delta R \bar{p} + \bar{p}$$

$$\delta \bar{p}_{Rot} = (-\Delta R + I_3) \bar{p}$$

Plugging in the expansion of rotations from above:

$$\delta \bar{p}_{Rot} = \left( -\left[ I_3 + \hat{\omega} \sin(\delta\theta) + \hat{\omega}^2 (1 - \cos(\delta\theta)) \right] + I_3 \right) \bar{p}$$

$$\delta \bar{p}_{Rot} = -\left[ \hat{\omega} \sin(\delta\theta) + \hat{\omega}^2 (1 - \cos(\delta\theta)) \right] \bar{p}$$

$$\delta \bar{p}_{Rot} = -\left[ \sin(\delta\theta) \hat{\omega} \bar{p} + \hat{\omega}^2 \bar{p} - \cos(\delta\theta) \hat{\omega}^2 \bar{p} \right]$$

Placing bounds of the error

$$\|\delta \bar{p}_{Rot}\| = \left\| \sin(\delta\theta) \hat{\omega} \bar{p} + \hat{\omega}^2 \bar{p} - \cos(\delta\theta) \hat{\omega}^2 \bar{p} \right\|$$

For small magnitudes of rotation error,  $\delta\theta$ , the following linearization can be made:

$$\sin(\delta\theta) \approx \delta\theta$$

$$\cos(\delta\theta) \approx 1$$

Plugging in the linearization and canceling like terms:

$$\|\delta \bar{p}_{Rot}\| \approx \left\| \delta\theta \hat{\omega} \bar{p} + \cancel{\hat{\omega}^2 \bar{p}} - \cancel{\hat{\omega}^2 \bar{p}} \right\|$$

$$\|\delta \bar{p}_{Rot}\| \approx \left\| \delta\theta \hat{\omega} \bar{p} \right\|$$

For consistent norms, we can split up the RHS and form an upper bound on the error:

$$\|\delta \bar{p}_{Rot}\| \leq |\delta\theta| \|\hat{\omega}\| \|\bar{p}\|$$

Since  $\|\hat{\omega}\| = 1$ , it can be removed to form a bound on the rotation-induced position error.

This result is just what we would have expected using the previous linearization.

$$\boxed{\|\delta \bar{p}_{Rot}\| \leq |\delta\theta| \|\bar{p}\|}$$

Where:  $|\delta\theta|$  is the magnitude of the rotation error introduced from fitting the frame with the best rigid rotation. Calculation of this bound is detailed in the following section.

Combining the two components, the **total position error** of the frame is as follows:

$$\|\delta \bar{p}_{RB}\| \leq \|\delta \bar{p}_{Trans}\| + \|\delta \bar{p}_{Rot}\|$$

Where:  $\|\delta \bar{p}_{Trans}\| \leq \max_i \|\delta \bar{p}_i\|$

$$\|\delta \bar{p}_{Rot}\| \leq |\delta \theta| \|\bar{p}\|$$

Plugging in the individual components, we get a bound on the position error for the RB frame:

$$\|\delta \bar{p}_{RB}\| \leq \max_i \|\delta \bar{p}_i\| + |\delta \theta_{RB}| \|\bar{p}\|$$

Going back to the previous definition of the first component of translation error to get a more realistic bound on the position error:

$$\|\delta \bar{p}_{RB}\| \leq \frac{1}{n} \sum_{i=1}^n \|\delta \bar{p}_i\| + |\delta \theta_{RB}| \|\bar{p}\|$$

Using this equation, we can investigate how to minimize the tracking error of a rigid body's frame. To minimize the first term, we want to minimize the average localization error (FLE). If this is a random variable, then by increasing the number of sensors,  $n$ , the registration error will be decreased. To minimize the second term, the key is to keep the origin of the rigid body as close to the C.G as possible. In terms of a tool, this means keeping the sensors close to the tip; in terms of a reference frame, the sensor should be evenly distributed around the area of interest. The angular element of the frame fitting error is investigated next.

## Rotation Error

This section details the calculation of bounds on the rotation error of the frame fitting procedure. This is a more complex problem than that of position error because it relies on analysis of the least squares problem of determining the optimal rigid rotation to align the sensor configuration to the measurement set.

Using the same definition as earlier for the  $X$  and  $Y$  matrices that represent the data sets to be aligned (including both position and orientation information):

$$\begin{aligned} X &\in \mathbb{R}^{3 \times 2n} && \text{Rigid Body configuration} \\ Y &\in \mathbb{R}^{3 \times 2n} && \text{Associated measurement set} \end{aligned}$$

And, subsets of these matrices are defined as:

$$\begin{aligned} \bar{x}_i &\in \mathbb{R}^3 \text{ is the } i^{\text{th}} \text{ column of } X \\ \bar{y}_i &\in \mathbb{R}^3 \text{ is the } i^{\text{th}} \text{ column of } Y \end{aligned}$$

The associated perturbations are:

$$\begin{aligned} d\bar{x}_i &= \bar{x}_i - \tilde{x}_i \in \mathbb{R}^3 \\ d\bar{y}_i &= \bar{y}_i - \tilde{y}_i \in \mathbb{R}^3 \end{aligned}$$

Where:  $\tilde{x}_i$  and  $\tilde{y}_i$  are the nominal values

We define the following matrix to be used for this analysis:

$$A \triangleq YX^T = \sum_{i=1}^{2n} \bar{y}_i \bar{x}_i^T \in \mathbb{R}^{3 \times 3}$$

Performing the polar decomposition on  $A$ :

$$A = RS$$

$$\text{Where: } R = \left( A^* A \right)^{\frac{1}{2}} \quad \text{Orthogonal rotation matrix}$$

$$S = A \left( A^* A \right)^{-\frac{1}{2}} \quad \text{Symmetric matrix}$$

From [Dosrt, 2005], the Rodriguez vector denoting the induced:

$$d\bar{a} = R \left( \text{tr}(S)I - S \right)^{-1} R^T \left( \sum_i (R\bar{x}_i) \times d\bar{y}_i + (Rd\bar{x}_i) \times \bar{y}_i \right)$$

Where:  $\|d\bar{a}\|$  is the magnitude of rotation in radians about the axis  $d\bar{a}$

To clean up the equations, the following definition is used:

$$H \triangleq (\text{tr}(S)I - S)^{-1}$$

Substituting into previous equation:

$$d\bar{a} = RHR^T \left( \sum_i (R\bar{x}_i) \times d\bar{y}_i + (Rd\bar{x}_i) \times \bar{y}_i \right)$$

If we assume that the RB configuration is known accurately:

$$d\bar{x}_i = [0, 0, 0]^T \forall i$$

Therefore, the error equation simplifies to:

$$d\bar{a} = RHR^T \left( \sum_i (R\bar{x}_i) \times d\bar{y}_i \right)$$

If we perturb the measurements of a tool that was already aligned using the best rigid alignment, then there is zero nominal rotation. We will assume the nominal rotation is the identity matrix, ( $R = I_3$ ), to simplify the analysis.

Therefore, the simplified equation for orientation error boils down to:

$$d\bar{a} = H \left( \sum_i \bar{x}_i \times d\bar{y}_i \right)$$

To determine the magnitude of rotation error we take the magnitude of  $d\bar{a}$  :

$$|\delta\theta_{RB}| \triangleq \|d\bar{a}\| = \left\| H \left( \sum_i \bar{x}_i \times d\bar{y}_i \right) \right\|$$

For consistent norms, we get the following bound:

$$|\delta\theta_{RB}| \leq \|H\| \left\| \left( \sum_i \bar{x}_i \times d\bar{y}_i \right) \right\|$$

Using triangle inequality, the norm can be brought inside the sum:

$$|\delta\theta_{RB}| \leq \|H\| \sum_i \|\bar{x}_i \times d\bar{y}_i\|$$

Using the definition of the cross product:

$$|\delta\theta_{RB}| \leq \|H\| \sum_i \|\bar{x}_i\| \|d\bar{y}_i\| |\sin(\phi)|$$

Where:  $\phi$  is the angle between the vector from the origin to the point,  $\bar{x}_i$ , and the vector representing the measurement distortion,  $d\bar{y}_i$ .

Setting  $\phi = \pm\pi/2$  represents error normal to  $\bar{x}_i$ , which is the worst-case scenario. On the contrary, if the error were along this vector for a particular measurement, then there would be no induced rotation error.

For the worst case,  $\sin(\phi) = 1$ . This leaves the following equation:

$$|\delta\theta_{RB}| \leq \|H\| \sum_i \|\bar{x}_i\| \|d\bar{y}_i\|$$

Plugging in the system accuracy for the distortion term and putting in consistent notation:

$$\boxed{|\delta\theta_{RB}| \leq \|H_{RB}\| \sum_i \|\bar{p}_i\| \|\delta\bar{p}_i\|} \quad \text{OR} \quad |\delta\theta_{RB}| \leq \|H_{RB}\| [\|\bar{p}_1\|, \dots, \|\bar{p}_{2n}\|] [\|\delta\bar{p}_1\|, \dots, \|\delta\bar{p}_{2n}\|]^T$$

Where:  $\|\bar{p}_i\| = \|\bar{x}_i\|$

$$\|\delta\bar{p}_i\| = \|d\bar{y}_i\|$$

Note that for  $i = 1 \dots n$  this represents position error, and for  $i = n + 1 \dots 2n$  this represents orientation error from the definition of  $X$  and  $Y$ .

This is saying that:

1. We want to minimize the product of  $\|\bar{p}_i\| \|\delta\bar{p}_i\|$

Since we have no control over  $\|\delta\bar{p}_i\|$ , we could minimize this product by minimizing the distance from the origin of sensors prone to higher errors.

2. We want to minimize  $\|H\|$

$$\text{Where: } H = (\text{tr}(S)I - S)^{-1}$$

This boils down to:

$$\text{minimize}(\|H\|) \rightarrow \text{maximize}(\|\text{tr}(S)I - S\|) \rightarrow \begin{matrix} \text{maximize}(\|\text{tr}(S)\|) \\ \text{minimize}(\|S\|) \end{matrix}$$

If we assume that  $S$  is diagonal and we are using the Frobenius norm:

$$\text{tr}(S) = s_1 + s_2 + s_3 \quad \|S\| = \sqrt{s_1^2 + s_2^2 + s_3^2}$$

Therefore, we want large terms for  $s_1, s_2, s_3$  to maximize  $\text{tr}(S)$ , but evenly spread to minimize  $\|S\|$ . Ideally,  $s_1 = s_2 = s_3 = K$ , where  $K$  is large.

## Comparison to Physical Inertia Tensor:

In order to get a physical understanding for what is going on, we look a little closer at the equations derived above. By looking at the previous equations, it should be clear that what we are really trying to do to minimize tracking error of a rigid body is analogous to trying to increase the body's inertia. This is saying that the sensors positions and orientations act on the alignment of the RB at a given distance away from the centroid with a given mass.

The definition of a physical inertia tensor is:

$$\tilde{I} = \sum_{i=1}^n m_i (\bar{y}_i \bullet \bar{y}_i I_3 - \bar{y}_i \bar{y}_i^T)$$

From above analysis:

$$H = (tr(S)I_3 - S)^{-1}$$

Making the following definition:

$$\tilde{H} \triangleq H^{-1} = tr(S)I_3 - S$$

$$\text{Where: } A = YX^T = RS$$

To determine system inertia, assume no distortion or systematic rotation. This is valid because inertia magnitude is invariant of rotation and the change due to distortion is negligible.

Therefore, we set:

$$Y = X$$

$$R = I_3$$

Solving for  $S$ :

$$S = YY^T = \sum_{i=1}^{2n} \bar{y}_i \bar{y}_i^T$$

And the trace of  $S$ :

$$tr(S) = \sum_{i=1}^{2n} \bar{y}_i^T \bar{y}_i = \sum_{i=1}^{2n} \|\bar{y}_i\|^2$$



Plugging  $S$  and  $tr(S)$  into  $\tilde{H}$  :

$$\tilde{H} = \sum_{i=1}^{2n} \bar{y}_i^T \bar{y}_i I_3 - \sum_{i=1}^{2n} \bar{y}_i \bar{y}_i^T = \sum_{i=1}^{2n} (\bar{y}_i^T \bar{y}_i I_3 - \bar{y}_i \bar{y}_i^T)$$

Therefore,  $\tilde{H} = H^{-1}$  is in the form of a physical Inertia Tensor, less the mass term (i.e. unit mass is assumed for all points).

Further,  $\tilde{H}$  observes the definition of a  $2^{nd}$  order tensor in that it can be transformed as follows:

$$\tilde{H}' = R\tilde{H}R^T \quad \text{and} \quad \tilde{H}, \tilde{H}' \in \mathbb{R}^{3 \times 3}$$

Where:  $R \in SO(3)$

This representation for  $\tilde{H}$  is the equivalent of representing the inertia in the base frame as opposed to the body-fixed frame. Since for the perturbation analysis,  $R = I_3$ , choosing the appropriate frame to represent the inertia tensor in will not be an issue. Also,  $tr(S)$ , is invariant to rotation.

Compare  $\tilde{H}$  to an inertia tensor below:

$$\tilde{I} = \sum_{i=1}^n m_i (\bar{y}_i^T \bar{y}_i I_3 - \bar{y}_i \bar{y}_i^T) \quad \text{Inertia Tensor}$$

$$\tilde{H} = \sum_{i=1}^{2n} (\bar{y}_i^T \bar{y}_i I_3 - \bar{y}_i \bar{y}_i^T) \quad \text{Calculated Term}$$

(Note that they are identical except for the mass term)

This comparison allows us to make a physical analogy that may help in design of a tool. Essentially, the best tool design is equivalent to maximizing the inertia about all of the principal axes if the sensors and their orientations are assumed to be masses at a given distance and location from the centroid. Note that this is not the same as just increasing the magnitude of the inertia, as this will create a tool that is stable about one axis, but not the others. The key is to spread the inertia evenly among the principal axes.

## Uncertainty of Frame Fitting

Now that we have bounds on the frame fitting error, it is useful to determine the distribution of this error. Here, we determine what the covariance of the best-fit frame is for each RB. Again, this can be split into two components: position and orientation.

### Position Error Covariance of a Rigid Body Fit

The uncertainty in the translational component of the distortion is only dependent on the positional accuracy of the sensors. This measurement error is denoted:

$$\delta\bar{y} = [\sigma_x \quad \sigma_y \quad \sigma_z]^T \in \mathbb{R}^3$$

If the RB origin is at the CG [Dosrt, 2005]:

$$W_{Pos} = \frac{1}{n}(W_y + W_x) \in \mathbb{R}^{3 \times 3}$$

Since the sensor configuration is known accurately:

$$W_x = \mathbf{0}_{3 \times 3}$$

Therefore:

$$W_{Pos} = \frac{1}{n}W_y$$

Where:

$$W_y = \delta\bar{y}\delta\bar{y}^T = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{bmatrix} \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y & \sigma_x\sigma_z \\ \sigma_x\sigma_y & \sigma_y^2 & \sigma_y\sigma_z \\ \sigma_x\sigma_z & \sigma_y\sigma_z & \sigma_z^2 \end{bmatrix}$$

If isotropic error is assumed, then this matrix becomes:

$$W_y = \sigma_{Pos}^2 \mathbf{1}_{3 \times 3}$$

Where:  $\mathbf{1}_{3 \times 3} \triangleq \mathit{ones}(3,3)$

This is valid for the covariance of the frame whose origin is coincident with the C.G. of the measured points (i.e.  $\bar{p} = 0$ ). If this is not the case, the rotational uncertainty will also play a role in the uncertainty of the fitted frame position. For simplicity, the frames can be calculated as if they are aligned with principal axes; the uncertainty at any other arbitrary frame can be calculated the same way that target registration error (TRE) is calculated.

## Rotation Error Covariance of a Rigid Body Fit

Here the covariance associated with the best rigid rotation is calculated. This section assumes the same notation as the previous section on fitting bounds on this transformation.

From [Dorst, 2005], the covariance of this error is:

$$W_{d\bar{a}} = -RHR^T \left( \sum_i (\widehat{R\bar{x}_i}) W_Y (\widehat{R\bar{x}_i}) + \sum_i \hat{y}_i (RW_X R^T) \hat{y}_i \right) RHR^T$$

Since the set of points in the rigid body configuration,  $X$ , is known accurately, it can be assumed that the covariance of these points is negligible (i.e.  $W_x = \text{zeros}(3,3)$ ). This should be a good assumption because the machining accuracy is about two orders of magnitude better than the measurement accuracy. Therefore, we have:

$$W_{d\bar{a}} = -RHR^T \left( \sum_i (\widehat{R\bar{x}_i}) W_Y (\widehat{R\bar{x}_i}) \right) RHR^T$$

If we are trying to determine the error induced by perturbing points, but not causing a nominal rotation we set  $R$  to the identity:

$$R = I_3$$

Therefore, the covariance matrix simplifies to:

$$W_{d\bar{a}} = -H \left( \sum_i \hat{x}_i W_Y \hat{x}_i \right) H$$

Note that for  $i = 1 \dots n$ ,  $W_y$  represents uncertainty in the measurement's position error, and for  $i = n + 1 \dots 2n$ ,  $W_y$  represents uncertainty in measurement's orientation error.

This is saying that:

1. We want to minimize the product of  $\hat{x}_i W_Y \hat{x}_i$

Since we have no control over  $W_y$ , we can minimize the product by minimizing the distance from the origin of more uncertain sensors.

2. We want to minimize  $\|H\|$

$$\text{Where: } H = (\text{tr}(S)I - S)^{-1}$$

As before, this boils down to:

$$\text{minimize}(\|H\|) \rightarrow \text{maximize}(\|\text{tr}(S)I - S\|) \rightarrow \begin{array}{l} \text{maximize}(\|\text{tr}(S)\|) \\ \text{minimize}(\|S\|) \end{array}$$

Again, we want a large inertia, but evenly spread among the principal axes.

## Total Covariance for a rigid body fit

Combining the covariances determined above, we get the following complete covariance of a rigid body fit for both position and orientation.

$$W_{RB} = \begin{bmatrix} W_{ori} & 0 \\ 0 & W_{pos} \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

Where:

$$W_{ori} = \begin{bmatrix} -H \left( \sum_i \hat{x}_i (\delta \bar{y} \delta \bar{y}^T) \hat{x}_i \right) H \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

$$W_{pos} = \begin{bmatrix} \frac{1}{n} \delta \bar{y} \delta \bar{y}^T \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

Therefore, we end up with the following RB covariance matrix:

$$W_{RB} = \begin{bmatrix} -H \left( \sum_i \hat{x}_i (\delta \bar{y} \delta \bar{y}^T) \hat{x}_i \right) H & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{1}{n} \delta \bar{y} \delta \bar{y}^T \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

This is the covariance matrix presented earlier that is to be propagated through frame transformations using Jacobians in order to determine tool tip registration error. It will have to be calculated for both the reference frame and the tool rigid bodies.

## Experimental Validation

From before, the general equation for the rigid body covariance is:

$$W_{RB} = \begin{bmatrix} -H \left( \sum_i \hat{x}_i (\delta \bar{y} \delta \bar{y}^T) \hat{x}_i \right) H & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{1}{n} \delta \bar{y} \delta \bar{y}^T \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

Where:  $\delta \bar{y} = [\sigma_x \quad \sigma_y \quad \sigma_z]^T$  is the standard deviation of the position error

$$H = \left[ \sum_{i=1}^{2n} (\bar{y}_i^T \bar{y}_i I_3 - \bar{y}_i \bar{y}_i^T) \right]^{-1}$$

$\hat{x}_i = skew(\bar{x}_i)$  is the  $i^{\text{th}}$  sensor position with respect to the frame C.G.

$\bar{y}_i$  is the  $i^{\text{th}}$  measured position with respect to the measurement C.G.,

and is assumed to be the same as  $\bar{x}_i$  for these calculations

$n$  is the number of sensors in the tool (each sensor has two contributions)

We now calculate the theoretical values of the covariance matrix for an actual tool rigid body and compare them to simulated results. The tool used for this experiment is composed of two orthogonal sensors and is configured as shown in Table 4-1. This is the configuration for the plastic *Test Tool* shown earlier in Figure 3-26.

	<u>Position</u>	<u>Orientation</u>
Sensor 1	$\bar{x}_1 = \begin{bmatrix} 9.65 \\ -0.15 \\ 0 \end{bmatrix}$	$\bar{x}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$
Sensor 2	$\bar{x}_2 = \begin{bmatrix} -9.65 \\ 0.15 \\ 0 \end{bmatrix}$	$\bar{x}_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
Tool Configuration for <i>Test Tool</i>		

**Table 4-1**

Uncertainty is represented in the following form:

$$\bar{\sigma} = \begin{bmatrix} \sigma_{r_x} & \sigma_{r_y} & \sigma_{r_z} & \vdots & \sigma_{p_x} & \sigma_{p_y} & \sigma_{p_z} \\ w\sigma_{\theta_x} & w\sigma_{\theta_y} & w\sigma_{\theta_z} & \vdots & \sigma_x & \sigma_y & \sigma_z \end{bmatrix}^T$$

We choose the following to represent the uncertainty for the experiment:

$$\bar{\sigma} = \begin{bmatrix} w1.3\frac{\pi}{180} & w1.3\frac{\pi}{180} & w1.3\frac{\pi}{180} & \vdots & 2.5 & 2.5 & 2.5 \end{bmatrix}^T$$

Setting the relative orientation weight,  $w=100$ , we get:

$$\bar{\sigma} = [2.27 \quad 2.27 \quad 2.27 \quad \vdots \quad 2.5 \quad 2.5 \quad 2.5]^T$$

To simulate this scenario, we take the tool and arbitrarily choose a position and orientation. With the tool in that pose, we can back out where the corresponding sensors are based on the tool configuration; these are the  $\bar{x}_i$  values. We then distort the sensors measurements according to a given error distribution; this gives us the ‘measured’ sensor readings,  $\bar{y}_i$ . Using the same function as described earlier for aligning frames, we find the transformation that aligns the set of  $\bar{x}_i$  to the measurements  $\bar{y}_i$ . We then calculate the associated covariance of this after 10,000 simulated trials.

The results of this experiment are follows. Note the close agreement between the a priori calculated covariance and that of the simulation results.

W_calc =						
	3.1250	3.1250	3.1250	0	0	0
	3.1250	3.1250	3.1250	0	0	0
	3.1250	3.1250	3.1250	0	0	0
	0	0	0	0.0005	-0.0000	-0.0003
	0	0	0	-0.0000	0.0005	-0.0003
	0	0	0	-0.0003	-0.0003	0.0003
W_sim =						
	3.1906	3.1698	3.1459	0	0	0
	3.1698	3.1491	3.1254	0	0	0
	3.1459	3.1254	3.1019	0	0	0
	0	0	0	0.0005	0.0005	0.0003
	0	0	0	0.0005	0.0005	0.0003
	0	0	0	0.0003	0.0003	0.0002

These preliminary results show very good agreement between the experimental values and the predicted values for the covariance matrix. Clearly more analysis and verification are in order before this should be used as a predictor of tool accuracy, but this is a promising start. Unfortunately, in a real environment the error is not normally distributed. So, these design tools are better suited for determining how to minimize the covariance, rather than explicitly trying to calculate it.

## Overall Error Bounds from Localization Error

The key result of all this analysis is the determination of the accuracy of the tool tip with respect to the target in the frame of a CRF. Therefore, this section brings together the previous results.

### Orientation Error:

$$\left| \delta \theta_{CRF}^{Tool} \right| \leq \left| \delta \theta_{Base}^{CRF} \right| + \left| \delta \theta_{Base}^{Tool} \right|$$

Where:  $\left| \delta \theta_{Base}^{Tool} \right| \leq \|H_{Tool}\| \sum_i \|\bar{p}_{Tool,i}\| \|\delta \bar{p}_{Tool,i}\|$

$$\left| \delta \theta_{Base}^{CRF} \right| \leq \|H_{CRF}\| \sum_i \|\bar{p}_{CRF,i}\| \|\delta \bar{p}_{CRF,i}\|$$

Plugging in the above terms:

$$\left| \delta \theta_{CRF}^{Tool} \right| \leq \|H_{Tool}\| \sum_i \|\bar{p}_{Tool,i}\| \|\delta \bar{p}_{Tool,i}\| + \|H_{CRF}\| \sum_i \|\bar{p}_{CRF,i}\| \|\delta \bar{p}_{CRF,i}\|$$

As before, note that for  $i = 1 \dots n$ ,  $\bar{p}_i$  represents the sensor position in a RB with respect to its C.G. and  $\delta \bar{p}_i$  represents the translation error of this point. For  $i = n + 1 \dots 2n$ ,  $\bar{p}_i$  represents the weighted orientation vectors aligned with the  $i^{\text{th}}$  sensor z-axis,  $w\bar{n}_i$ , and  $\delta \bar{p}_i$  represents the error at the tip of these vectors induced by rotation error.

Splitting up the position and orientation terms:

$$\left| \delta \theta_{CRF}^{Tool} \right| \leq \|H_{Tool}\| \sum_{i=1}^n \|\bar{p}_{Tool,i}\| \|\delta \bar{p}_{Tool,i}\| + \|H_{CRF}\| \sum_{i=1}^n \|\bar{p}_{CRF,i}\| \|\delta \bar{p}_{CRF,i}\|$$

$$+ \|H_{Tool}\| \sum_{i=n+1}^{2n} \|\bar{p}_{Tool,i}\| \|\delta \bar{p}_{Tool,i}\| + \|H_{CRF}\| \sum_{i=n+1}^{2n} \|\bar{p}_{CRF,i}\| \|\delta \bar{p}_{CRF,i}\|$$

Putting into a more standard notation in terms of FLE components:

$$\left| \delta \theta_{CRF}^{Tool} \right| \leq \|H_{Tool}\| \sum_{i=1}^n \|\bar{p}_{Tool,i}\| |\delta \bar{p}_{EMTS}| + \|H_{CRF}\| \sum_{i=1}^n \|\bar{p}_{CRF,i}\| |\delta \bar{p}_{EMTS}|$$

$$+ \|H_{Tool}\| w^2 \sum_{i=1}^n |\delta \theta_{EMTS}| + \|H_{CRF}\| w^2 \sum_{i=1}^n |\delta \theta_{EMTS}|$$

Where:  $\bar{p}_{RB,i} \forall i \in [n + 1 \dots 2n] = \|w\bar{n}_{RB,i}\| = w$

$$\delta \bar{p}_{RB,i} \forall i \in [n + 1 \dots 2n] \approx |\delta \theta_{EMTS}| \|w\bar{n}_{RB,i}\| = w |\delta \theta_{EMTS}|$$

$|\delta \theta_{EMTS}|$  is the angular accuracy of the system (radians)

$|\delta \bar{p}_{EMTS}|$  is the position accuracy of the system (mm)

Making some simplifications, we end up with the bound on the orientation error:

$$\left| \delta \theta_{\substack{Tool \\ CRF}} \right| \leq n |\delta \bar{p}_{EMTS}| \left( \|H_{Tool}\| |\tilde{p}_{Tool}| + \|H_{CRF}\| |\tilde{p}_{CRF}| \right) + w^2 n |\delta \theta_{EMTS}| \left( \|H_{Tool}\| + \|H_{CRF}\| \right)$$

Where:  $|\tilde{p}_{RB}|$  is the average distance of the sensors in a RB from the C.G. for each rigid body (Tool and CRF)

This results looks counterintuitive at first because it seems as though there should be fewer sensors and they should be closer together to increase the accuracy. This is **not** true, as will be shown in the next step.

Plugging the following into  $H$ :

$$H = (tr(S)I_3 - S)^{-1}$$

$$\text{Where: } S = YY^T = \sum_{i=1}^n \bar{y}_i \bar{y}_i^T$$

$$tr(S) = \sum_{i=1}^n \bar{y}_i \cdot \bar{y}_i = \sum_{i=1}^n \|\bar{y}_i\|^2$$

We want to minimize  $\|H\|$ , and in order to do that:

$$\text{minimize}(\|H\|) \rightarrow \begin{array}{l} \text{maximize}(|tr(S)|) \\ \text{minimize}(\|S\|) \end{array}$$

Therefore, in order to minimize  $H$ , we want to maximize  $tr(S)$  while minimizing the magnitude of  $S$ . Since the terms here are squared, they override the effects of the linear relationship shown above. Therefore, in fact we want to have a tool that has many sensors, and those sensors are placed such that they maximize the effective inertia about their C.G. This is the same problem as we have seen twice before above.



**Position Error:**

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \left\| \delta \bar{p}_{Base}^{Tool} \right\| + \left\| \delta \bar{p}_{Base}^{CRF} \right\|$$

$$\text{Where: } \left\| \delta \bar{p}_{Base}^{Tool} \right\| \leq \max_i \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| + \left| \delta \theta_{Base}^{Tool} \right\| \left\| \bar{p}_{Base}^{Tool} \right\|$$

$$\left\| \delta \bar{p}_{Base}^{CRF} \right\| \leq \max_i \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| + \left| \delta \theta_{Base}^{CRF} \right\| \left\| \bar{p}_{Base}^{CRF} \right\|$$

Plugging in for the position error terms:

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \left( \max_i \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| + \left| \delta \theta_{Base}^{Tool} \right\| \left\| \bar{p}_{Base}^{Tool} \right\| \right) + \left( \max_i \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| + \left| \delta \theta_{Base}^{CRF} \right\| \left\| \bar{p}_{Base}^{CRF} \right\| \right)$$

Using the triangle inequality again to separate the components:

$$\left\| \delta \bar{p}_{CRF}^{Tool} \right\| \leq \max_i \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| + \max_i \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| + \left| \delta \theta_{Base}^{Tool} \right\| \left\| \bar{p}_{Base}^{Tool} \right\| + \left| \delta \theta_{Base}^{CRF} \right\| \left\| \bar{p}_{Base}^{CRF} \right\|$$

Plugging in the orientation error terms from the earlier calculations:

$$\begin{aligned} \left\| \delta \bar{p}_{CRF}^{Tool} \right\| &\leq \max_i \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| + \max_i \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| \\ &+ \left\| H_{Base}^{Tool} \right\| \sum_i \left( \left\| \bar{p}_{Base,i}^{Tool} \right\| \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| \right) \left\| \bar{p}_{Base}^{Tool} \right\| + \left\| H_{Base}^{CRF} \right\| \sum_i \left( \left\| \bar{p}_{Base,i}^{CRF} \right\| \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| \right) \left\| \bar{p}_{Base}^{CRF} \right\| \end{aligned}$$

As before, note that for  $i = 1 \dots n$ ,  $\bar{p}_i$  and  $\delta \bar{p}_i$  represent the positions and the position errors, and for  $i = n + 1 \dots 2n$ , these represent the orientations and the orientation errors.

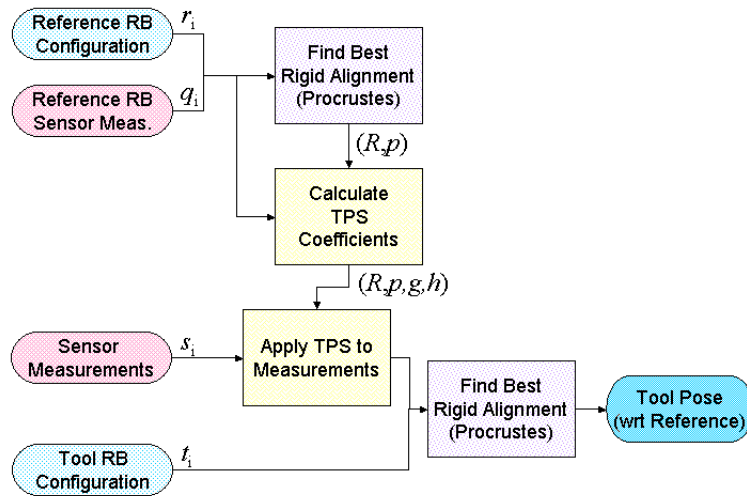
The above equation gives bounds on the tracking accuracy of a tool with respect to a reference rigid body (CRF). This bound is a relatively liberal bound on the error that shows the worst-case position error. By plugging in the alternate definition for the position error we get the following equation, which is more representative of the position error bounds, but is still a high upper limit.

$$\begin{aligned} \left\| \delta \bar{p}_{CRF}^{Tool} \right\| &\leq \frac{1}{n} \sum_{i=1}^n \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| + \frac{1}{n} \sum_{i=1}^n \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| \\ &+ \left\| H_{Base}^{Tool} \right\| \sum_i \left( \left\| \bar{p}_{Base,i}^{Tool} \right\| \left\| \delta \bar{p}_{Base,i}^{Tool} \right\| \right) \left\| \bar{p}_{Base}^{Tool} \right\| + \left\| H_{Base}^{CRF} \right\| \sum_i \left( \left\| \bar{p}_{Base,i}^{CRF} \right\| \left\| \delta \bar{p}_{Base,i}^{CRF} \right\| \right) \left\| \bar{p}_{Base}^{CRF} \right\| \end{aligned}$$

## Thin-Plate Spline Measurement Compensation

### Local Distortion Modeling

Ideally, one would have a characterized environment when performing a procedure where it is necessary to track a tool with respect to a reference frame. Unfortunately, this is often not a practical solution. So, in order to at least offer some improvement over naive frame transformation between the reference and the tool I propose a method, which takes into account a local distortion model. This model is represented using Thin-Plate Spline (TPS) interpolation techniques, which are used to interpolate a particular sensor's measurement error based upon the residual error of the sensors used to align the reference frame. Figure 4-12 shows a block diagram that represents the data flow in such an algorithm.



Thin-Plate Spline interpolation to improve tool tracking wrt. a reference  
**Figure 4-12**

### Algorithm

1. Read sensor measurements corresponding to reference rigid body,  
 $\bar{q}_i = (\bar{a}_i, \bar{m}_i) \in \mathbb{R}^6$ .
2. Find the best rigid transformation  $(R, \bar{p}) \in SE(3)$  that aligns  $\bar{q}_i$  to the rigid body configuration,  $\bar{r}_i = (\bar{b}_i, \bar{n}_i) \in \mathbb{R}^6$ .

3. Transform the sensor measurements into the reference RB's frame:

$$\bar{a}'_i = R\bar{a}_i + \bar{p} \in \mathbb{R}^3$$

$$\bar{m}'_i = R\bar{m}_i \in \mathbb{R}^3$$

These measurements will now be in the same frame as the RB's configuration.

4. Calculate the residual error between  $\bar{q}_i' = (\bar{a}_i', \bar{m}_i')$  and  $\bar{r}_i = (\bar{b}_i, \bar{n}_i)$ .

Position Error:  $\bar{\varepsilon} = \bar{b}_i - \bar{a}_i' \in \mathbb{R}^3$

Orientation Error:  $\bar{\delta} = Rod(\bar{m}_i', \bar{n}_i) \in \mathbb{R}^3$

Where:  $Rod(a,b)$  is the Rodriguez vector corresponding to the rotation that aligns the vectors  $a$  to the vector  $b$ .

5. Calculate the TPS coefficients that take sensor measurements in the reference RB frame,  $\bar{q}_i' = (\bar{a}_i', \bar{m}_i')$ , as the input and outputs the associated errors,  $(\bar{\varepsilon}, \bar{\delta})$ .
6. Compensate for the other sensor measurements using the TPS coefficients.

- a. Transform the sensor measurements into the reference RB's frame:

$$\bar{s}_i \xrightarrow{(R,p)} \bar{s}_i' \in \mathbb{R}^6$$

- b. Calculate the associated errors using TPS  
c. Use interpolated errors to compensate measurements appropriately.

7. Find the best rigid transformation  $(R_i, \bar{p}_i) \in SE(3)$  that aligns  $\bar{s}_i'$  to the rigid body configuration,  $\bar{t}_i$ .

This is the relative frame of the tool with respect to the reference, which is exactly what we need for the tool tracking scenario.

Following is a detailed look at TPS interpolation techniques and how it applies to this problem.

## Thin-Plate Spline Interpolation

A detailed method for applying thin-plate spline interpolation is presented in [Bookstein, 1991]. Here, the primary goal is interpolation of a warping of an image  $\in \mathbb{R}^2$ . We will follow this up with a proposed extension into 6 DoF interpolation.

The thin plate spline interpolation technique used looks like this:

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^K w_i U(|Z_i - (x, y)|)$$

Where:  $f$  is the measure being interpolated

$x$  and  $y$  are the measurement coordinates

$a_i$  represent the Affine coefficients

$w_i$  represent the TPS coefficients

$U(d)$  represents a bending energy function associated with the TPS

$$U(d) = d^2 \log d^2$$

$d_i$  is the distance of the measurement from the  $i^{\text{th}}$  node/knot point,  $Z_i$

$$d_i = Z_i - (x, y)$$

The proposed framework for solving this problem takes the following form.

$$P = \begin{bmatrix} 0 & U(d_{1,2}) & \cdots & U(d_{1,K}) \\ U(d_{2,1}) & 0 & & \vdots \\ \vdots & & 0 & U(d_{K-1,K}) \\ U(d_{K,1}) & \cdots & U(d_{K,K-1}) & 0 \end{bmatrix} \in \mathbb{R}^{K \times K}$$

$$Q = \begin{bmatrix} 1 & x_1 & \cdots & z_1 \\ 1 & x_2 & \cdots & z_2 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_K & \cdots & z_K \end{bmatrix} \in \mathbb{R}^{K \times 3}$$

$$L = \begin{bmatrix} P & Q \\ Q^T & 0 \end{bmatrix} \in \mathbb{R}^{(K+3) \times (K+3)}$$

Where:  $K$  is the number of input locations

We want to solve the following equation in the least-squares sense:

$$LX \approx Y$$

$$\text{Where: } X = \begin{bmatrix} \bar{w} \\ \vdots \\ \bar{a} \end{bmatrix} \in \mathbb{R}^{K+3} \quad \text{represents the coefficients}$$

$$Y = \begin{bmatrix} \bar{v} \\ \vdots \\ \bar{0} \end{bmatrix} \in \mathbb{R}^{K+3} \quad \text{represents the values being interpolated}$$

Solving for the coefficients, we get:

$$L^{-1}Y \approx \begin{bmatrix} w_1, \dots, w_k \mid a_1, a_x, a_y \end{bmatrix} Y$$

These values can then be plugged directly into the interpolation function  $f(x,y)$ . If interpolating values in higher dimensions, there would effectively be a separate spline for each dimension (e.g.  $f(x,y)$ ,  $g(x,y)$ , etc.).

In order for this technique to be useful for our purposes, we must modify it such that we can interpolate 6 DoF errors from a 6 DoF measurement. The first step is to decide what the reference frame will be for the measurements, because it must obviously be the same for the initial measurements and the later ones being interpolated. It was decided that the best solution to this is to bas all measurement off of the frame of the reference body. This way, when done with the calculation, we have the measurement of the tool's sensors directly in the frame we want already.

The next critical step is to determine what will be interpolated. One option would be to have the TPS directly predict the desired pose from the original measurement, but this proved difficult when dealing with the orientation. The decision was to model the error using the TPS; for position this is simply the Euclidean distance, for orientation this is the Rodriguez vector of the rotation that corrects the measurements. Therefore, we have six inputs (position and axis of sensor) and six outputs (position error and rotation error) to the TPS compensation.

Probably the most important step in determining the best way to extend the algorithm from 2D into 6D is to decide on what to do about the distance function. For 2D, or even 3D, points, this is clearly just the Euclidean distance between the sensor origins. However, the Euclidean norm of a vector on  $\mathbb{R}^6$  that contains both position and orientation information has no real physical meaning. A possible solution appears to be a weighted combination of the position error, as a Euclidean distance, and the magnitude of the rotation error. The key to making this algorithm work is determining the appropriate weight of the translational distance and the orientation 'distance.' This weight is related to how much we feel that the error is contributed to by a change in orientation relative to a change in position of the sensor. Since from experience, the error seems to be very orientation dependent, it would be expected that this weight should be quite high.

Since we already have a best rigid alignment of the points, we decided to drop the Affine term from the earlier equation, and are only determining the thin-plate term.

For 4 sensors being used to generate the TPS:

$$L = P = \begin{bmatrix} 0 & U(d_{1,2}) & U(d_{1,3}) & U(d_{1,4}) \\ U(d_{2,1}) & 0 & U(d_{2,3}) & U(d_{2,4}) \\ U(d_{3,1}) & U(d_{3,2}) & 0 & U(d_{3,4}) \\ U(d_{4,1}) & U(d_{4,2}) & U(d_{4,3}) & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$

Where:

$d_{i,j} = \|a'_i - a'_j\| + w \left| \cos^{-1}(m'_i \bullet m'_j) \right|$  is the proposed distance function

$\bar{q}'_i = (\bar{a}'_i, \bar{m}'_i)$  is the  $i^{\text{th}}$  sensor measurement in the reference RB's frame

$$U(d) = d^2 \log d$$

Again, we want to solve:

$$LX=Y$$

$$\text{Where: } X = \begin{bmatrix} w_{1_x} & w_{2_x} & w_{3_x} & w_{4_x} \\ w_{1_y} & w_{2_y} & w_{3_y} & w_{4_y} \\ w_{1_z} & w_{2_z} & w_{3_z} & w_{4_z} \\ w_{1_{Rx}} & w_{2_{Rx}} & w_{3_{Rx}} & w_{4_{Rx}} \\ w_{1_{Ry}} & w_{2_{Ry}} & w_{3_{Ry}} & w_{4_{Ry}} \\ w_{1_{Rz}} & w_{2_{Rz}} & w_{3_{Rz}} & w_{4_{Rz}} \end{bmatrix}^T \in \mathbb{R}^{4 \times 6} \text{ are the measurements at the nodes}$$

$$Y = \begin{bmatrix} v_{1_x} & v_{2_x} & v_{3_x} & v_{4_x} \\ v_{1_y} & v_{2_y} & v_{3_y} & v_{4_y} \\ v_{1_z} & v_{2_z} & v_{3_z} & v_{4_z} \\ v_{1_{Rx}} & v_{2_{Rx}} & v_{3_{Rx}} & v_{4_{Rx}} \\ v_{1_{Ry}} & v_{2_{Ry}} & v_{3_{Ry}} & v_{4_{Ry}} \\ v_{1_{Rz}} & v_{2_{Rz}} & v_{3_{Rz}} & v_{4_{Rz}} \end{bmatrix}^T \in \mathbb{R}^{4 \times 7} \text{ are the associated errors at the nodes}$$

Solving for the coefficients in  $X$ , we can then generate the TPS interpolation equation for a tool sensor measurement in the reference RB frame,  $s$ , as follows:

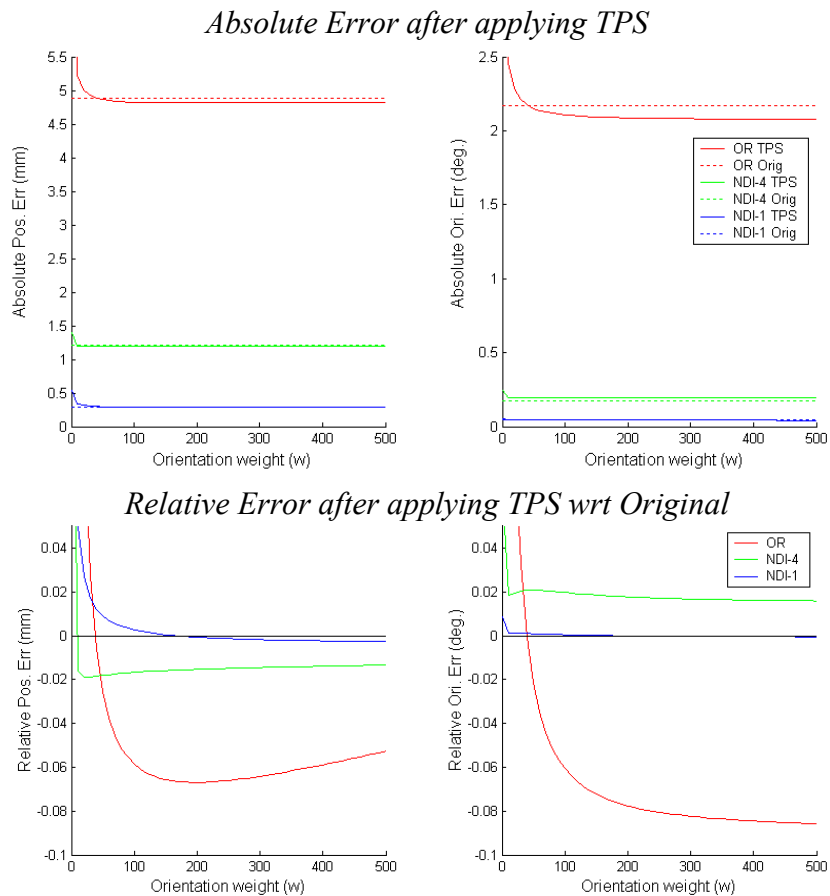
$$\begin{bmatrix} \mathcal{E}_x \\ \mathcal{E}_y \\ \mathcal{E}_z \\ \delta_{Rx} \\ \delta_{Ry} \\ \delta_{Rz} \end{bmatrix} (s) = \sum_{i=1}^K \begin{bmatrix} w_{i_x} \\ w_{i_y} \\ w_{i_z} \\ w_{i_{Rx}} \\ w_{i_{Ry}} \\ w_{i_{Rz}} \end{bmatrix} U(d(Z_i, s))$$

Then, the errors can be applied to the measurement, such that  $s$  can be compensated based on the local distortion field prior to fitting a tool RB frame to the measurements.

## TPS Compensation Results

The first step to make the TPS algorithm useful is to determine the appropriate weighting of the translation and the orientation ‘distances.’ For this experiment, a reference rigid body was generated with four sensors in a realistic configuration for a cranial reference. Then, 1000 random sensor measurements were made around the reference. The measurements were made in a realistic pattern for ENT procedures and were distorted in the simulator using several 3<sup>rd</sup> order polynomial models of the field distortion. The magnitude of the tracking error after applying TPS compensation was calculated for each sensor measurement based on the known original measurement that was input into the simulator; this was compared to the error with no compensation.

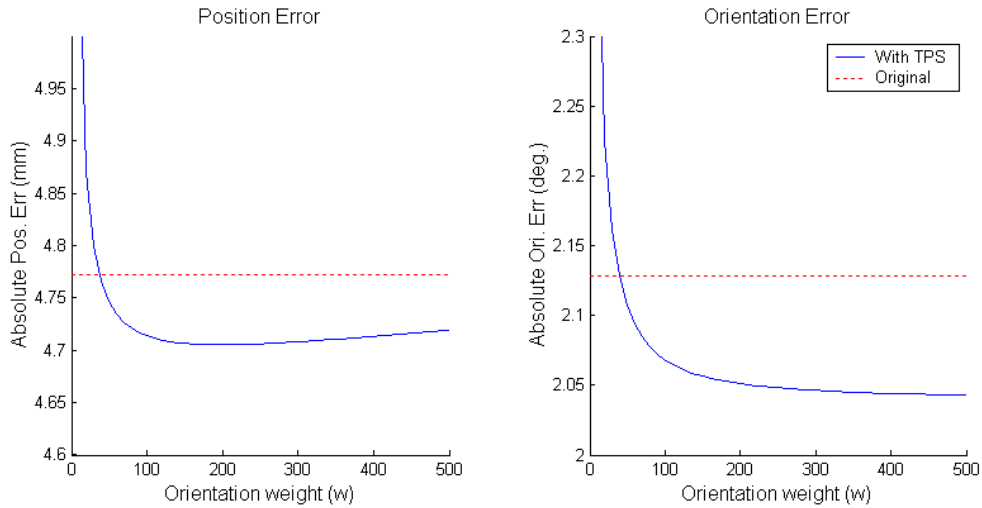
The plot on the top of Figure 4-13 shows the RMS error values, and the bottom shows the difference between compensated and uncompensated errors as a function of the position vs. orientation weighting factor used in the distance function. The weighting factor was incremented from zero to 500 in increments of ten. As was expected, the greater the weight, the lower the error in general; this indicates a high dependence of both position and orientation error on the orientation of the sensor.



Sensor tracking error using TPS compensation for three environments  
**Figure 4-13**

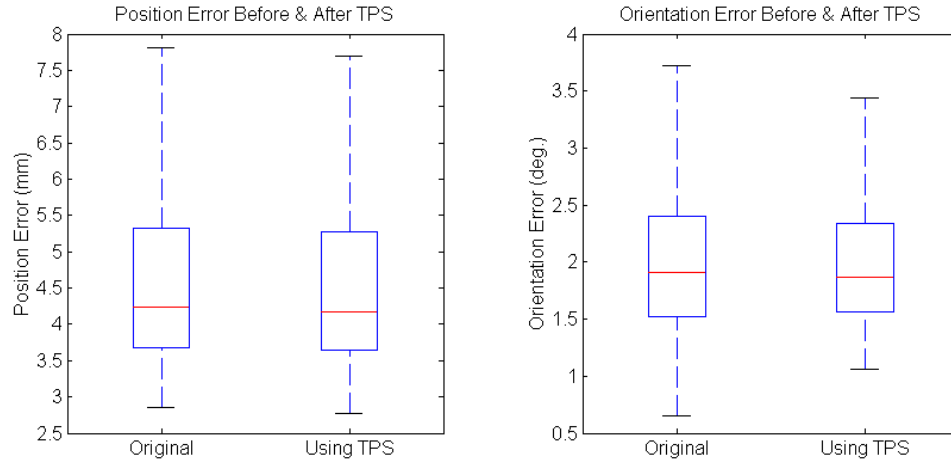


To get a better appreciation for the affects, we look in more detail and the results from applying this technique in the simulate operating room since this is the target environment. In Figure 4-13 we show the tracking error of a single sensor measurement from a collection of 1000 simulated measurement in the same experiment as above. However, here the results shown the absolute tracking error so that it is clearer what the magnitude of the improvement is with respect to the baseline accuracy.



Absolute sensor tracking error using TPS compensation in simulated OR environment  
**Figure 4-14**

From these plot in Figure 4-14, it is clear that there is an optimal weighting factor that we can use. It appears that an orientation weight of  $w=250$  falls in the minimum of the position error and is close to where the orientation error approaches an asymptote. Now, applying thin-plate spline interpolation to the random sensor measurements in the simulated OR using this weight, we get the following results in Figure 4-15. These results show the distribution of tracking error of the sensors both before and after applying TPS interpolation to measurements distorted suing the same 3<sup>rd</sup> order Bernstein polynomial model of the OR data set as before. Tracking error was determined by comparing the measurements to the original sensor pose that was input to the simulator.



Tracking error distribution for simulated 5 DoF measurements in the OR with and without the assistance TPS interpolation

(Red line at median, box extents at 1<sup>st</sup> & 3<sup>rd</sup> quartiles, whiskers extend to 1.5\*IQR)

**Figure 4-15**

It is clear that there is in fact an improvement in both the mean error and also the variance within the measurements. Although this is a relatively small improvement, it serves as proof that this technique does have some validity. This is just a representative result, but the results are very similar with other distortion models and for other distributions of points. Clearly, the effectiveness of this technique depends on the number, configuration, and alignment of the sensors in the reference frame, and also the nominal poses of the sensors in the tool with respect to the reference frame.

## ***Contributions***

This chapter discussed the analytical design of tracked tools for image guided surgery, and in particular electromagnetically tracked tools. This work was originally intended to following the footsteps of optically tracked tool design papers such as [West, 2004] and [Fitzpatrick, 2001]. However, the goal was to extend these techniques to be more adequate for the EM tracking scenario. This involved inclusion of techniques for accounting for tool orientation and incorporation of field distortion information. Further, this chapter detailed covariance propagation and estimation techniques for tracked tools. This work is primarily focused on the extension and adaptation of the uncertainty propagation work presented in [Dorst, 2005], [Pennec, 1995], and [Smith, 1990]. The end result was a detailed analysis of the bounds and covariance of frame fitting and propagation through frame transformations to allow for prediction of bounds on the error and uncertainty of a tracked EM tool. An important note is that the primary goal was not so much for actual calculation of these uncertainties, but as a method for determining how to minimize these bounds and uncertainties. The design tools presented can be used alone, or more practically they can be used to help design an initial set of tools to be tested in the simulator as described in Chapters 5 and 6.

## Chapter 5: Measurement and Simulation Software

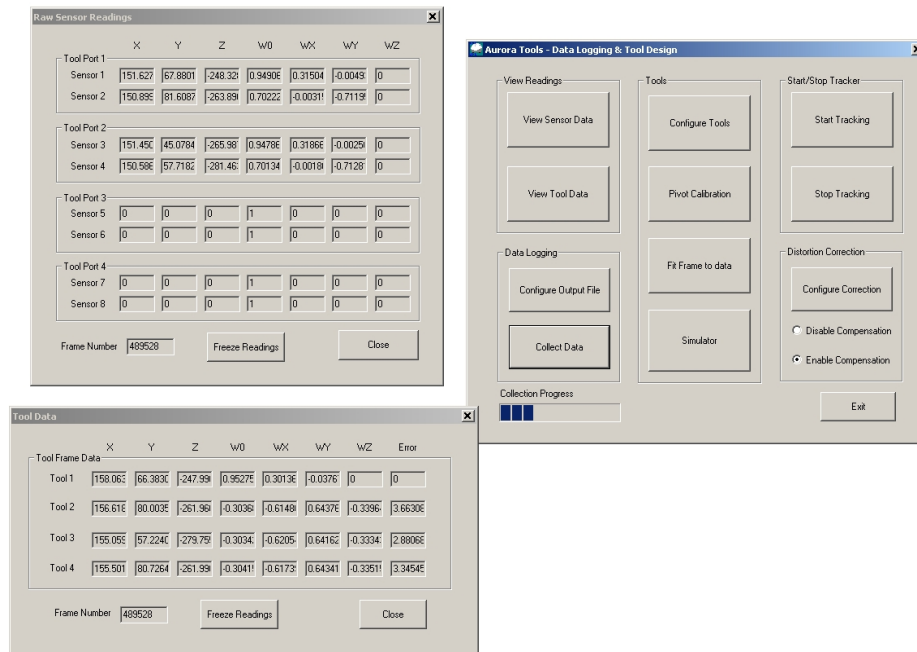
### **Overview**

Developing an EM tracked tool can be a very time consuming process. In order to optimize the tool it would most likely take quite a few design iterations. Traditionally for each of these iterations it would be necessary to build the tool, collect sample data, and perform an error analysis. Therefore, a simulator that would allow the user to virtually design and test a tool would be an invaluable asset, and that is exactly what is introduced in the next section. In the system, the user is able to generate an arbitrary Tool RB configuration and arbitrary Reference RB configuration of any number of sensors. The reference frame is fixed in an arbitrary, user specified pose in space, and the tool is virtually moved around the reference in a given pattern, while the sensor measurements of both are artificially distorted according to a particular error field mapping. By doing this, it is possible to predict the tool tip registration error for a particular tool in a particular scenario before even building the tool. This section details the functionality and use of the simulator program and details the methodology behind the software.

# Aurora Tools Software

## Summary

The *Aurora Tools* software is a compilation of many features that are useful for designing and analyzing EM tracked tools. It allows for data collection, frame fitting, logging, distortion compensation, and sensor/tool simulation. This section details the functionality of the software, followed by a detailed look at the methods behind that functionality. A screen shot of the software in action while performing data collection and real-time measurement and tool frame compensation is displayed in Figure 5-1.



Aurora Tools software during data collection

Figure 5-1

## User Guide

The user guide gives an overview of the main functionality of the software. It shows a very straightforward, step-by-step approach to the use of the program's main features. Note that there is further functionality available by directly customizing the underlying code, but the main useful features are all accessible through the GUI. Following is a much more detailed description of what is included in the software, and how these elements work and interact.

**USER GUIDE IS INCLUDED IN APPENDIX A**

## Function Descriptions

- `int CAurora_ToolsDlg::ConfigureDistCorrection()`  
Choose *.tre* file that contains the description of the distortion model, including the polynomial coefficients representing a distortion mapping for a given environment.
- `void CAurora_ToolsDlg::OnCollectData()`  
Opens the data files specified for logging and sets the flag (*CollectData*) for enabling data logging.
- `void CAurora_ToolsDlg::OnButtonSensorData()`  
Opens the window to display the sensor positions and orientation in real time for the user.
- `void CAurora_ToolsDlg::OnButtonToolData()`  
Opens the window to display the tool positions, orientations, and 5d-6D fitting error in real time for the user.
- `void CAurora_ToolsDlg::OnButtonStartTracking()`  
Enables the Aurora to start tracking and sets a flag (*Aur\_tracking*). The first time this is clicked the Aurora is initialized. After that, the unit only needs to be given a command to start measuring if it is stopped.
- `void CAurora_ToolsDlg::OnConfigOutput()`  
Opens the window for configuring data collection parameters and also for specifying what data should be logged to file and the file names to use for each.
- `void CAurora_ToolsDlg::OnBUTTONSimulator()`  
Opens the Simulator window.
- `void CAurora_ToolsDlg::OnBUTTONPivotCal()`  
Opens window for performing a pivot calibration.
- `void CAurora_ToolsDlg::OnButtonConfigureTool()`  
Use to configure which tool frames are to be calculated, displayed, and logged.
- `void CAurora_ToolsDlg::OnCONFIGDistCorrection()`  
Opens window for specifying which polynomial file and configuration to use for the distortion mapping.
- `int CAurora_ToolsDlg::ConfigureDistCorrection()`  
Loads and initializes the polynomial for distortion compensation. A default polynomial is loaded if the user does not input a new one. This is automatically run if the input filename is changed by the user.

```
int CAurora_ToolsDlg::AuroraConfig()
    Initializes and configures the Aurora and begins tracking.

void CAurora_ToolsDlg::OnTimer(UINT nIDEvent)
    Where the real-time tracking, compensation, tool frame fitting, display, and
    logging is called on at a specified timer interval.

void CAurora_ToolsDlg::UpdateDisplay()
    Updates the windows showing the sensor measurements and the tool
    measurements.

void CAurora_ToolsDlg::ReadAurRomCfg(char* save_filename)
    Reads in the tool configuration from file, this is used for tool frame fitting of real-
    time data as well as for the simulator.
```

**OR**

```
void CAurora_ToolsDlg::SetAurRom()
    Uses the hard-coded tool configuration, this is used for tool frame fitting of real-
    time data as well as for the simulator.
```

```
cisTrackerTool*
CAurora_ToolsDlg::findframe( CArray<cisTrackerTool*,
cisTrackerTool*> &Meas_Sens, CArray<cisTrackerTool*,
cisTrackerTool*> &RB_Sens, CArray<int,int> &RB_Index,
cisDouble *frame_qual)
    Determines the best fit frame that transforms the sensor configuration (RB_Sens)
    to the measured sensors (Meas_Sens). The input data is two arrays of Tracker
    Tools with an index that determines which sensors in the configuration are used.
    The output is a single Tracker Tool representing the best fit transformation and a
    number representing how good of a fit was produced (frame_qual).
```

```
cisVec3 CAurora_ToolsDlg::Pivot(
CArray<cisFrame, cisFrame> &F_tool, cisDouble *RMS_Error)
    Performs pivot calibration: determines tip location of a tool with respect to that
    tools frame from a set of frames representing that tool (F_tool). Also returns the
    quality of the fit (RMS_Error).
```

```
void CAurora_ToolsDlg::FindToolError(
cisVec3 toolPos, cisUnitQuat toolOri,
cisVec3 *CorrectedToolPostion,
cisUnitQuat *CorrectedToolOrientation)
    Corrects the position and orientation of a sensor using the specified polynomial fit.
toolPos and toolOri are the original Position and Orientation respectively. The
    outputs are CorrectedToolPostion and CorrectedToolOrientation.
```

```
void Simulator::OnRunSimulator()  
    Runs the simulator for a single trial of the tool and reference frames as input into  
    the GUI.  
  
void Simulator::OnSimulateRangeOfMotion()  
    Runs the simulator for a tool moved throughout the given range of motion about  
    the user inputted tool frame with respect to fixed reference frame as input by the  
    user through GUI.  
  
void Simulator::OnSimulateInputData()  
    Runs the simulator for the tool and reference frame pairs that are listed in the  
    given text file.  
  
void Simulator::RunSimulation(bool write_data)  
    The actual simulation where a tool frame and reference frame are distorted and  
    measured with respect to eachother. The variable write_data indicates whether or  
    not to output the results to the given text file.
```



## 5D to 6D Frame Generation

### Algorithm

```
cisTrackerTool* CAurora_ToolsDlg::findframe( CArray<cisTrackerTool*,
cisTrackerTool*> &Meas_Sens, CArray<cisTrackerTool*, cisTrackerTool*>
&RB_Sens, CArray<int,int> &RB_Index, cisDouble *frame_qual)
```

*RB\_Sens* is an array of type *cisTrackerTool* with a length corresponding to the number of sensors in that tool. This data type contains 6 DOF position and orientation information plus other data such as the time stamp and frame number. This variable contains the ideal frame of each sensor in the tool with respect to the tool's frame.

*Meas\_Sens* is an array of type *cisTrackerTool* with a length corresponding to the number of sensors being measured. This variable contains the frame of each measured sensor with respect to the Aurora Field Generator frame.

*RB\_Index* is an array of integers with the same length as *Meas\_Sens*. This variable contains the indices of the frames in *Meas\_Sens* that correspond to the entries in *RB\_Sens*.

*F\_Sens* and *F\_Meas* are of type *cisFrame* and both have a length of the number of sensors in the tool. They contain the frames pulled out of the corresponding entries of *RB\_Sens* and *Meas\_Sens* respectively.

*tool\_marker* and *tool\_marker\_meas* are the positions pulled out of the frames *F\_Sens* and *F\_Meas*. The means of these positions (*Amean* & *Bmean*) are calculated for each and subtracted out.

The matrices *Adiff* and *Bdiff* are generated using the demeaned positions and the z vectors of the rotations in *F\_Sens* and *F\_Meas*. The orientations are weighted using a weight *w* to equate the magnitudes to that of the demeaned positions.

The contents of these variables look something like this:

$$A_{diff} = \left[ \begin{array}{c} F_{Sens}[0].P-A_{mean} \\ \vdots \\ F_{Sens}[num\_sens-1].P-A_{mean} \\ \hline w \left( \begin{array}{c} F_{Sens}[0].Rz \\ \vdots \\ F_{Sens}[num\_sens-1].Rz \end{array} \right) \end{array} \right] \quad B_{diff} = \left[ \begin{array}{c} F_{Meas}[0].P-A_{mean} \\ \vdots \\ F_{Meas}[num\_sens-1].P-A_{mean} \\ \hline w \left( \begin{array}{c} F_{Meas}[0].Rz \\ \vdots \\ F_{Meas}[num\_sens-1].Rz \end{array} \right) \end{array} \right]$$

The CIS function `cisComputeBestRigidRotation(Adiff, Bdiff, R)` is run on these variables to generate the appropriate rotation to align the frame of the specified tool to that of the measured sensors. This is equivalent to a point cloud to point cloud registration.

## Least Squares Solution

The method used here is a modified version of the Arun method [Arun, 1987] for solving for the best rigid point cloud to point cloud rotation problem in the LS sense as discussed earlier in the Frame Fitting section of Chapter 4. The weighted orientations are treated the same way as the demeaned points; weights keep the position and orientation contributions balanced.

From before, write with more standard notation:

$$A = \begin{bmatrix} (\bar{p}_{\text{sens}_1} - \bar{p}_{\text{sens}})^T \\ \vdots \\ (\bar{p}_{\text{sens}_n} - \bar{p}_{\text{sens}})^T \\ \hline w \begin{pmatrix} \bar{n}_{\text{sens}_1}^T \\ \vdots \\ \bar{n}_{\text{sens}_n}^T \end{pmatrix} \end{bmatrix} \quad B = \begin{bmatrix} (\bar{p}_{\text{meas}_1} - \bar{p}_{\text{meas}})^T \\ \vdots \\ (\bar{p}_{\text{meas}_n} - \bar{p}_{\text{meas}})^T \\ \hline w \begin{pmatrix} \bar{n}_{\text{meas}_1}^T \\ \vdots \\ \bar{n}_{\text{meas}_n}^T \end{pmatrix} \end{bmatrix}$$

Where:  $\bar{p}$  represents the mean position of the given measurement set

$\bar{n}$  represents the unit vector pointing along the  $z$  axis of the given sensor

Define:

$$\begin{aligned} q_i &= A(i, 1:3)^T \\ q'_i &= B(i, 1:3)^T \end{aligned} \quad i = \{1, \dots, n\}$$

$$H = \sum_{i=1}^n q_i q_i'^T$$

Take the Singular Value Decomposition (SVD) of  $H$ :

$$H = U \Lambda V^T$$

Calculate the optimal rotation in the Least Squares (LS) sense:

$$R = X = V U^T$$

Verify the solution:

$$\det(X) = \begin{cases} +1 & \text{Set: } R = X \\ -1 & \text{Algorithm fails} \end{cases}$$

Using the LS SVD solution for the optimal rotation  $R$ , the position vector  $\bar{v}$  is calculated by rotating the mean position of the RB points with the given rotation  $R$  and subtracting that from the mean position of the sensor measurements.

The frame representing the location of the tool in Aurora coordinates is given by:

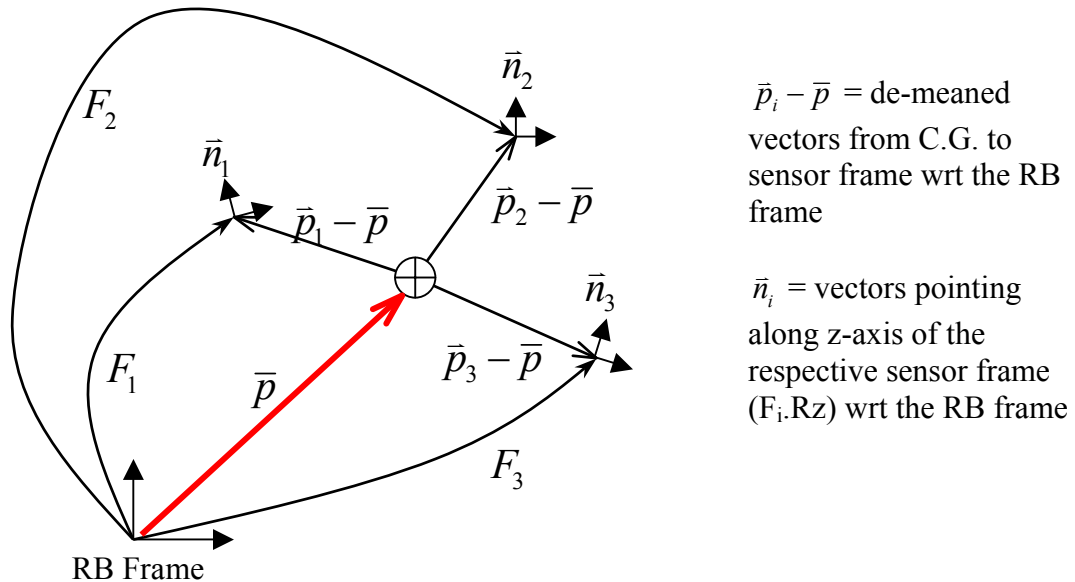
$$F = \text{cisFrame}(R, B_{\text{mean}} - R * A_{\text{mean}});$$

$$F_{\text{sens} \rightarrow \text{meas}} = \begin{bmatrix} R & \bar{v} \\ 0 & 1 \end{bmatrix} \in SE(3)$$

$$\text{Where: } \bar{v} = \bar{p}_{\text{meas}} - R * \bar{p}_{\text{sens}}$$

The variable *frame\_qual* is a quality measure of the fit performed as above. It can be calculated in one of two ways: either based solely on position, or also including orientation. By basing it only on position, this variable contains the RMS residual error between the measured points and those in the fit tool frame; this is often termed the fiducial registration error (FRE). To include orientation, a point can be placed along the z-axis of each sensor at a given distance, and these points can be compared to ones placed similarly on the ideal sensor location in the frame. By adjusting the distance, the weighting between position and orientation in the quality measure can be adjusted.

Figure 5-2 is a visual representation of three 5 DoF sensors in known positions and orientations with respect to the rigid body frame. This diagram is very similar for the three 5 DoF sensor measurements in the Aurora frame of reference.

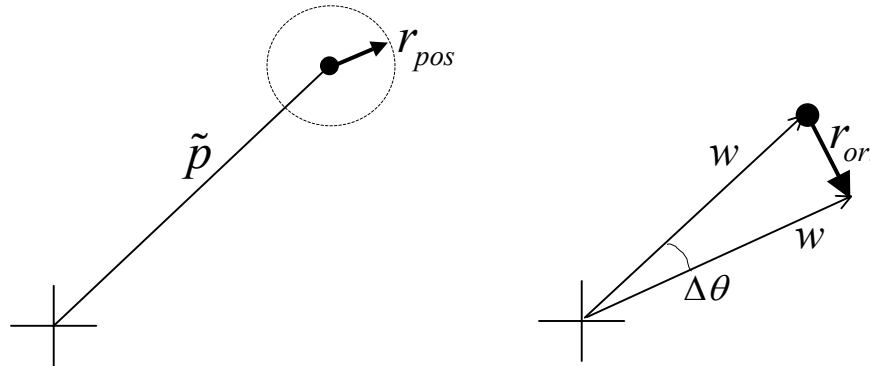


Sensor positions and orientations in a rigid body frame,  
and their location with respect to the C.G. of the positions

**Figure 5-2**

## Error Weight Calculation

In order to have the ‘optimal’ solution for the previously mention problem, it is necessary to determine the appropriate weighting factor for the angular component vs. the translational component of the registration least squares problem. First the magnitudes of the vectors for the two components are made to have the same magnitude by scaling the orientation vectors by the magnitude of the average distance of the sensors from the centroid of the rigid body. Then, these values are scaled by the relative radii of their respective “error ball,” representing a sphere in which we can assume the readings will actually fall in. This is based upon the accuracy of the system and the distribution of the sensors. After working though the problem, it is determined that the scaling by the average distance of the sensors from the mean cancels out with the mean distance in the error calculation due to orientation. Therefore, **the final estimate of the weight contains only the system’s positional accuracy and the orientation accuracy.**



**Figure 5-3**

Notation:  $\tilde{p} = \frac{1}{n} \sum_{i=1}^n \|\bar{p}_i - \bar{p}\|$  is the average distance of the sensors from their C.G.  
 $r_{pos}$  and  $r_{ori}$  are the radii of their respective error spheres (mm)  
 $w$  is the orientation weight we are trying to determine

The radius of the sphere enclosing the rotation-induced error is described as follows:

$$\frac{1}{2} r_{ori} = w \sin\left(\frac{\Delta\theta}{2}\right)$$

Where:  $\Delta\theta = \delta_{ori}$  is the angular accuracy of the system

Due to small values for  $\Delta\theta$ , we can make the following simplifying assumption:

$$r_{ori} \approx w \Delta\theta$$

Note that  $r_{ori}$  is a function of  $w$

The radius of the sphere enclosing the position-induced error is very straightforward:

$$r_{pos} = \delta_{pos}$$

Where:  $\delta_{pos}$  is the positional accuracy of the system

In order to determine the appropriate weighting factor, we want to set up the problem such that the relative contributions of the error to the inputs of the least squares problem are equal. The least squares problem aligns the clouds of points such that it minimizes the FRE; in this case, this cloud of points includes the actual sensor locations with respect to their C.G and the points lying at the tip of the scaled orientation vectors. Therefore, the weighting factor should be such that it equates the radii of the error spheres including the position error and the tip of the perturbed scaled orientation vectors.

Setting up the following equality allows us to solve for the weighting factor:

$$r_{pos} \triangleq r_{ori}(w)$$

Plugging in the above results for the radii of the error spheres:

$$\delta_{pos} = w\delta_{ori}$$

Solving for the error weight,  $w$ :

$$w = \frac{\delta_{pos}}{\delta_{ori}}$$

Where:  $\delta_{pos}$  is the positional accuracy of the system

$\delta_{ori}$  is the rotational accuracy of the system

### Verification of Analysis:

Here we try to confirm the above analytical solution. First the theoretical weights are determined. Then, a pivot calibration where the tool frames are fit to the sensor data is performed for various weighting factors. The residual tip errors are plotted and used as a gauge of the quality of the frame's fit. For this data, the sensors are centered on a 50mm diameter circle, so  $\tilde{p} = 25$  mm.

Using the Aurora's stated accuracy:

Position      0.7 mm RMS

Orientation    0.3° RMS

$$w_{ori} = \frac{0.7}{0.3 \frac{\pi}{180}} = 134$$

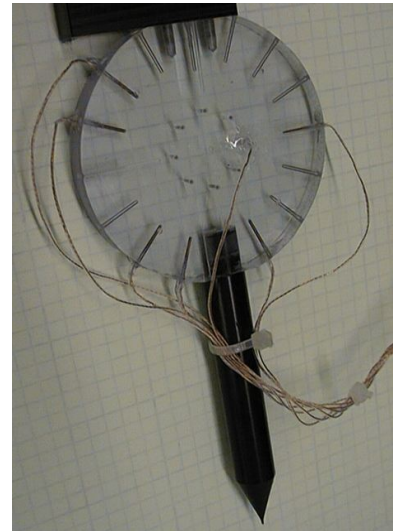
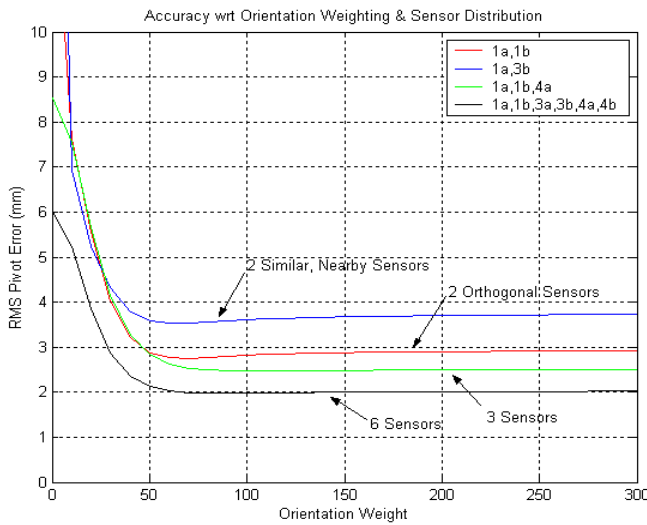
Using the accuracy determined in the experimental environment (Using Robot Calibration Results):

Position      2.23 mm RMS

Orientation    1.78° RMS

$$w_{ori} = \frac{1.2}{0.6 \frac{\pi}{180}} = 72$$

The algorithm for frame fitting is tested by incrementing the error weight from 0 to 300 by increments of 10 and fitting the measured sensors to a known (from manufacturing) frame. A pivot calibration was performed for each set of sensors, and the plot below shows the residual error at the tip after the pivot calibration was performed and the transformation applied to each measured tool frame. This experiment was repeated for several sets of sensors, which are labeled on the diagram. It is clear from Figure 5-4 that there is a distinct increase in accuracy (drop in RMS error) as the orientation weight is applied, a weight of zero corresponds to a point-to-point calibration only, with no influence of the sensor orientations. There is a consistent minimum in the trials between 50 and 125; the empirical results correlate well with the analytical results. Typically, when running the 5D to 6D algorithm, a weight of 100 is used; this seems to work well universally.



Accuracy as a function of orientation weighting  
**Figure 5-4**

Tool used for pivot data collection

## Real-time Measurement Compensation

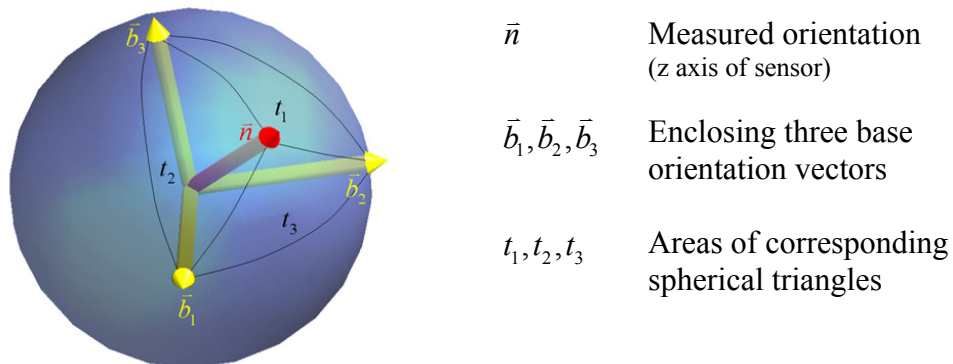
```
void CAurora_ToolsDlg::FindToolError(cisVec3 toolPos,
cisUnitQuat toolOri, cisVec3 *CorrectedToolPostion,
cisUnitQuat *CorrectedToolOrientation)
```

This function is used to determine the estimate of the actual position and orientation of a sensor for a given position and orientation measurement. It is based upon the polynomial approximation of the field distortion. The key to this function is the `GetError()` function which is part of the *AuroraCalibration* Software, which is accessible as a library:

```
PolyTree->GetError(toolPos, toolOri.Rotation().Rz, PosErr, OriErr);
```

This function is used to determine the estimate of the position and orientation error that the polynomial approximation predicts for a given position and orientation measurement. It does so by using the pre-calculated polynomial coefficients  $c_{i,j,k}^b$  that are stored in the specified *.tre* file. There is a set of coefficients for each dimension being interpolated along ( $X, Y, Z, Rx, Ry, Rz$ ). Inside each of those is a separate set of coefficient corresponding to each of the base orientation vectors.

To determine the estimate of position and orientation error, the space is divided into base orientations / basis vectors. For each base orientation, a polynomial is generated that describes the error as a function of ( $x, y, z$ ) position only. To solve for an arbitrary orientation, the three closest surrounding basis vectors are found as shown in Figure 5-5, and the relative contribution of each is calculated based on the size of the corresponding spherical triangle.



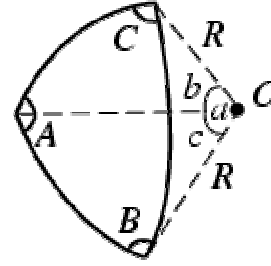
Definition of the spherical triangles used for spherical interpolation  
**Figure 5-5**

The area of a spherical triangle is defined in Figure 5-6 as:

$$\Delta = R^2 ((A + B + C) - \pi)$$

For this case:

$$t_b = (A + B + C) - \pi$$



**Figure 5-6**

Where:  $A, B, C$  are the interior angles of the opposing spherical triangle

The error weights  $w_b$ , for each of the three closest basis vectors ( $\bar{b}_1, \bar{b}_2, \bar{b}_3$ ) are calculated as follows:

$$w_b = \frac{t_b}{t_1 + t_2 + t_3} \quad \text{Such that: } \sum_b w_b = 1$$

These weights are plugged into the following equation:

$$\begin{bmatrix} \bar{e}_{pos} \\ \dots \\ \bar{e}_{ori} \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_z \\ e_{Rx} \\ e_{Ry} \\ e_{Rz} \end{bmatrix} = \sum_{b=1}^3 w_b \left( \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \begin{bmatrix} C_{i,j,k_x}^b \\ C_{i,j,k_y}^b \\ C_{i,j,k_z}^b \\ C_{i,j,k_{Rx}}^b \\ C_{i,j,k_{Ry}}^b \\ C_{i,j,k_{Rz}}^b \end{bmatrix} B_i^n(x) B_j^n(y) B_k^n(z) \right)$$

This equation calculates the predicted error at the specified (x,y,z) location for each of the three base vectors specified. Then these values are multiplied by the corresponding weights to interpolate what the error would be for the orientation of the sensor. Now that the errors are known, they are applied to the initial sensor measurement in order to compensate for them.

Position error compensation is simple since the error measurement corresponds directly to the (x,y,z) offset in millimeters.

$$\bar{e}_{pos} = \begin{pmatrix} x_{measured} \\ y_{measured} \\ z_{measured} \end{pmatrix} - \begin{pmatrix} x_{reference} \\ y_{reference} \\ z_{reference} \end{pmatrix}$$

To compensate for position error:

$$EF.P = toolPos - PosErr;$$

Therefore, to generate the compensated position:

$$\bar{P}_{corrected} = \bar{P}_{measured} - \bar{e}_{pos}$$



The orientation error is specified by a Rodriguez vector representing the axis of rotation between the ideal and measured readings and the magnitude of the angle of rotation such that:

$$\bar{e}_{ori} = \bar{\omega}\theta \frac{180}{\pi}$$

Where:  $\bar{\omega}$  is the unit vector pointing along the axis of rotation

$\theta$  is the angle of rotation between the reference and measured vectors  $\bar{n}$  measured in radians

When written out with the information available, we have:

$$\bar{e}_{ori} = \frac{\left( \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right)}{\left\| \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right\|} \arcsin \left\| \left( \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right) \right\| * \frac{180}{\pi}$$

This is accomplished with the following software commands:

```
(Fmeas.R*Fref.R.Inverse()).ExtractAngleAndAxis(tAngle,tAxis);
OriErr=(tAxis*tAngle*(180.0/PI_C));
```

To compensate for orientation error:

```
cisVec3 de = OriErr*PI_C/180.0;
```

$$de = \frac{\left( \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right)}{\left\| \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right\|} \arcsin \left\| \begin{matrix} \bar{R}z_{ideal} \\ \times \\ \bar{R}z_{measured} \end{matrix} \right\| = \bar{\omega}\theta$$

```
EF.R=cisRotation(de.Length(),
de/de.Length()).Inverse()*toolOri.Rotation();
```

Therefore, to generate the compensated rotation:

$$R_{corrected} = R_{error}^{-1} R_{measured}$$

Where:

$$R_{error} = \text{cisRotation}(\theta, \bar{\omega}) = e^{\hat{\omega}\theta} \in SO(3)$$

## Measurement Distortion Simulation

```
void CAurora_ToolsDlg::DistortTool(cisVec3 toolPos,
cisUnitQuat toolOri, cisVec3 *CorrectedToolPostion,
cisUnitQuat *CorrectedToolOrientation)
```

This function is very similar in functionality to `FindToolError()`. However, it is used to add the error on to an undisturbed reading, as opposed to correcting a distorted reading.

To distort the position:

```
EF.P = toolPos + PosErr;
```

$$\bar{p}_{distorted} = \bar{p}_{actual} + \bar{e}_{pos}$$

To distort the orientation:

```
cisVec3 de = -OriErr*PI_C/180.0;
```

$$de = \frac{-\begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{measured} \\ \bar{R}z_{ideal} \end{pmatrix}}{\left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{measured} \\ \bar{R}z_{ideal} \end{pmatrix} \right\|} \arcsin \left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{measured} \\ \bar{R}z_{ideal} \end{pmatrix} \right\| = -\bar{\omega}\theta$$

```
EF.R=cisRotation(de.Length(),
de/de.Length()).Inverse()*toolOri.Rotation();
```

Therefore, to generate the compensated rotation:

$$R_{distorted} = R_{distortion}^{-1} R_{actual}$$

$$\text{Where: } R_{distortion} = \text{cisRotation}(\theta, -\bar{\omega}) = e^{-\hat{\omega}\theta}$$

$$R_{distortion}^{-1} = \text{cisRotation}(\theta, \bar{\omega}) = e^{\hat{\omega}\theta} \in SO(3) = R_{error} \\ \text{(from previous section)}$$

Equating this result to the equation for distortion compensation:

$$R_{distorted} = R_{error} R_{actual} \Leftrightarrow R_{actual} = R_{error}^{-1} R_{measured}$$

## Simulation

### Simulation Algorithm

The simulation predicts error in the same way that compensation is usually performed. However, instead of subtracting off the error as before, it is added on to the measurements to artificially distort them in such a way that the measurements should be very similar to those that would actually be collected for such a scenario. Below is the mathematics behind how the simulator works for simulating the distorted position and orientation of a given sensor location.

#### Position Error

$$\bar{P}_{distorted} = \bar{P}_{actual} + \bar{e}_{pos}$$

#### Orientation Error

$$R_{distorted} = R_{distortion}^{-1} R_{actual}$$

Where:  $R_{distortion}(-\bar{\omega}, \theta) = e^{-\hat{\omega}\theta} \in SO(3)$

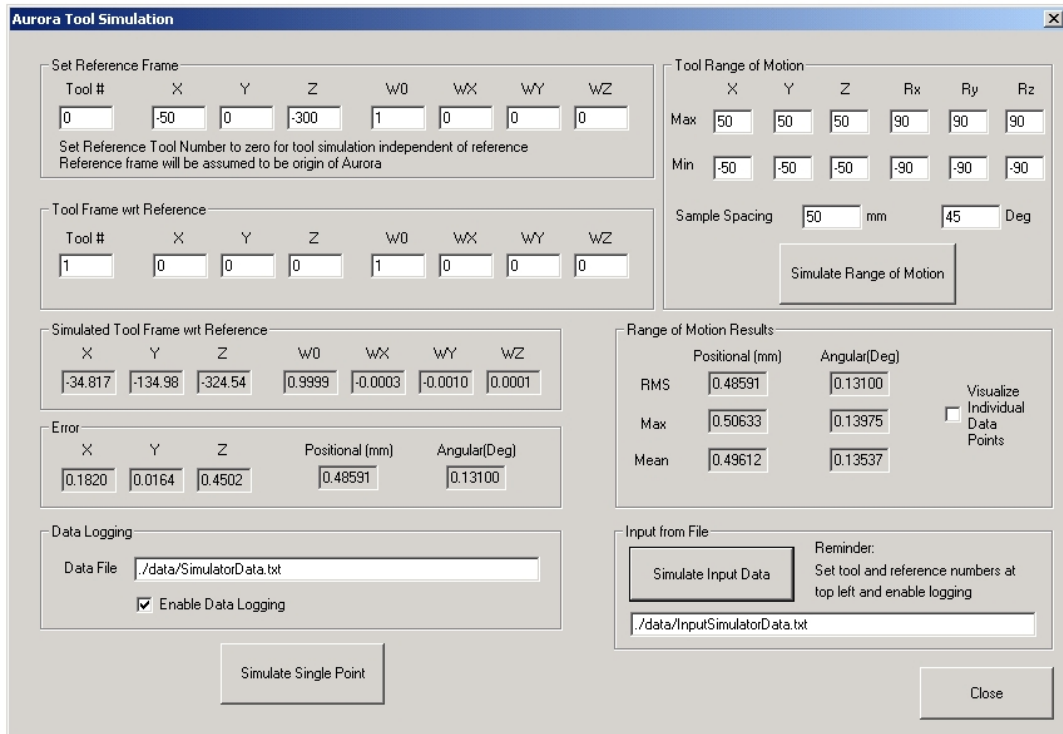
$$\theta = \arcsin \left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \right\|$$
$$\bar{\omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \frac{\begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix}}{\left\| \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \times \begin{pmatrix} \bar{R}z_{ideal} \\ \bar{R}z_{measured} \end{pmatrix} \right\|} \in \mathbb{R}^3$$
$$\hat{\omega} = skew(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in so(3)$$

Therefore:

$$R_{distorted} = e^{\hat{\omega}\theta} R_{actual}$$

## Simulation Results

Here some representative results from running the simulator are presented. Figure 5-7 shows a screen capture of the graphical user interface front end for the simulator software. The software allows for 3 modes of operation: simulating a single measurement, simulating a range of motion about a reference frame, and simulating data input from a file (either collected real data or synthetic data from Matlab, etc.). The simulation can be run using any pre-calculated polynomial file as selected in another menu.

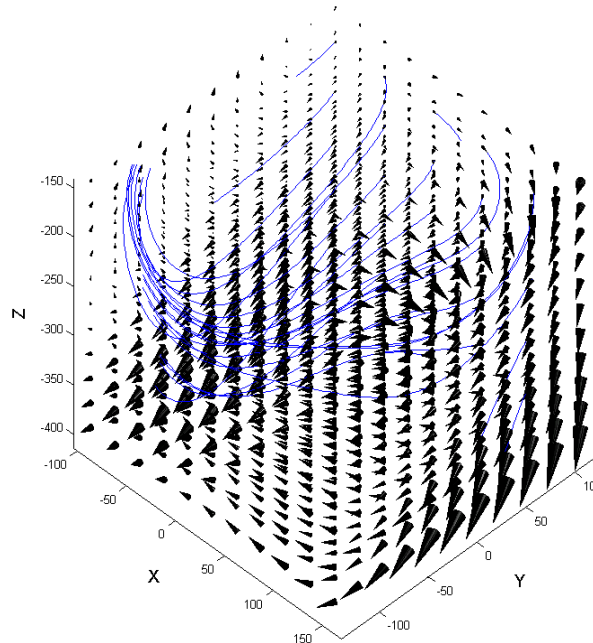


Screen capture of the Simulator's GUI

Figure 5-7

Figure 5-8 shows representative results of the simulator where the relative errors are displayed between a tool and a fixed coordinate reference frame (CRF). In this case, Traxtal 6-DoF dynamic reference bodies are used as both the CRF and the tool being tracked. One reference is fixed in the center, and the other is moved with respect to it; these results are for a constant orientation.

*Simulated Relative Position Error Between Dynamic References*



11x11x11 grid of simulated measurement errors of a 6-DoF tool with respect to a 6-DoF CRF for a single orientation

**Figure 5-8**

## ***Tool Design Using Simulator***

The simulator proves to be an excellent tool for design of electromagnetically tracked instruments. Since we can try out a tool and see how it performs without ever even building it, it is possible to experiment with a large number of designs. In general, the procedure for tool design is as follows:

1. Generate a CRF design (skip this step if one is already available)
2. Simulate the CRF design in the appropriate environment with respect to the EMTS base. Many different environments can be used for the experiments.
3. Analyze the results and decide if design satisfies requirements. If Yes, continue; if No, return to step 1.
4. Generate a Tool design
5. Simulate the CRF design in the appropriate environment with respect to the EMTS base. Again, many different environments can be used for the experiments.
6. Analyze the results and decide if design satisfies requirements. If Yes, continue; if No, return to step 4.
7. Simulate the tracking error of the tool with respect to the reference frame in a characterization of the target environment.
8. Analyze the results and decide if the designs satisfy the design requirements. If Yes, then the design is complete. If No, redesign the tool and/or the reference.
9. Build the reference and tool and compare the results.

This design procedure is followed later for design of ENT specific tools. See the ENT Specific Tool and Reference Design section for a detailed look at the process.

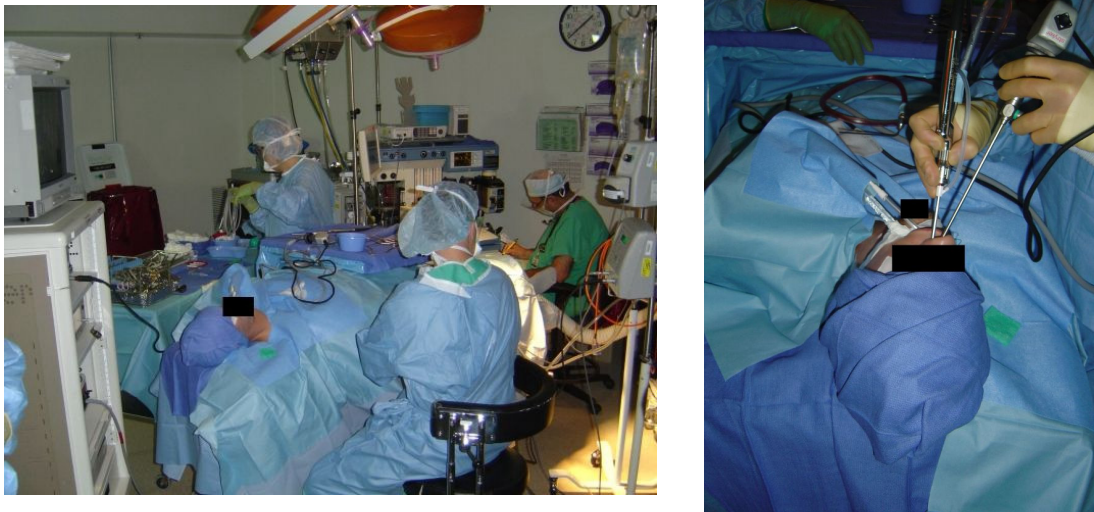
## ***Contributions***

This chapter presented what appears to be an entirely novel approach to the design of electromagnetically tracked tools. The method uses the distortion model from Chapter 3 in such a way that tools can be virtually tracked and measurements can be distorted as they would be in the actual environment. A software package was developed that allows for many features that are useful to the Aurora system. The software allows for data collection, tool frame fitting of any number of sensors, real-time measurement compensation, and simulation. The primary contribution was a system for tracker tool simulation that allows for rapid prototyping of tools without ever having to construct a tool or perform tedious, time-consuming data collection tasks until verification of a final design. In Chapter 6 we use this tool to aid in the design of a suite of electromagnetically tracked ENT instruments.

## Chapter 6: ENT-Specific Tool Design

### Overview

Ear, Nose, and Throat (ENT) surgery seems to be an ideal application for image guided surgery (IGS) procedures to take hold. This is due to the fact that almost all of these surgeries involve precise procedures in small spaces, though small openings, and with limited visibility. Also, due to the proximity to vital structures, such procedures can sometimes be rather risky. By tracking surgical instruments and providing visualization of their locations with respect to target anatomy, it may be possible to increase the safety, accuracy, and end-result quality. Figure 6-1 contains photographs that portray the typical configuration of an operating room (OR) for a sinus procedure.



Typical OR configuration for ENT surgery  
(From sinus surgery with Dr. Ishii at Bayview Medical Center)

**Figure 6-1**

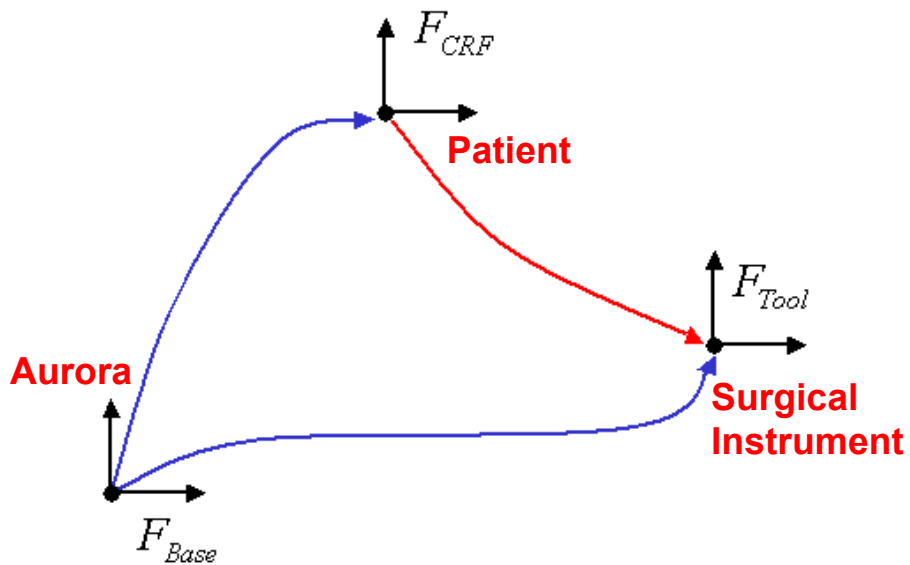


A realistic scenario for an ENT IGS surgery using an Aurora EMTS system is shown in Figure 6-2. The Aurora would most likely be placed up near the head as shown. Clearly, there is much to worry about as far as metals and other distortion causing elements as has been discussed in detail earlier. Therefore, careful consideration must be made in the placement of the Aurora FG and the design of the tools. The following image, Figure 6-3, shows the frame transformations required for tracking a tool with respect to the patient-fixed coordinate reference frame (CRF).



Synthetic image showing the NDI Aurora in a realistic tracking scenario in the surgical field

Figure 6-2



Important frame transformations for tool tracking with respect to a patient-fixed reference

Figure 6-3

## Literature and Product Review

In the introduction there was an overview of IGS systems and tracking devices that could be used for them. The focus here is to explore the commercially available IGS systems that specialize in cranial and ENT surgery, in particular the ones based on electromagnetic tracking technology. All images are directly from manufacturer web sites.

Starting off with the NDI Aurora, although custom IGS solutions exist, the only commercial system involving this technology to date is produced by Brainlab (Heimstetten, Germany). This system was implemented as the EM version of their Kolibri system, shown in Figure 6-4. Some of the salient features of the Brainlab systems are the ease of portability (in particular the Kolibri), and a unique method of headset-free registration involving the z-touch laser scanner.



Figure 6-4

One of the leaders in the field of image-guided surgery is Medtronic. Their key contribution to the field is the StealthStation. The system is primarily an optical system like the Brainlab systems, but they have a much wider array of track instruments; it even includes mounts for the ENT drill that was used for our experiments. Their contribution to the EM tracking field is their own EMTS, the AXIEM, shown in Figure 6-5. This interfaces directly with the StealthStation TREON to allow for EM tracking with the same familiar user interface.



Figure 6-5

<sup>1</sup> Information about Brainlab IGS systems is available at <http://www.brainlab.com>

<sup>2</sup> Information about Medtronic SNT IGS systems is available at <http://www.stealthstation.com>

General Electric (GE) has a dedicated electromagnetic system for ENT surgery called the ENTrak, shown in Figure 6-6. In addition, they have a more general purpose, but similar, device called the Instatrak. Although there is sparse detail, GE claims to have a proprietary algorithm where the system automatically detects and compensates for metal in the field called MagneticIntelligence. Unfortunately, the specifics of the algorithm are not publicly available. One key feature of the GE system is the use of special headsets that allows for automatic registration of the patient. This registration is accomplished with EM-tracked headsets that can be repeatedly replaced on the patient, and therefore can be attached for preoperative imaging, and again during the procedure.



**Figure 6-6**

Clearly, this is not an exhaustive list. As mentioned in the introduction, there are several companies that produce stand-alone EM tracking units, and it is possible to use these systems for custom IGS solution. Further, there are other surgical navigation systems such as the J&J Biosense Webster CARTO XP that use the technology, but not for ENT applications. Also, there are other ENT IGS systems that do not offer EM versions. Two examples of these are the Radionics Omnisight system with the SinuSight ENT module and the Stryker ENT navigation module.

<sup>1</sup> Information about General Electric(GE) IGS systems is available at <http://www.gehealthcare.com>

## ***ENT Specific Tool and Reference Design***

### **Overview**

In this section, the tools presented above are used to develop a set of instruments for use in ENT surgery. These instruments are designed and simulated using the analytical expressions and the simulation software. The tools are then fabricated and tested experimentally. The results of the actual experiments and a comparison with the predicted results are presented.

The primary tools being designed are:

- Coordinate reference frames(CRF)
- Tracked tissue shaver/debrider/drill
- Tracked 5 DoF tools and pointers
- Tracked endoscope

These tool designs are being restricted to having two sensors per tool such that the Aurora system can handle them directly without custom user software for frame fitting. Clearly, having more sensors would improve tracking accuracy and there will be some experiments with such designs. Further, the basic designs assume we are tracking a tool with respect to a reference, but not as a specific pair. Although probably not practically feasible in a surgical setting, better tracking results can be obtained if the reference RB is optimized for a particular tool RB where the sensors have similar orientations. This is especially true if a technique such as Thin-Plate Spline Measurement Compensation is used to improve tracking accuracy.

## Cranial Reference Body

The key to accurate registration and tracking is a well-designed coordinate reference frame (CRF). This frame is rigidly affixed to the patient and serves as a way of tracking the tools with respect to the patient anatomy. For ENT applications, this CRF is affixed directly to the patient's skull; this can be either by a semi-permanent fixation with screws or by some method of repeatable replacement of the device. This work details the accuracy associated with tracking a tool with respect to the CRF only. An additional critical source of error that must be considered, not in the scope of this work, is registration of the patient's pre-operative images to this frame. There has been much research in this field. One particularly applicable example is [Knott, 2004], which deals with the problem of headset-based registration for sinus surgery.

As mentioned before, design is restricted to two sensors per rigid body because the current commercial Aurora system will only fit a tool to sensors on a single tool port, which can manage only two sensors. Larger numbers of sensors have a dramatic increase in accuracy, and they can be used with my software; however, tools with more than two sensors will not work for general Aurora applications.

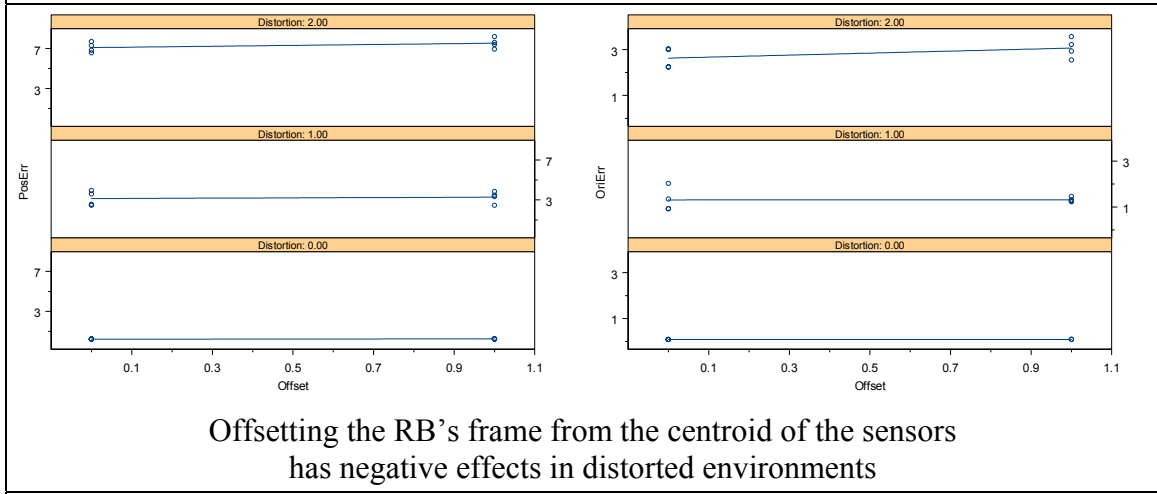
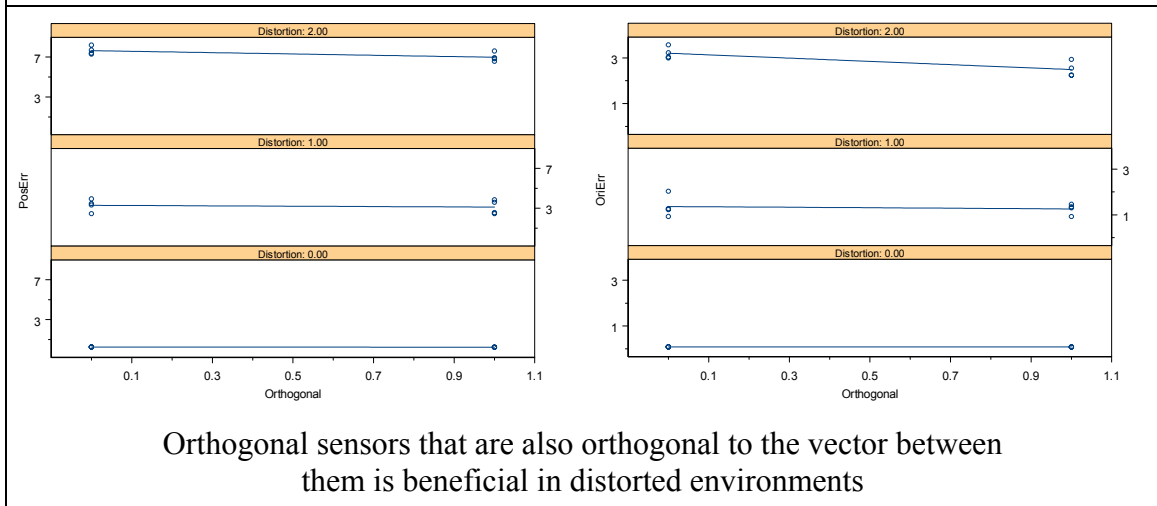
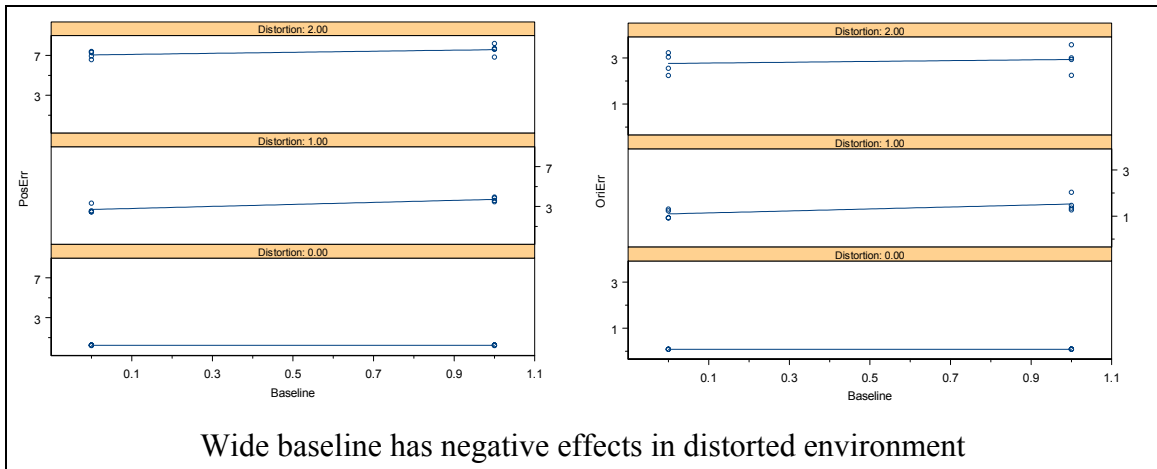
With only two sensors in a tool, the configuration parameters that can be adjusted are as follows:

- The baseline between the sensors
- The offset of the C.G. of the sensors from the tool's frame
- The relative orientations of the sensors

Simulations were performed with eight different, practical tool configurations that can be used to test each combination of the above-mentioned parameters independently. Table 6-1 lists the tool configurations that were tested in this experiment. Results showing the trends resulting from simulations are shown in Table 6-2.

<b>Configuration Number</b>	<b>Sensor 1 Direction</b>	<b>Sensor 2 Direction</b>	<b>Baseline (X Direction)</b>	<b>C.G. Offset (Y Direction)</b>
1	Y axis	Z axis (1)	150 mm (0)	None (0)
2	Y axis	X axis (0)	150 mm (0)	None (0)
3	Y axis	Z axis (1)	200 mm (1)	None (0)
4	Y axis	X axis (0)	200 mm (1)	None (0)
5	Y axis	Z axis (1)	150 mm (0)	75 mm (1)
6	Y axis	X axis (0)	150 mm (0)	75 mm (1)
7	Y axis	Z axis (1)	200 mm (1)	75 mm (1)
8	Y axis	X axis (0)	200 mm (1)	75 mm (1)
Tool configurations simulated for CRF design				

**Table 6-1**



Investigation of the effects of several variables on the position and orientation error in tracking a rigid body, conditioned on the magnitude of the field distortion

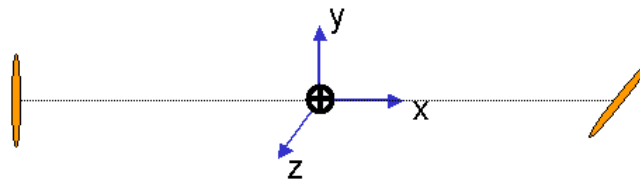
Table 6-2

The results shown in Table 6-2 represent the output from the simulator for each tool design listed. Three plots are shown for each position and orientation error pair; each plot investigates the effect of one parameter on the respective error. Each parameter is encoded as either a zero or a one as shown in Table 6-1. The simulation was performed for each tool design in each of three characterized environments; ‘ideal’ NDI Set #1 (0), disturbed NDI Set #4 (1), and very disturbed OR data set (2). The results in the table show trends in errors associated with each of the three parameters conditioned on the level of distortion.

Fortunately, it appears that the behavior of position and orientation errors is similar with respect to these parameters. This allows us to determine the optimal configuration without having to deal with a trade-off between position and orientation accuracy. It is clear that:

- A wider baseline in a distorted environment is problematic. This is in opposition to intuition and the earlier analysis, but is due to the curvature of the distortion field. The original predictions hold in the ‘ideal’ environment in that a wider baseline is better due to the more isotropic nature of the error.
- The sensors should be orthogonal not only to the selves but also to the offset between them.
- The centroid of the sensors should ideally lie on RB frame’s origin. If this is not possible, the RB frame and area of interest should be as close as possible to the C.G. of the sensors.

By following these basic guidelines, we come up with a sensor configuration like that shown in Figure 6-7. Here, the sensors are aligned with the y-axis and the z-axis, while the offset between them is aligned with the x-axis. Further, the RB’s frame is right at the C.G. of the sensors. The baseline is determined more by practicality than optimizing the tracking error. This is to be designed as a cranial reference, and therefore the two sensors should be spaced such that they are about the width of a skull apart (approximately 150mm).

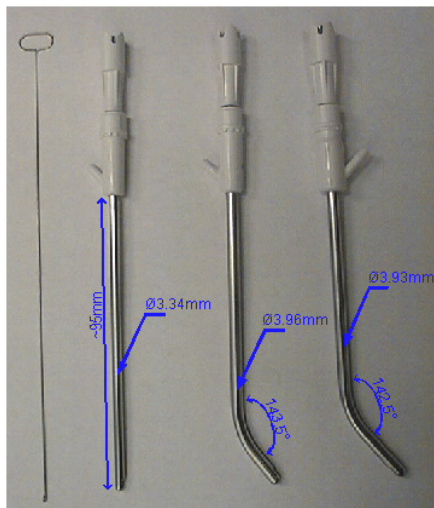


Optimal sensor configuration for 2-sensor CRF  
(Configuration #1 from Table 6-1)

**Figure 6-7**

## Tissue Shaver

The tissue shaver is a critical instrument for performing ENT procedures. It allows for access and tissue removal in very hard to reach places that conventional instruments would have a very tough time being manipulated in. In particular for throat and sinus surgery, shavers such as these are used extensively; Figure 6-8 shows three types of shaver bits used for sinus surgery; longer but otherwise similar versions are used for throat surgery. However, since these tools do allow tissue removal in hard to reach and hard to visualize spaces, they are ideal candidates for tracking and use with IGS systems.



ENT tissue shavers and drills  
(Medtronic / Xomed)

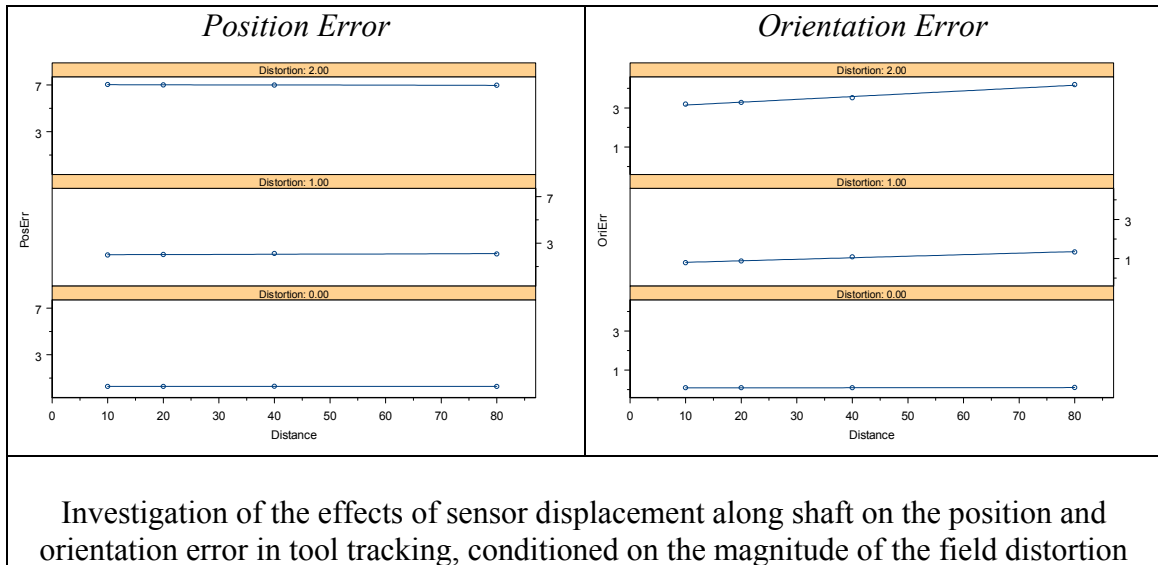
**Figure 6-8**



Prototype tracked tissue shaver with  
two attached 5 DoF sensors



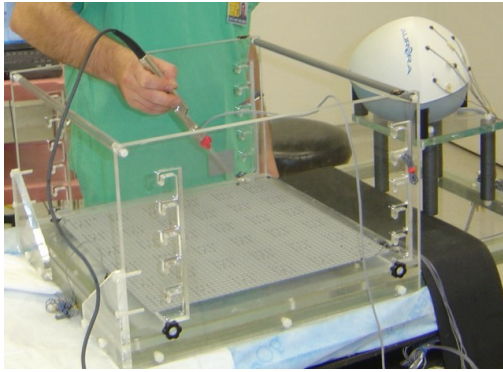
Unlike the CRF, which we were able to design from scratch, sensor placement for this tool is subject to constraints due to the shape of the tool. As we have learned, we want to keep the centroid of the sensors as close as possible to the point of interest on the tool, in this case the tip. Therefore, we know that we must keep one sensor right near the tip of the tool. Since we cannot increase the width of the tool by any appreciable amount, the first sensor must lie right on the surface of the tool as close to the tip as possible. The second sensor is where we have some more freedom. Since we know that we want the sensors to be as close to orthogonal to each other as possible, the other sensor must lie on the shaft past the bend (or on an offshoot which could make the two sensors orthogonal, but could make the tool too awkward to use). So, the one parameter to tune is the spacing of the second sensor along the shaft.



**Table 6-3**

Table 6-3 presents the results from a simulation experiment with the tissue shaver. The simulation was performed for each tool design in each of three characterized environments; ‘ideal’ NDI Set #1 (0), disturbed NDI Set #4 (1), and very disturbed OR data set (2). As with the CRF design, the tool was virtually moved in a 200mm cube centered about the center of the characterized volumes by increments of 25mm. At each location, the orientation was simulated at  $\pm 90^\circ$  about each axis by increments of  $45^\circ$ . By examining the results, it appears that the overall trend is to decrease the baseline to improve the tracking accuracy; this is especially true for the orientation error of the tool. Therefore, the optimal design would have the two sensors situated such that one is as close to the distal tip of the tool as possible, and the other as close a possible on the opposite side of the bend.

Unfortunately, the relative locations and orientations of the sensors on the tool are not the only factor affecting the tracking accuracy. By doing some experiments in the OR with the instrumented tissue shaver bit and the actual drill handle, it became evident that there was significant additional distortion present. In Figure 6-9, on the right, is the tool in use in an actual sinus procedure. On the left is an experiment where the goal was to determine how accurately the shaver can be tracked in a realistic scenario.



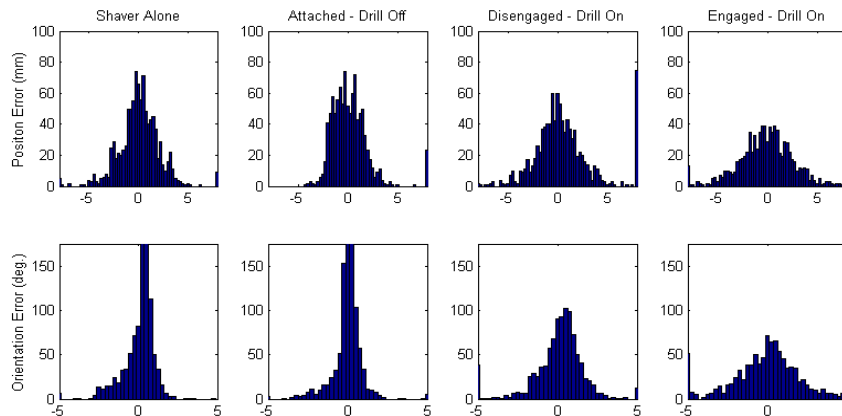
Shaver accuracy experiments



Shaver in use in OR

**Figure 6-9**

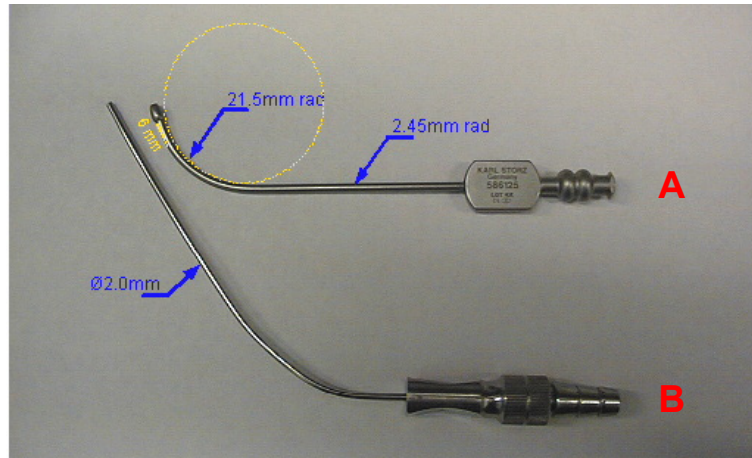
The results from the experiment shown above are presented in Figure 6-10. Here, the histograms represent the distribution of error in the relative position and orientation of the two sensors on the tool from the mean. The trial is based on 1000 measurements collected as the tool was moved in a realistic pattern in the working volume of the Aurora. The trial was repeated in four scenarios: 1) the shaver bit alone, 2) the shaver bit attached to the drill but not turned on, 3) the shaver attached to the drill with the drill running but the bit not spinning, and 4) the bit attached and engaged to operating drill. As can be seen, there is not a terrible change in error as the bit is coupled with the drill handle, but the error quickly becomes much worse as the drill is turned on. The obvious way to perform tracking that is somewhat immune to these problems is to make sure that the tool is aligned before running the drill. For confirmation, the drill must be turned off. Since the drill is operated with a foot pedal and is normally paused to check the progress, this is not a significant impedance to the standard workflow.



Relative error between two sensors on tissue shaver under various conditions in OR  
**Figure 6-10**

## Suction Devices and 5 DoF Pointers

Suction devices, as shown in Figure 6-11, are very often used in ENT surgery. Not only are they used to clean up the surgical site, but often these devices end up being used as pointers to help the surgeon navigate in the sinus cavity. These devices would be especially useful if they were tracked such that they could be used in IGS applications.



Common suction instruments

**Figure 6-11**

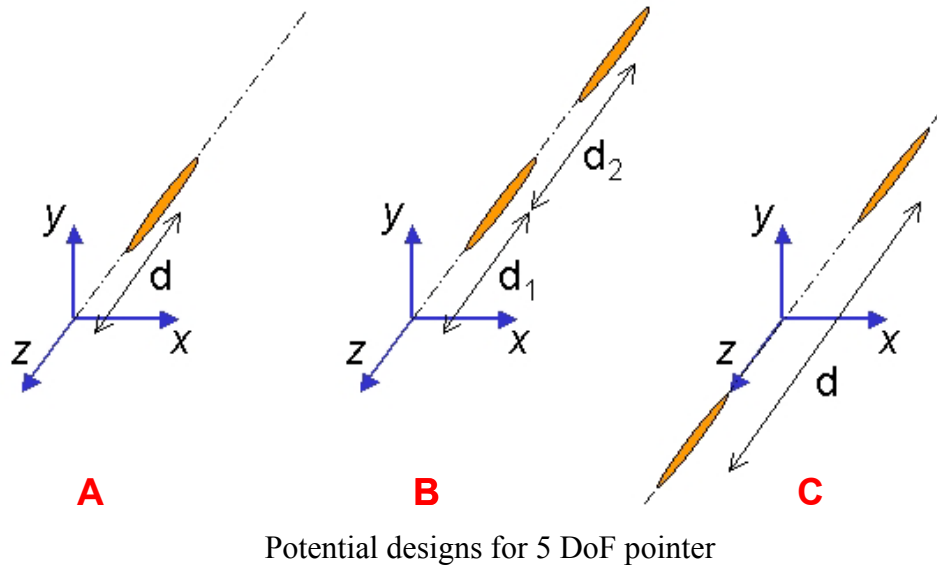
The two devices shown in Figure 6-11 require different designs. Tool 'A' can be tracked similarly to the tissue shaver bit as mentioned before, so it will not be discussed here. The interesting tool here is Tool 'B.' The reason it is of interest is because it really only needs to be tracked with 5 DoF and is not only representative of suction devices, but also pointers, needles, catheters, and other 5 DoF tools.

The following sketch in Figure 6-12 represents possible tracking solutions for a 5 DoF tool. The configurations are: A) a single sensor at a given displacement along its axis from the tip, B) two aligned sensors that are a given distance from the tip with a given spacing, and C) a less practical solution with two aligned sensors spaced evenly about the target in the center.

The goal is to determine:

- The advantage of having two aligned sensors vs. a single sensor
- The effect of distance of sensors from target / tool tip
- The effect of sensor baseline length (sensor spacing)

The details of the configurations used in the trial are presented in Table 6-4. Each configuration was simulated in two environments: OR and moderately disturbed NDI Set #4. In the simulation, the tool was moved in a 200mm cube centered about the center of the characterized volume by increments of 25mm. At each location, the orientation was simulated at  $\pm 90^\circ$  about each axis by increments of  $45^\circ$ . Accompanying the configuration are representative results taken from simulations with the moderately disturbed data set.



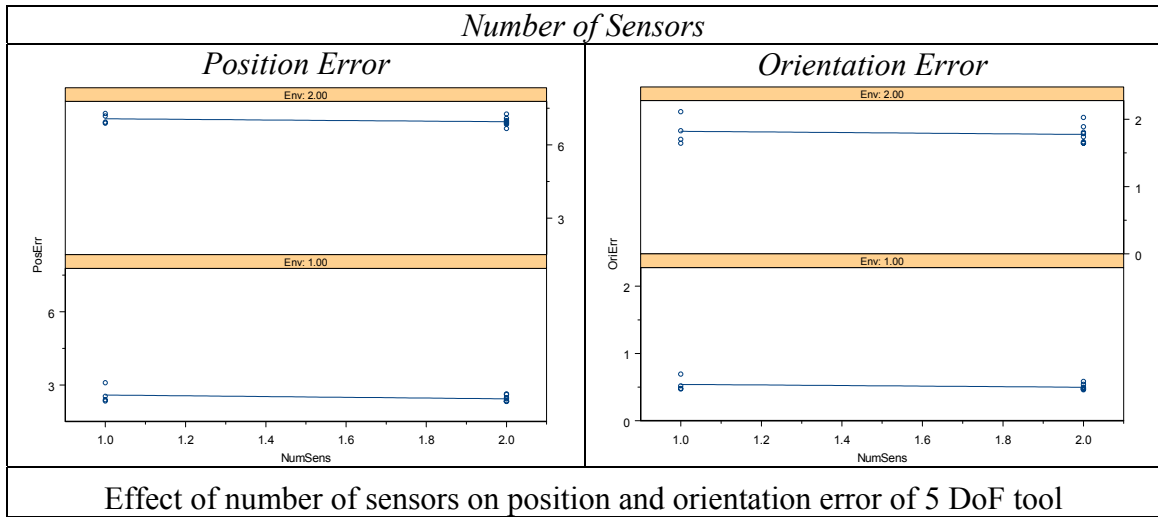
**Figure 6-12**

Configuration	# of Sensors	$d(d_1) / d_2$	RMS Pos. Error (mm)	RMS Ori. Error (deg.)
A-1	1	0 / NA	2.342	0.471
A-2	1	10 / NA	2.392	0.480
A-3	1	25 / NA	2.537	0.519
A-4	1	50 / NA	3.093	0.692
B-1	2	10 / 10	2.428	0.488
B-2	2	10 / 20	2.474	0.499
B-3	2	10 / 40	2.609	0.541
B-4	2	10 / 80	2.632	0.585
C-1	2	10 / NA	2.340	0.470
C-2	2	20 / NA	2.336	0.467
C-3	2	40 / NA	2.326	0.459
C-4	2	80 / NA	2.333	0.479

Tool configurations simulated for 5 DoF pointer design  
Results from simulation in disturbed environment (NDI Set #4)

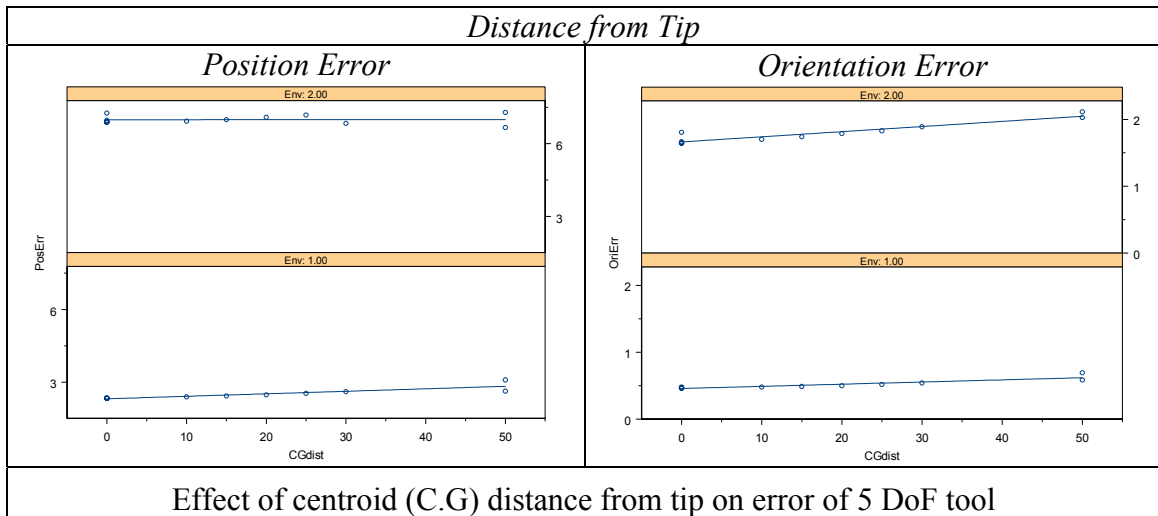
**Table 6-4**

The following plots help us to investigate how changing the above parameters affects the tracking error. Each plot shows the respective error plotted against the parameter of interest. The plots are conditioned on the distortion level of the environment, where  $Env=2$  denotes the highly disturbed OR and  $Env=1$  denotes the moderately disturbed NDI Set #4. The first results, shown in Table 6-5, show how tracking error is dependent on the number of sensors on the tool. There is a clear improvement in the tracking accuracy in both the tip position and the tool axis for two coaxial sensors over a single coil. This is an expected result; clearly more sensors should improve the accuracy.



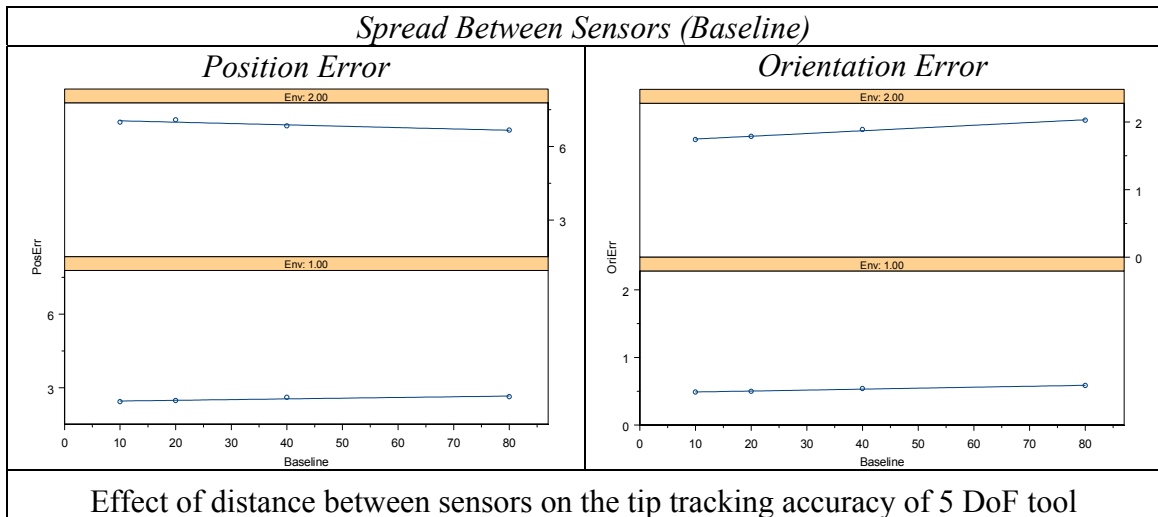
**Table 6-5**

The next trial studies the effect of distance of the sensors from the tracked tip of a tool on the tip tracking error. The measure here is the distance of the centroid (C.G.) of the sensors from the tip of the tool that is being tracked. If there is a single sensor, then the C.G. is the origin of that sensor. Again, as can be seen in Table 6-6, we get an expected result in that the error in both position and orientation (orientation especially) is positively correlated to the distance of the sensors' centroid from the tip.



**Table 6-6**

The final key result is to study the effect of a wider sensor baseline on the tracking accuracy of a 5 DoF tool. Table 6-7 shows the effect of the increasing the distance between the sensors on the error for configuration ‘B’ only. As with all of the other plots, this is conditioned on distortion environment. This is the first result in this experiment that appears to have a strong interaction between the levels of measurement distortion and the error pattern. For most of the data, there appears to be an increase in error as the baseline is increased. This is in general counterintuitive for trackers, but correlates well with earlier studies. This is attributable to the fact that if the curvature of the error field is high, then the increase in accuracy by widening the baseline is countered by the larger differences in distortion between the two sensors. This is the case only for distortion fields with high curvature; as I discussed earlier, the greater the distortion magnitude, the smoother the field is in general. This could explain why the translational accuracy actually gets better in the higher distortion field as the baseline is increased.



**Table 6-7**

## Endoscope

Due to the small spaces and depths at which it is necessary to work, visualization without assistance is almost impossible for many ENT procedures. Therefore, the endoscope is the key to allowing the surgeon to see what is going on at the surgical site. However, manipulating an endoscope still requires significant caution and intimate knowledge of the anatomy. By tracking the endoscope with respect to the anatomy, the surgeon can know exactly where the tip of endoscope is and exactly where it is looking. In addition, with a tracked endoscope it is possible to perform camera calibration and reconstruct the internal surfaces; this information can be used for registration and other navigation-related purposes. Images of the actual device and a possible prototype tracked model are shown in Figure 6-13. This prototype shows a design using custom coils that would be the optimal design with regards with respect to ergonomics, but the large distance of the proximal sensor to the tip may cause poor tracking results.



Standard 45° ENT endoscope

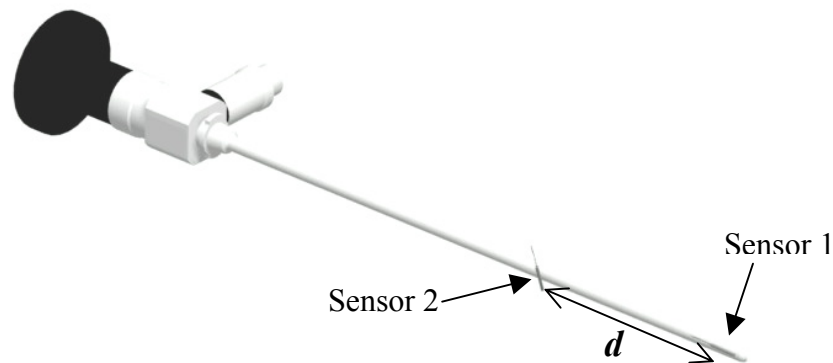


CAD model of a tracked endoscope  
using custom coil designs

**Figure 6-13**

At first glance the endoscope looks like it would be an easy tool for design of a tracked prototype. It looks this way because there are very few possible combinations of sensor poses on the tool that can be obtained. However, this is misleading because the obvious poses (one at the tip and one on the light input shaft) produce a very poor tool design. By placing the sensors here, the tool has a very high aspect ratio. By this I mean that the ‘inertia’ of the sensors is high about two axes, but very low about the axis of the tool. This means that we would have very poor rotational accuracy about this axis; this would be an enormous problem because it would mean that the endoscope is not looking where we think it is. If we were using the endoscope to perform vision-based registration, then our registration error would be very high.

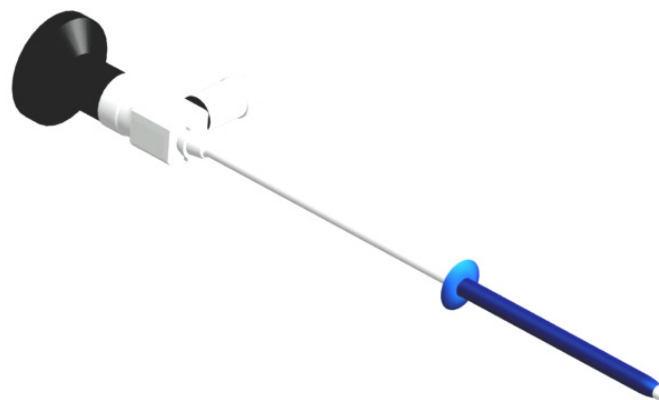
Optimal design of the tool means that we want to 1) keep the centroid of the sensors as close to the tip as possible, and 2) we want to keep the ‘inertia’ of the sensors high along all of the principal axes. The ‘optimal’ design would have the first sensor parallel to the shaft at the distal end, and the second sensor a short ways up and orthogonal to the shaft. If the endoscope is not going to be inserted far into a cavity, then we can get away with having an arm jut out of it. However, this may not be feasible for a general-use surgical tool and we will have to stick with putting the second sensor further up.



CAD model of sensor configuration for tracked endoscope

**Figure 6-14**

Figure 6-14 shows the ‘optimal’ sensor design for the endoscope. This design is optimal in that it allows for accurate tracking while maximizing the ‘inertia’ in a compact space. Similarly to the tissue shaver, better results would be achieved if *Sensor 2* from the figure is shifted closer to the distal tip and *Sensor 1*. However,  $d$  must be kept large enough such that the second sensor will not interfere with the surgery. In order to decrease the obtrusiveness, a smooth sleeve with the sensors molded inside could be slipped over the tip of the tool. This would allow for minimal increase in width at the tip (~1mm) and a wider home for the second sensor further up the shaft. See Figure 6-15 for a possible solution for this design, which could be a sterile, tracked disposable sleeve for endoscopes and other tools.



Potential solution for enclosing two 5 DoF sensors in a sleeve

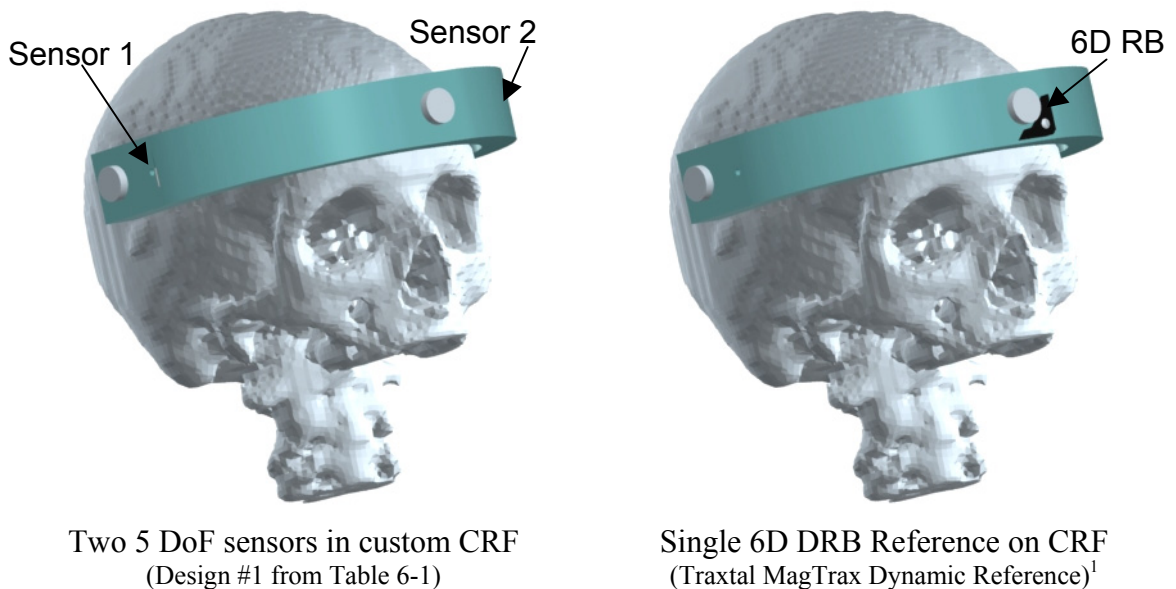
**Figure 6-15**



## Verification of Tool Design

To validate the previous tool design techniques, an experiment was performed to investigate the tracking accuracy of a tool with respect to the designed CRF. For comparison, the designed CRF was matched up against the standard NDI/Traxtal 6D MagTrax Dynamic Reference Body (6D DRB). In this experiment, the simulator was used to gauge the accuracy of tracking a tool with respect to the reference body. This scenario was simulated using the above designed CRF (Design #1 with a baseline between the sensors of 150mm) and the 6D DRB placed at a distance of 75mm from the reference frame origin. This way, all sensors in both designs are placed along a 75mm radius circle surrounding the operating area. The two configurations are shown in the CAD models in Figure 6-16.

*Reference Frame Configurations Compared*

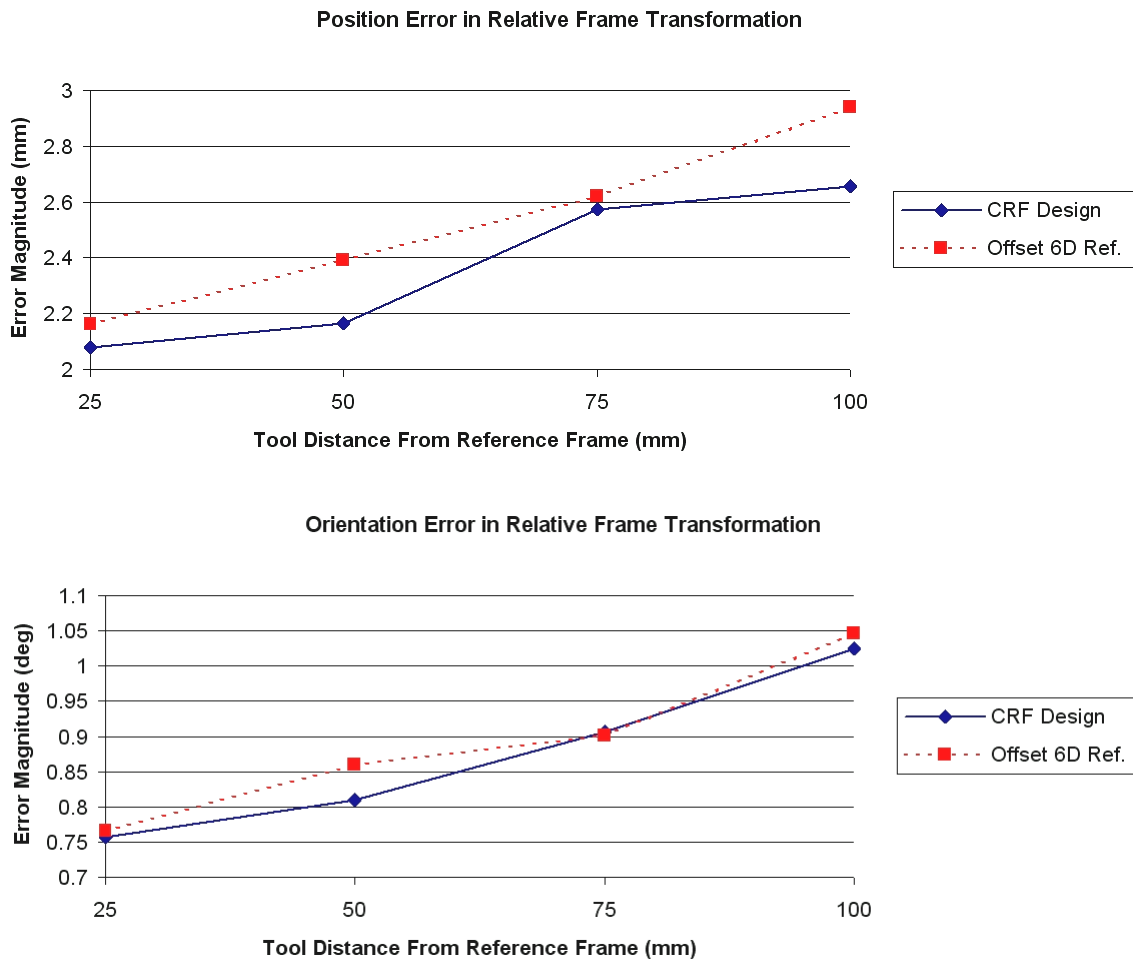


**Figure 6-16**

<sup>1</sup> Information about the Traxtal MagTrax Dynamic Reference Body is available at <http://www.traxtal.com>

The following protocol was used for both reference frames. In order to make a fair comparison, both were used to track another NDI/Traxtal MagTrax 6D DRB as the Tool. Error was gauged by the difference in relative position and orientation between the Tool and the Reference as measured and as input into the simulator. The reference RB was aligned in 5000 random poses within a 100mm cube surrounding the center of the Aurora's working volume. Then, the tool was placed in a random, but known, poses with respect to the reference frame. To judge how well the references work for different sensor distributions, the sensor measurements were generated four times with different bounds on their positions with respect to the reference frame's origin each time. The results are shown in Figure 6-17. They seem quite positive in that the frame designed using the help of the simulator outperforms the off-the-shelf 6D RB in all cases. These simulations were performed using the 3<sup>rd</sup> order model of the reasonably disturbed data set (NDI Set #4). Similar results are obtained for other environments.

*Comparison of Designed CRF to NDI/Traxtal 6D Reference*



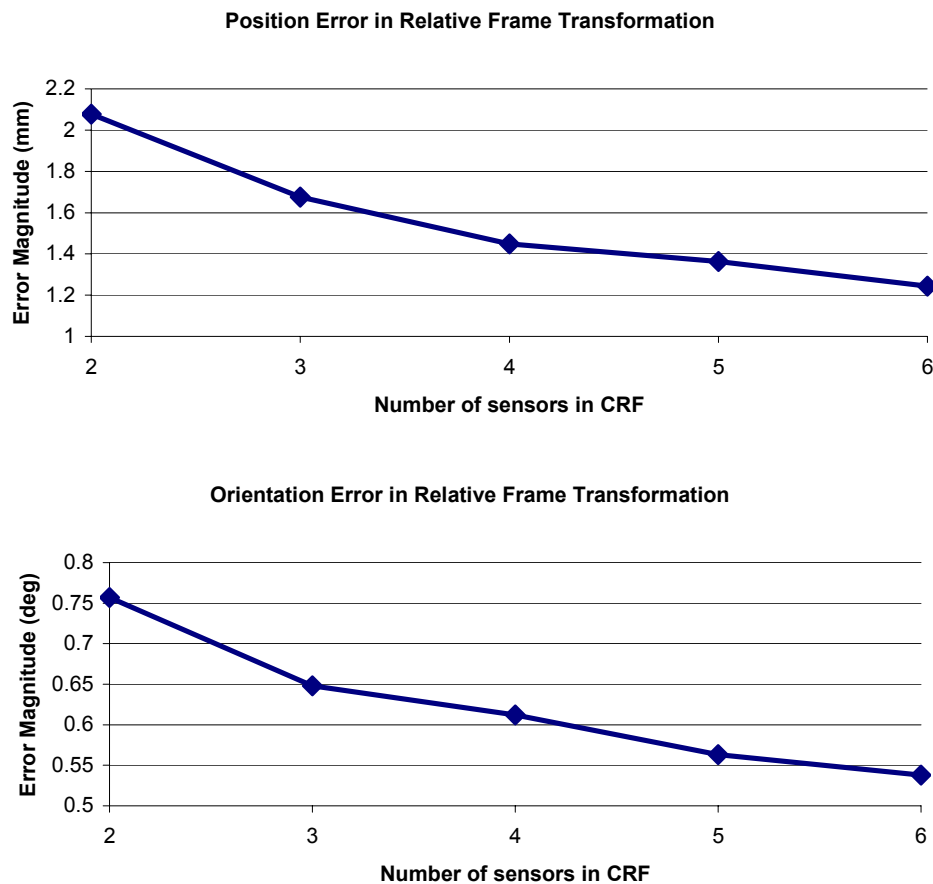
Error in relative transformation between Reference and Tool frames vs. bound on spread of Tool distances with respect to Reference

**Figure 6-17**

As stated at the beginning of this section, the tools designed were restricted to two sensors because this is the maximum that is natively allowed by the Aurora system. However, my algorithms presented here do not restrict the number of sensors. So, this experiment studies how the accuracy is affected by adding sensors to the reference frame.

The experimental protocol is the same as for the previous trial except that this trial is simulated in the reasonably disturbed NDI Set #4 environment. A set of 5000 random poses within a 100mm cube surrounding the center of the Aurora's working volume was assigned to the CRF being tested. Then, the tool (Traxtal Magtrax DRB) was placed in random, but known, poses with respect to the reference frame. The CRFs tested here have {2,3,4,5,6} sensors. They are all aligned with principal axes and placed along a 150mm diameter sphere surrounding the CRF origin. The first configuration is identical to the one above (CRF Design #1). Further configurations add sensors to this base. Clearly, more sensors have a rather dramatic positive effect on the tracking accuracy. In particular, the four sensor reference looks like a good tradeoff between RB complexity and accuracy.

*Tool Tracking Error vs. Number of Sensors in CRF*



Error in relative transformation between Reference and Tool frames vs. number of sensors in the Reference RB

**Figure 6-18**

## ***Contributions***

This chapter used just about everything that had been discussed previously to aid in the design of a set of real instruments and reference bodies for electromagnetically tracked, image guided surgery applications. These tool's basic designs were constructed based upon intuition gained through experience and the analytical tool design section of Chapter 4. Then these designs were varied along several parameters and the optimal design was determined based on simulation in several environments that were mapped using the techniques of Chapters 2 and 3. As a final verification, a standard tool was tracked with respect to both the designed reference and an off-the-shelf reference body. The errors consistently appeared to be smaller when tracking a tool with respect to the custom-designed reference frame.

## Chapter 7: Conclusions

### **Contributions**

This document has covered many aspects of the field of image guided surgery (IGS), and in particular the use of electromagnetic (EM) trackers in IGS applications. The document first gives an overview of the potential tracking solutions and then focuses on the EM tracking scenario. The goal was to optimize the use of EM trackers such that they can function with improved tracking accuracy.

The first key result was characterization of the measurement errors induced by magnetic field distortion for the Aurora system. Our approach was based on that of Wu *et al.* [Wu, 2003c]. The main extensions have been significant improvements in the robustness and efficiency of the implementation and extensive experimentation, together with new designs for fixtures and calibration phantoms. Chapter 2 discussed how to calibrate a hybrid or digitized data collection system, and then how to use this system to gather synchronized sensor measurements and reference measurement. Using these, the field distortion was determined. Distortion was investigated for many environments to give a useful comparison of the field distortion of the Aurora system in these conditions. The work here can be generalized to other EM trackers, and in fact just about any type of tracking system.

The second key result was modeling the distortion field and use of the models for real-time compensation of errors to improve tracking accuracy. Again, the basic approach was that of [Wu, 2003c]. Extensions again were primarily directed toward robustness, efficiency, and experimentation. Chapter 3 detailed methods for modeling the measurement error with Bernstein polynomial models. The main contribution was to determine how to find the optimal parameters to use for the data collection and model generation such that a high quality model can be created with a practical data collection. These parameters are based on the environment and the accuracy requirements.

Our final key result was development of analytical and empirical design of tools using EM tracking devices. Prior analytical approaches (e.g., [West, 2004] and [Fitzpatrick, 2001]) for designing tracked tools generally ignore the field distortions and pose dependencies found with EM tracking devices. Further, these approaches are based on 3 DoF sensors, while the Aurora sensors have 5 DoF. In Chapter 4 we detailed the generation of analytical bounds on the errors and determined the propagation of measurement uncertainty through frame transformations. Using this information, we determined how to maximize the tracking accuracy of a tool with respect to a reference frame as is necessary for IGS applications. For empirical tool design a simulator was created as described in Chapter 5. This simulator allows the user to place arbitrarily designed virtual EM tools and references in a model of any characterized environment and simulate the tracking accuracy of the combination. This is useful because it allows testing of many tools designs rapidly since the physical tool need not be tested or moved

around an actual measurement field. Actual tools, specifically for ENT IGS applications, were presented in Chapter 6.

Overall, we have covered just about all of the key areas of tool design and error analysis for electromagnetic tracking applications, and in particular for image guided surgery applications. We hope this reference proves useful to those in the field as an aid for system characterization and tool design.

## ***Future Development***

This research has an incredible breadth and can in no way be exhausted in a single work. There is much work that can be done to both further my initial work here, and to progress into new topics. The following represent some of the areas that I feel would be the next crucial projects to tackle.

### **Interactive OR Field Characterization**

It would be ideal if the working volume of the EMTS could be characterized just prior to an operation when all of the equipment is in place just as it would be for the procedure. In order to realistically be able to compensate for the field distortion in an operating room, however, it is essential to have a rapid method of data collection. The current methods of calibration are either very time consuming or require impractical amounts of additional equipment. One possible solution would be a specialized data acquisition collection device such as a small plastic robot. Another would be to build a fixture that allowed for a large amount of data to be collected simultaneously and only needed minimal user interaction and time. These solutions will be investigated in the future.

### **Updated Field Distortion Model**

The current methods for modeling the magnetic field distortion are based on a Bernstein polynomial basis. This type of model produces a very good global representation of the error. However, an adaptive field distortion map that subdivides the space to allow for local distortion models of higher order than the global model may be superior for error fields with high curvature. Using B-splines and other techniques will be explored. Further, I believe that the local distortion model provided through the use of Thin Plate Splines may be a very useful tool for improving tracking accuracy without the need for a full field characterization. This should be investigated much closer to determine if it can in fact be of assistance.

### **Further Tool Design**

Obviously, the last section of this document only detailed the development of a small subset of instruments for use in ENT surgery. Other surgical tools are prime targets for EM tracked IGS applications, and hopefully this work will be of great assistance in their development. In addition, ENT is clearly not the only client for EM tracked instruments; other application such as neurosurgery, arthroscopy, reconstructive surgery, percutaneous therapy, and joint replacement are prime candidates. In order to get optimal accuracy, in addition to the tools, a CRF should be designed specifically for each application site.

## Image Guided Surgery Software

Although we present detailed methods and results of distortion mapping and error compensation, such methods will not be practical for use in a clinical setting unless they are well-integrated into an IGS software suite. Therefore, tracking, data collection, field characterization, measurement compensation, frame fitting, tool design, visualization, and registration software must all be combined into a fully integrated software solution for IGS.

## Robot Interaction

The next generation of surgical equipment will undoubtedly involve robotics in some form or another. Shown in Figure 7-1, two systems in development at the CISST ERC for ENT surgery are: 1) the Steady Hand Robot (SHR) for cooperative control of a tissue shaver for sinus surgery applications, and 2) the Snake Robot for performing intricate procedures down in the depths of the throat. Both of these applications will require knowledge of where exactly the tools are with respect to the patient, and the use of EM tracking technology may be the key. In addition to tracking robot-assisted surgery, another hurdle to address is how to dynamically compensate for field distortion of a moving metallic device in the working volume of the EMTS. In particular, if the robot is tracked, it is important to determine if the distortion that is caused by it can be modeled as a function of the joint configuration and dynamically compensated for.

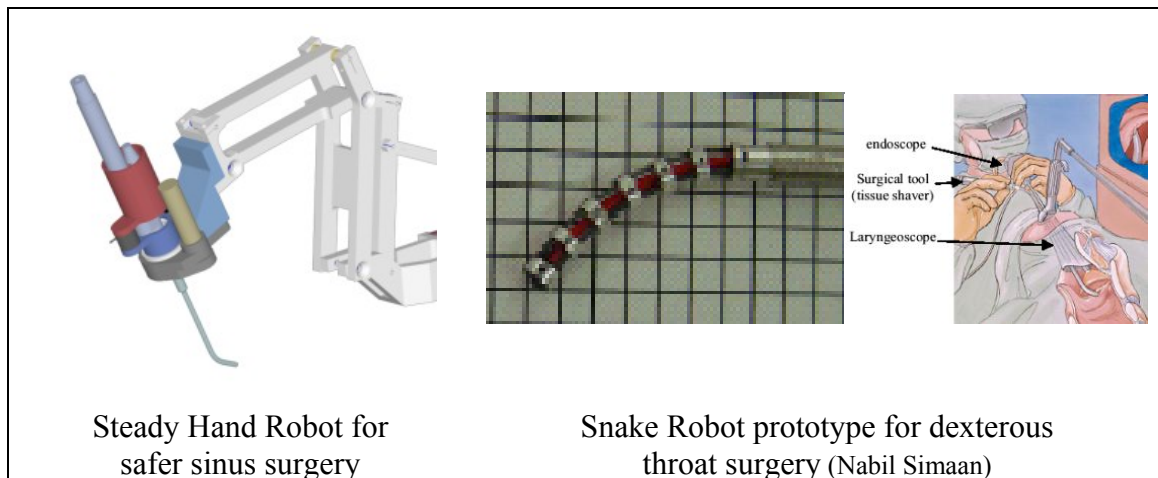


Figure 7-1



## References

Following is a list of useful references that are related to this work. They are organized by type. First are related textbooks. Following are papers that are sorted by category. Finally, there are patents and other references. In each section, the works are arranged in descending chronological order. See the Bibliography for a more traditional list of all references.

### **Books**

[Kutner, 2005] Kutner, M.H., Nachtsheim, C.J., Neter, J. and Li, W. Applied Linear Statistical Models. Boston, MA: McGraw-Hill, 2005.

[Farin, 2002] Farin, Gerald. Curves and Surfaces for CAGD: A Practical Guide. San Francisco, CA: Morgan Kaufmann, 2002.

[Farin, 2000] Farin, Gerald and Hansford, Dianne. The Essentials of CAGD: A Practical Guide. Natick, MA: A K Peters, 2000.

[Murray, 1994] Murray, R.M., Li, Z.X. and Sastry, S.S. A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.

[Bookstein, 1991] Bookstein, Fred L. Morphometric Tools for Landmark Detection: Geometry and Biology. New York, NY. Cambridge University Press, 1994.

[Deif, 1986] Deif, Assem. Sensitivity Analysis in Linear Systems. Berlin, Germany: Springer-Verlag, 1986.

[Deif, 1982] Deif, Assem. Advanced Matrix Theory for Scientists and Engineers. Kent, England: Abacus Press, 1982.

[Lawson, 1974] Lawson, Charles and Hanson, Richard. Solving Least Squares Problems. Englewood Cliffs, NJ: Prentice Hall, 1974.

## ***Tool Design, Registration, and Uncertainty Related Papers***

[Dorst, 2005] Leo Dorst. "First Order Error Propagation of the Procrustes Method of 3D Attitude Estimation." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 27. No. 2. February 2005: 221-229.

[Knott, 2004] P.D. Knott, C.R. Maurer, R. Gallivan and HJ Roh. "The Impact of Fiducial Distribution on Headset-based Registration in Image-guided Sinus Surgery." Otolaryngology-Head and Neck Surgery. Vol. 131. 2004: 666-672.

[West, 2004] Jay B. West and Calvin R. Maurer, Jr. "Designing Optically Tracked Instruments for Image-Guided Surgery." IEEE Transactions on Medical Imaging, Vol. 23. No. 5. May 2004: 533-545.

[Fitzpatrick , 2001] J. Michael Fitzpatrick and Jay B. West. "The Distribution of Target Registration Error in Rigid-Body Point-Based Registration." IEEE Transactions on Medical Imaging. Vol. 20. No. 9. September 2001: 917-927.

[Batchelor , 2000] P. Batchelor and J. M. Fitzpatrick. "A Study of the Anisotropically Weighted Procrustes Problem." IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2000. June 2000. 212 – 218.

[Fitzpatrick , 1998] J. Michael Fitzpatrick, Jay B. West, and Calvin R. Maurer, Jr. "Predicting Error in Rigid-Body Point-Based Registration." IEEE Transactions on Medical Imaging. Vol. 17. No. 6. October 1998: 694-702.

[Pennec , 1995] X. Pennec and J.P. Thirion. "Validation of 3-D Registration Methods based on Points and Frames." INRIA Technical Report. No. 2470. January 1995.

[Curtis , 1993] W.D. Curtis, A.L. Janin, and K. Zikan. "A Note on Averaging Rotations." Virtual Reality Annual International Symposium. 1993: 377-385.

[Umeyama , 1991] Shinji Umeyama. "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 13. No. 4. April 1991: 376-380.

[Goodall , 1991] Colin Goodall. "Procrustes Methods in the Statistical Analysis of Shape." Journal of the Royal Statistical Society. Series B. Vol. 53. No. 2. 1991: 285-339.

[Smith , 1990] R. Smith, M. Self, and P. Cheeseman. "Estimating Uncertain Spatial Relationships in Robotics." Autonomous Robot Vehicles. Springer-Verlag, NY. 1990:167-193.

[Arun, 1987] K.S. Arun, T.S. Huang, and S.D. Blostein. "Least-Squares Fitting of Two 3-D Point Sets." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 9. No. 5. September 1987: 698-700.

[Smith, 1987] R. Smith and P. Cheeseman. "On the Representation and Estimation of Spatial Uncertainty." International Journal of Robotics Research. Vol. 4. No. 4. 1987:56-68.

[Clangron, 1985] S.P. Clangron and A.J. Collins. "Perturbation Theory for Generalized Procrustes Analysis." Journal of the Royal Statistical Society. Series B. Vol. 47. No. 2. 1985: 277-284.

[Sibson , 1979] Robin Sibson. "Studies in the Robustness of Multidimensional Scaling: Perturbational Analysis of Classical Scaling." Journal of the Royal Statistical Society. Series B. Vol. 41. No. 2. 1979: 217-229.

[Sibson , 1978] Robin Sibson. "Studies in the Robustness of Multidimensional Scaling: Procrustes Statistics." Journal of the Royal Statistical Society. Series B. Vol. 40. No. 2. 1978: 234-238.

[Schonemann, 1970] P.H. Schonemann and R.M. Carroll. "Fitting One Matrix to Another Under Choice of a Central Dilation and a Rigid Motion." Psychometrika. Vol. 35. No. 2. June 1970: 245-255.

[Schonemann, 1966] Peter H. Schonemann. "A Generalized Solution to the Orthogonal Procrustes Problem." Psychometrika. Vol. 31. No. 1. March 1966: 1-10.

## ***EM Tracking Related Papers***

[Chung , 2004] A.J. Chung, P.J. Edwards, F. Deligianni, and G. Yang. “Freehand Cocalibration of Optical and Electromagnetic Trackers for Navigated Bronchoscopy.” MIAR 2004 – 2<sup>nd</sup> International Workshop: Medical Imaging and Augmented Reality. Beijing, China, August 2004, Springer LNCS 3150: 320-328.

[Frantz, 2004] D. D. Frantz, S. R. Kirsch, and A.D. Wiles. “Specifying 3D Tracking Accuracy: One Manufacturer’s Views.” BVM 2004.

[Wu, 2003a] Xiaohui Wu and Russell Taylor. “A Direction Space Interpolation Technique for Calibration of Electromagnetic Surgical Navigation Systems.” MICCAI 2003 – Medical Image Computing and Computer-Assisted Intervention. November 2003. Springer LNCS 2879: 215-222.

[Kanada, 2003] K. Kanada, et. al. “A Rapid Method for Magnetic Tracker Calibration Using a Magneto-Optical Hybrid Tracker.” MICCAI 2003 – Medical Image Computing and Computer-Assisted Intervention. November 2003. Springer LNCS 2879: 285-293.

[Frantz, 2003] D. D. Frantz, A.D. Wiles, S.E. Leis, and S. R. Kirsch. “Accuracy Assessment Protocols for Electromagnetic Tracking Systems.” Physics in Medicine and Biology. Vol. 48, July 2003: 2241-2251.

[Wu, 2003b] Xiaohui Wu and Russell Taylor. “A Framework for Calibration of Electromagnetic Surgical Navigation Systems.” IROS 2003. Las Vegas. Vol. 1: 547-552.

[Hummel , 2002] J. Hummel, M. Figl, C. Kollmann, H. Bergmann, and W. Birkfellner. “Evaluation of a Miniature Electromagnetic Position Tracker.” Medical Physics. October 2002. Vol. 29, No. 10: 2205-2212.

[Glossop, 2002] N.D. Glossop, K. Cleary, and F. Banovac. “Needle Tracking Using the Aurora Magnetic Position System.” CAOS 2002. June 2002.

[Wagner, 2002] A. Wagner, K. Schicho, W. Birkfellner et. al. “Quantitative Analysis of Factors Affecting Intraoperative Precision and Stability of Optoelectronic and Electromagnetic Tracking Systems.” Medical Physics. May 2002. Vol. 29, No. 5: 905-912.

[Poulin, 2002] Francois Poulin and L.P. Amiot. “Interference During the Use of an Electromagnetic Tracking System Under OR Conditions.” Journal of Biomechanics. Vol. 35. No. 6. 2002: 733-737.

[Ikits, 2001] M. Ikits, J.D. Brenderson, C.D. Hanse, J.M. Hollerbach. “An Improved Framework for Electromagnetic Tracking Devices.” VR 2001 – IEEE Virtual Reality. March 2001: 63-70.

[Tian, 2001] X. Tian, K.V. Ling, and W.S. Ng. "Magnetic Tracker Calibration for an Augmented Reality System for Therapy." *Proc. Inst of Mech. Eng.* Vol. 215. No. 1. 2001: 51-59.

[Cartellier, 2001] M. Cartellier, F. Vorbeck, and J. Kremser. "Comparison of Six Three-dimensional Navigation Systems During Sinus Surgery." *Acta Otolaryngology.* Vol. 121. 2001: 500-504.

[Kindratenko, 2000a] V. Kindratenko and A. Bennet." Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems. EGVE 2000 – Eurographics Workshop on Virtual Environments. June 2000.

[Kindratenko, 2000b] V. Kindratenko. "A Survey of Electromagnetic Position Tracker Calibration Techniques." *Virtual Reality: Research, Development, and Applications.* Vol. 5. No. 3. 2000: 169-182.

[Birkfellner, 1998a] W. Birkfellner, et. al. "Systematic Distortions in Magnetic Position Digitizers." *Medical Physics.* Vol. 25. No. 11. November 1998: 2242-2248.

[Birkfellner, 1998b] W. Birkfellner, et. al. "Calibration of Tracking Systems in a Surgical Environment." *IEEE Transactions on Medical Imaging.* Vol. 17. No. 5. October 1998: 737-742.

[Zachman, 1997] Gabriel Zachman. "Distortion Correction of Magnetic Fields for Position Tracking." *CGI 1997 – Proc. Computer Graphics International.* June 1997.

[Livingston, 1997] M.A. Livingston and A. State. "Magnetic Tracker Calibration for Improved Augmented Reality Registration." *PRESENCE: Teleoperators and Virtual Environments.* Vol. 6. No. 5. 1997.

[Leotta, 1995] D.F. Leotta, et. al. "Three-Dimensional Ultrasound Imaging Using Multiple Magnetic Tracking Systems and Miniature Magnetic Sensors." *IEEE Ultrasonics Symposium.* 1995: 1415-1418

[Bryson, 1992] Steve Bryson. "Measurement and Calibration of Static Distortions of Position Data From 3D Trackers." *RNR Technical Report.* No. RNR-92-011. March 1992

## **Patents**

[Ben-Haim, 2004] Schlomo Ben-Haim, Daniel Osadchy, Udi Peless, and Ilan Greenberg (Biosense). “System for Determining the Location and Orientation of an Invasive Medical Instrument.” U.S. Patent No. 6,690,963 B2. February 10, 2004.

[Kirsch, 2003a] Stefan R. Kirsch, Hans R. Child, and Christian J. Schilling (Northern Digital). “Gain Factor and Position Determination System.” U.S. Patent No. 6,625,563 B2. September 23, 2003.

[Kirsch, 2003b] Stefan R. Kirsch and Christian J. Schilling (Northern Digital). “Errors in Systems Using Magnetic Fields to Locate Objects.” U.S. Patent No. 6,553,326 B1. April 22, 2003.

[Govari, 2002] Assaf Govari (Biosense). “Electromagnetic Position Single Axis System.” U.S. Patent No. 6,484,118 B1. November 19, 2002.

[Boksberger, 2002] Hans Boksberger, Urs Greuter, Stefan Kirsch, Paul Seiler, and Christian Schilling (Mednetix / Northern Digital). “Device and Process for determining Position.” U.S. Patent No. 6,385,482 B1. May 7, 2002.

[Frantz, 2001] Donald Frantz, Stephen Leis, Stefan Kirsch and Christian Schilling (Northern Digital). “System for Determining Position and/or Orientation of One or More Objects.” U.S. Patent No. 6,228,785 B1. September 11, 2001.

## **Other References**

[Weisstein, 2005] Eric W. Weisstein. “MathWorld--A Wolfram Web Resource.” <http://mathworld.wolfram.com>.

[Boztec, 2005] Andrew Boztec. “Uncertainty Analysis.” PhD Thesis. Ch. 5. 2005.

[Murry, 2004] Herschell Murry (Polhemus). “AC & Pulsed DC Motion Trackers.” <http://www.polhemus.com/ACDCpaper.htm>. 2004.

[Taylor, 2003] Russell Taylor. “Interpolation and Deformations: A Short Cookbook.” Class notes from CS 600.445. October 2003.

[Wu, 2003c] Xiaohui Wu. “JHU-NDI Aurora Project”. Johns Hopkins. May 2003.

[Netter, 2002] Frank Netter and John Hansen. Atlas of Human Anatomy. 3<sup>rd</sup> Edition. 2002.

# **Appendix**

## **Appendix A:**

**Aurora Tool Software User Guide**

## **Appendix B:**

**Related Software, Documents, and Data**  
(Included on attached media)

## Aurora Tools Software – User Guide

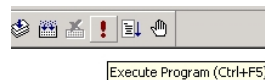
This program provides utilities for use with the Northern Digital Aurora Electromagnetic Tracking System (EMTS).

It allows for:

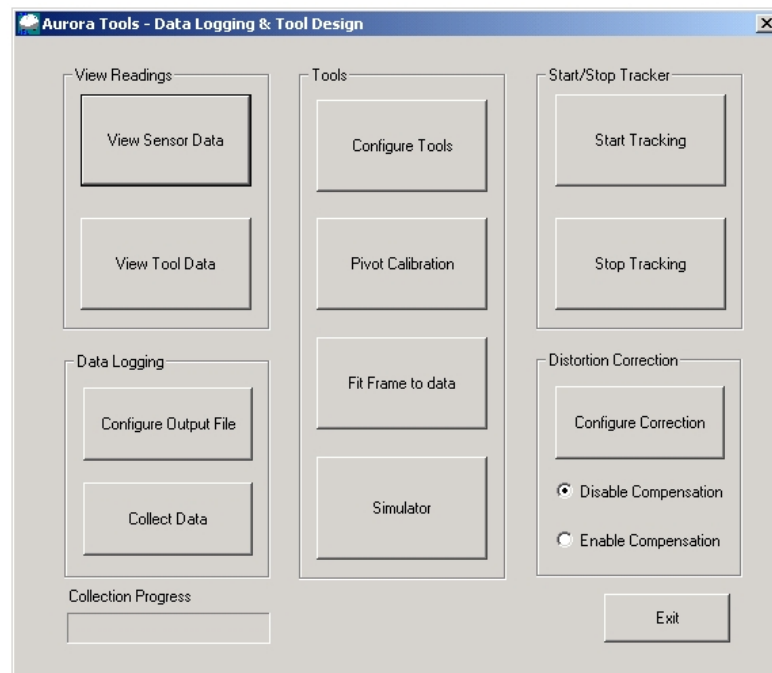
- Viewing sensor and tool readings
- Fitting arbitrary tools comprising of 1-8 sensors to the measurements
- Logging of sensor and tool readings
- Compensation of the sensors, and tools, using a polynomial representing the field distortion of the environment
- Simulation of the distortion effects on an arbitrary tool in a virtual environment specified.

### Initialization

Start the program either from the executable (*Aurora\_Tools.exe*) or by opening the Visual Studio 6.0 Workspace (*Aurora\_Tools*) and executing the software through the menu or with Ctrl-F5.



The initial screen looks like this:



Note: Read the following manual and prepare the configuration files before running the software.



## Real Time Tracking

For Real-time tracking of tools, Click the *Start Tracking* button. This initializes the enable and starts tracking the sensors.

NOTE: Initialization of the Aurora uses the *cisTrackerAurora* functions. There may be a delay before initialization is complete. The PC will beep when the Aurora is ready.

To disable tracking, click *Stop Tracking*. To resume click *Start Tracking* again. There will be no significant delay, the Aurora is NOT reinitialized.

## Viewing Sensor Measurements

To view the sensor readings in real-time, click the *View Sensor Data* button. All valid sensors will show up with the readings from the Aurora, if a tool is not attached, or if it does not produce a valid reading, zero will be displayed. To hold the readings constant in this display, click the *Freeze Readings* button; this will not stop tracking or interrupt data logging.

This screen will look like this:

	X	Y	Z	W0	WX	WY	WZ
Tool Port 1							
Sensor 1	0	0	0	0	0	0	0
Sensor 2	0	0	0	0	0	0	0
Tool Port 2							
Sensor 3	0	0	0	0	0	0	0
Sensor 4	0	0	0	0	0	0	0
Tool Port 3							
Sensor 5	0	0	0	0	0	0	0
Sensor 6	0	0	0	0	0	0	0
Tool Port 4							
Sensor 7	0	0	0	0	0	0	0
Sensor 8	0	0	0	0	0	0	0

Frame Number

## Viewing Tool Measurements

To view the tool readings in real-time, click the *View Tool Data* button. All active tools will show up. If a tool is not selected, or if it does not produce a valid reading, zero will be displayed. To hold the readings constant in this display, click the *Freeze Readings* button; this will not stop tracking or interrupt data logging.

The screen will look like this:

The screenshot shows a window titled "Tool Data" with a close button in the top right corner. Inside the window, there is a table of tool readings. The columns are labeled X, Y, Z, W0, WX, WY, WZ, and Error. The rows are labeled Tool 1, Tool 2, Tool 3, and Tool 4. Each cell in the table contains the number 0. Below the table, there is a "Frame Number" field with the value 0, a "Freeze Readings" button, and a "Close" button.

	X	Y	Z	W0	WX	WY	WZ	Error
Tool 1	0	0	0	0	0	0	0	0
Tool 2	0	0	0	0	0	0	0	0
Tool 3	0	0	0	0	0	0	0	0
Tool 4	0	0	0	0	0	0	0	0

Frame Number: 0

Freeze Readings

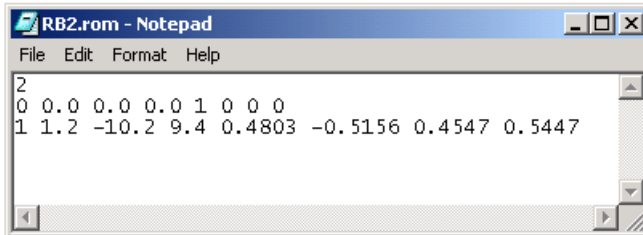
Close

Note: Be sure the tools to be viewed are enabled. They will all be enabled by default.

## Configuring Tools

To configure the tools, you must set up a configuration file for each tool. If two tools have identical configurations, the same one may be used twice.

An example of a tool file for a Traxtal 6 DoF Dynamic Reference is shown below:



```

2
0 0.0 0.0 0.0 1 0 0 0
1 1.2 -10.2 9.4 0.4803 -0.5156 0.4547 0.5447
  
```



The file format is as follows:

Line 1: The number of sensors used.

Line 2+: The sensor configurations with respect to that tool's frame (described below).

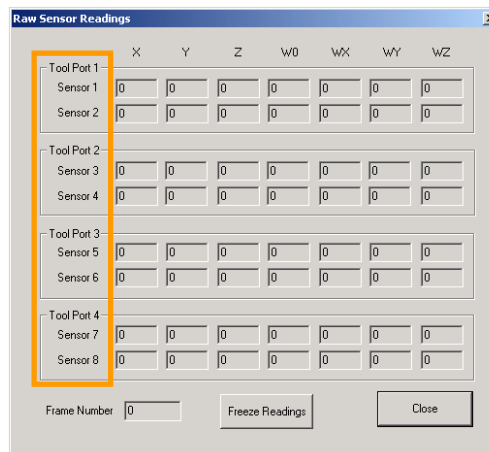
For Lines 2+:

Column 1: The sensor number, the numbers do not need to be consecutive or on the same tool port. The figure below shows the sensor ordering.

NOTE: The numbering in the file starts at 0, not 1

Column 2-4: The position of the sensor (x,y,z) wrt the tool frame (measured in mm).

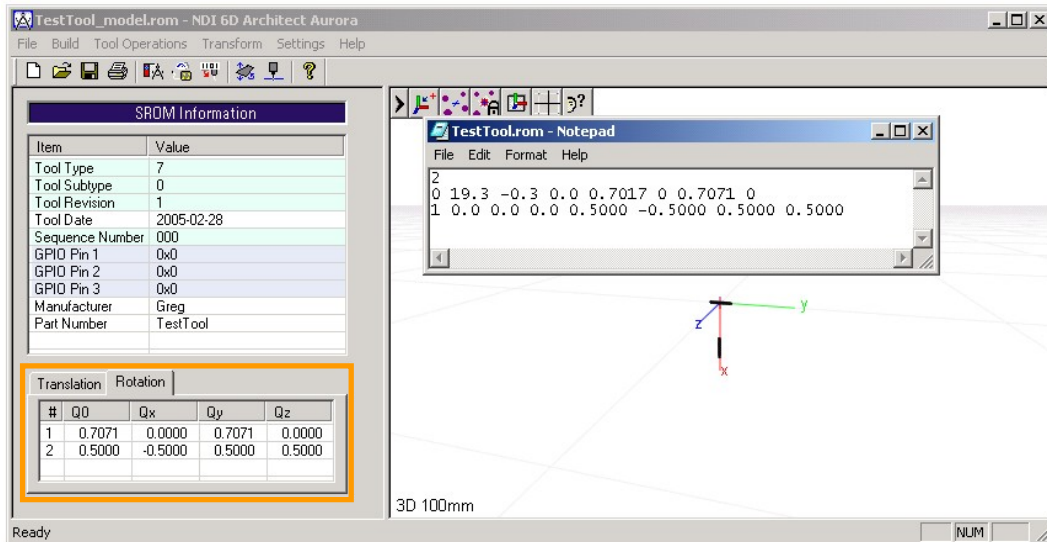
Column 5-7: The orientation of the sensor with respect to the tool frame, represented as a unit quaternion ( $q_0, q_x, q_y, q_z$ ).



## Generating the Tool Configuration

If the tool configuration is known precisely, either through manufacturing specifications or because it is a virtual tool, the data can be directly input into the tool configuration file.

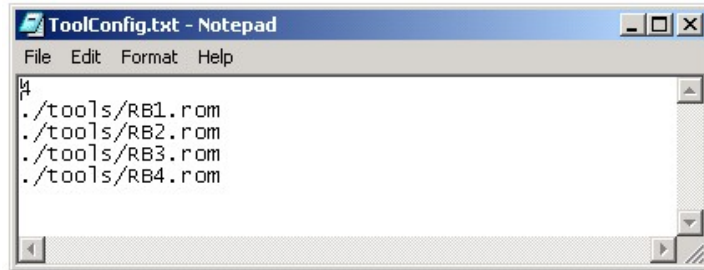
For existing tools, it is possible to generate the sensor positions and orientations by opening the supplied binary .rom file (from NDI, Traxtal, etc.) using the *NDI 6D Architect Aurora* software. From here, you will get a view like the one below where the *Translation* and *Rotation* can be copied into the tool configuration file as shown.



For manual design of tool configurations (either actual or virtual), it may also be useful to use the *6D Architect* software. In this case, the tool can be designed as one normally would be, and the *Translation* and *Rotation* information copied out when complete.

## Combining the Tool Configurations

Once all of the *.rom* files are complete, compile a list of them in *ToolConfig.txt* as shown here. Be sure to include the path, the tool configuration files are in the */tools* subdirectory.



The file format is as follows:

Line 1: The number of tools.

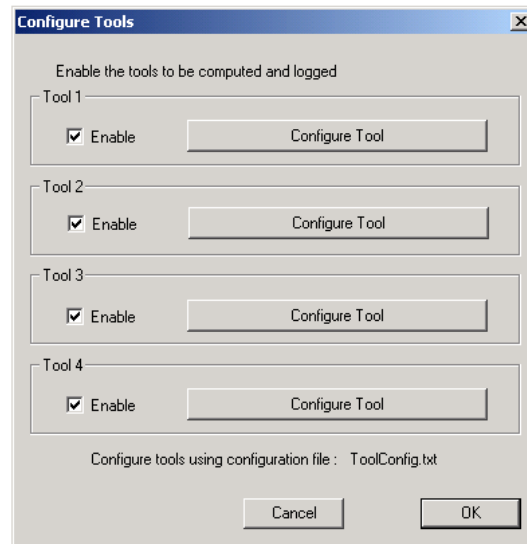
Line 2+: The filename corresponding to the respective tool. Up to four tools are currently supported due to Aurora's limit of 8 sensor measurements.

NOTE: It is possible to use the same sensor in multiple tools. This can be useful for the simulator when trying to optimize the number and relative position of sensors in a RB.

## Enabling Tools

When running the software, the user must select which tools are to be tracked, and logged if data logging is active. By default, all four tools are tracked.

Select which tools are active by clicking on the *Configure Tools* button. This will bring up a window like this:



Use the checkboxes to select which tools are active.

Note: Future revisions will allow for modifying the tool configuration inside of the software using the *Configure Tool* buttons, but they are currently inactive.

## Data Logging

To enable data logging, click the *Collect Data* button. Be sure to set the configuration manually if the default settings shown below are not appropriate.

Data logging is configured by clicking on *Configure Output File*. This will bring up a window that looks like this:

The screenshot shows a 'Data Logging Configuration' dialog box. It is divided into several sections. The first section, 'Sampling Parameters', contains three input fields: 'Sample Rate (hz)' with the value '20', 'Groups of Samples' with the value '10', and 'Samples per Group' with the value '100'. The second section, 'Raw Sensor Readings', has an 'Enable' checkbox checked and a 'Filename' field containing './data/Aurora\_SensorData.txt'. The third section, 'Corrected Sensor Readings', also has an 'Enable' checkbox checked and a 'Filename' field containing './data/Aurora\_CorrectedSensorData.txt'. The fourth section, 'Tool Frame Readings', has an 'Enable' checkbox checked and a 'Filename' field containing './data/Aurora\_FrameData.txt'. Below these sections are 'Cancel' and 'OK' buttons.

The key parameters are:

*Sample Rate* which specifies how fast the Aurora Readings are taken.

*Groups of Samples* which specifies how many sets of data are to be collected.

A window will pop up between sets and will begin collection of that set when the *OK* button is clicked.

*Samples per Group* which specifies the number of measurements used per group.

If only one group is used, this is the total number of points to collect.

Next, choose with the check boxes which of the three data types to log. The types are:

*Raw Sensor Readings* which is the actual sensor measurements

*Corrected Sensor Measurements* which is the sensor measurements after being compensated using the polynomial

*Tool Frame Readings* which is the position and orientation of the tool frames that were fit to the data. If compensation is enabled, these will be the compensated tool readings, otherwise they will be based on the raw sensor measurements.

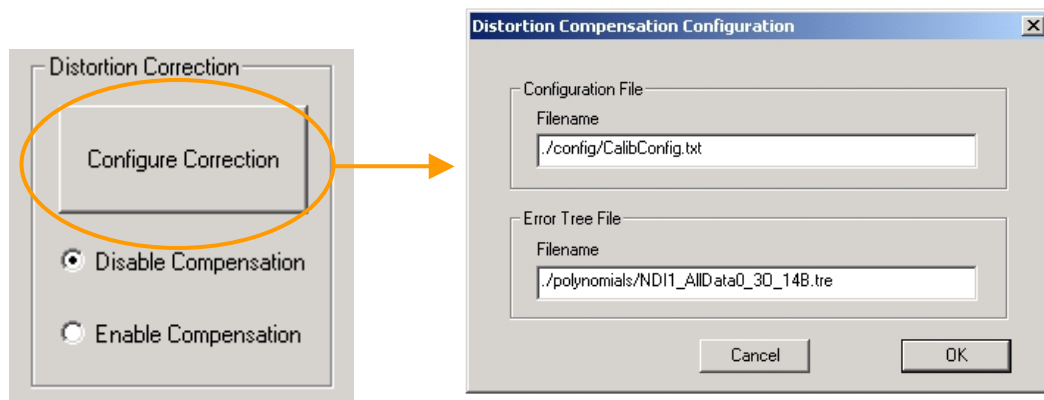
If desired, change the filenames and/or directories where the data is stored.

NOTE: *Sample Rate* affects the rate at which Aurora readings are polled. This parameter affects the readings even if data logging is not active.

## Compensation of Measurements

The software has the capability of compensating for magnetic field distortions by using polynomial models to interpolate what the position and orientation error will be at a given sensor position and orientation. The compensation is based on a polynomial generated from data collected in the environment that the system is working in. This can be either a compensation for magnetic field distortion due to the environment, or a characterization of the systematic distortion of a particular Aurora unit.

To configure the distortion Correction, click on the *Configure Correction* button on the main window as shown on the left. This will bring up the following window on the right.



There are two parameters that must be set:

The first is the *Configuration File*. This file is primarily used in generation of the polynomial, the important line that should be set is the one labeled `Poly Order=3`. Change the number three to the order corresponding to the polynomial being used. Note: this file is in the */config* subdirectory.

The second is the *Error Tree File*. This points to the file that contains all of the information about the specific polynomial to use. Note: this file is in the */polynomials* subdirectory.

Both parameters have default values set, but they should be changed to match the appropriate environment.

NOTE: This not only affects the real-time compensation, but also the Simulator.

### Enabling Compensation

To enable compensation, select the radio button for *Enable Compensation* as shown in the figure above to the left. To disable compensation, click on *Disable Compensation*.



## Simulator

The simulator is used to simulate the distortion of an arbitrary tool configuration and an arbitrary reference configuration as the tool is moved through the distortion field.

Access the simulator by clicking on *Simulator* on the main window. This will bring up the following window:

The screenshot shows the 'Aurora Tool Simulation' dialog box with the following sections and values:

- Set Reference Frame:** Tool # 0, X -50, Y 0, Z -300, W0 1, WX 0, WY 0, WZ 0. Note: Set Reference Tool Number to zero for tool simulation independent of reference. Reference frame will be assumed to be origin of Aurora.
- Tool Frame wrt Reference:** Tool # 1, X 0, Y 0, Z 0, W0 1, WX 0, WY 0, WZ 0.
- Simulated Tool Frame wrt Reference:** X -34.817, Y -134.98, Z -324.54, W0 0.9999, WX -0.0003, WY -0.0010, WZ 0.0001.
- Error:** X 0.1820, Y 0.0164, Z 0.4502, Positional (mm) 0.48591, Angular(Deg) 0.13100.
- Data Logging:** Data File ./data/SimulatorData.txt,  Enable Data Logging.
- Tool Range of Motion:** Max X 50, Y 50, Z 50, Rx 90, Ry 90, Rz 90; Min X -50, Y -50, Z -50, Rx -90, Ry -90, Rz -90; Sample Spacing 50 mm, 45 Deg.
- Range of Motion Results:** Positional (mm) RMS 0.48591, Max 0.50633, Mean 0.49612; Angular(Deg) 0.13100, Max 0.13975, Mean 0.13537.  Visualize Individual Data Points.
- Input from File:** Simulate Input Data button, Reminder: Set tool and reference numbers at top left and enable logging, Data File ./data/InputSimulatorData.txt.
- Buttons:** Simulate Single Point, Close.

### Directions for use in Single Point mode:

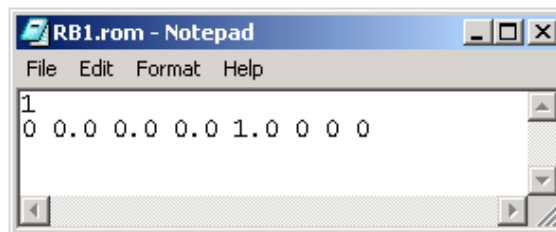
1. Set the Reference Frame
  - a. Choose which tool number represents the reference  
Use 0 for Tool # if measurement are to be made with respect to the Aurora Field Generator's frame of reference
  - b. Choose a position and orientation in the field for the reference
2. Set the Tool Frame
  - a. Choose which tool number represents the tool
  - b. Choose a position and orientation for the tool, this is with respect to the reference frame.
3. Click *Simulate Single Point*
4. Results in "real time"
  - a. The position and orientation of the tool wrt the reference after distortion will be shown in the box labeled *Simulated Tool Frame wrt Reference*
  - b. The error in relative position and orientation will be shown in the box labeled *Error*

**Directions for use in Range of Motion mode:**

1. Set the Reference Frame
  - a. Choose which tool number represents the reference  
Use 0 for Tool # if measurement are to be made with respect to the Aurora Field Generator's frame of reference
  - b. Choose a position and orientation in the field for the reference
2. Set the Range of Motion
  - a. Choose the extents of the motion of the tool wrt the reference  
Position measurements are in mm, Orientation measurements are in degrees
  - b. Choose the spacing of the samples between the extents  
Be aware of the cubic relationship of each spacing, too many readings may cause the simulation to take excessively long
3. Configure data logging
  - a. Choose a filename, may keep the default if desired
  - b. Click the checkbox to enable logging
4. Click *Simulate Range of Motion*
5. Results in "real time" (optional, not recommended)
  - a. Choose *Visualize Individual Data Points* to show each readings as it is calculated, this will significantly slow down the simulation.
  - b. The position and orientation of the tool wrt the reference after distortion will be shown in the box labeled *Simulated Tool Frame wrt Reference*
  - c. The error in position and orientation is shown in the box labeled *Error*
6. Results
  - a. Simple statistics representing the results of the simulation are shown in the box *Range of Motion Results*.
  - b. The full data set of the relative positional and angular error introduced for each position and orientation of the tool with respect to the reference is stored in the data file (if that option was selected).

NOTE: To simulate the distortion of a single sensor, make a simple tool file (.rom) with just one sensor and set it such that the position is zero and the orientation is identity.

It should look like this to simulate a single sensor in the field:



## Directions for use in File Input Data mode:

### 1. Generate the Input Data File

The file format is as follows:

Line 1: The number of measurement pairs to simulate.

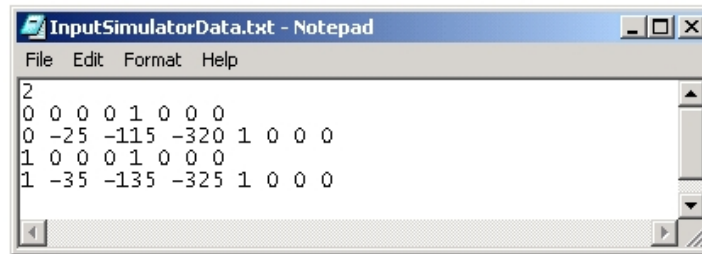
Line 2+: The reference and tool frames (described below).

Column 1: The number of the sensor pair (only for reference)

Column 2-4: The position of the frame (x,y,z) in mm.

Column 5-7: The orientation of the sensor as a unit quaternion ( $q_0, q_x, q_y, q_z$ ).

Inputs are grouped into two lines. The first line represents the frame of the reference in the base frame. The second line represents the tool frame **with respect to the reference**. The following figure shows a sample file.



### 2. Set the Reference & Tool Configurations

#### a. Choose which tool number represents the reference

Use 0 for Tool # if measurement are to be made with respect to the Aurora Field Generator's frame of reference, still must have "dummy lines" in input data file for reference locations

#### b. Choose which tool number represents the tool

### 3. Set the Input Filename

#### a. Input the name of the data file containing the reference and tool frames

The tool frame is treated as the frame relative to the reference.

It is **not** an absolute reference

### 4. Click *Simulate Input Data*

### 5. Results in "real time" (optional, not recommended)

#### a. Choose *Visualize Individual Data Points* to show each readings as it is calculated, this will slow down the simulation significantly.

#### b. The position and orientation of the tool wrt the reference after distortion will be shown in the box labeled *Simulated Tool Frame wrt Reference*

#### c. The error in position and orientation is shown in the box labeled *Error*

### 7. Results

#### a. Simple statistics representing the results of the simulation are shown in the box *Range of Motion Results*.

#### b. The full data set of the relative positional and angular error introduced for each position and orientation of the tool with respect to the reference is stored in the data file (**if that option was selected**).

## Bibliography

- [Arun, 1987] K.S. Arun, T.S. Huang, and S.D. Blostein. "Least-Squares Fitting of Two 3-D Point Sets." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 9. No. 5. September 1987: 698-700.
- [Batchelor , 2000] P. Batchelor and J. M. Fitzpatrick. "A Study of the Anisotropically Weighted Procrustes Problem." IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2000. June 2000: 212- 218.
- [Ben-Haim, 2004] Schlomo Ben-Haim, Daniel Osadchy, Udi Peless, and Ilan Greenberg (Biosense). "System for Determining the Location and Orientation of an Invasive Medical Instrument." U.S. Patent No. 6,690,963 B2. February 10, 2004.
- [Birkfellner, 1998a] W. Birkfellner, et. al. "Systematic Distortions in Magnetic Position Digitizers." Medical Physics. Vol. 25. No. 11. November 1998: 2242-2248.
- [Birkfellner, 1998b] W. Birkfellner, et. al. "Calibration of Tracking Systems in a Surgical Environment." IEEE Transactions on Medical Imaging. Vol. 17. No. 5. October 1998: 737-742.
- [Boctor, 2004] E. Boctor , G. Fischer, M. Choti, G. Fichtinger, R. Taylor. "A Dual-Armed Robotic System for Intraoperative Ultrasound Guided Hepatic Ablative Therapy." International Conference on Robotics and Automation - ICRA 2004. New Orleans, LA. IEEE 2004: 2517-2522.
- [Boksberger, 2002] Hans Boksberger, Urs Greuter, Stefan Kirsch, Paul Seiler, and Christian Schilling (Mednetix / Northern Digital). "Device and Process for determining Position." U.S. Patent No. 6,385,482 B1. May 7, 2002.
- [Bookstein, 1991] Bookstein, Fred L. Morphometric Tools for Landmark Detection: Geometry and Biology. New York, NY. Cambridge University Press, 1994.
- [Boztec, 2005] Andrew Boztec. "Uncertainty Analysis." PhD Thesis. Chapter 5. 2005.
- [Bryson, 1992] Steve Bryson. "Measurement and Calibration of Static Distortions of Position Data From 3D Trackers." RNR Technical Report. No. RNR-92-011. March 1992
- [Cartellier, 2001] M. Cartellier, F. Vorbeck, and J. Kremser. "Comparison of Six Three-dimensional Navigation Systems During Sinus Surgery." Acta Otolaryngology. Vol. 121. 2001: 500-504.
- [Chung , 2004] A.J. Chung, P.J. Edwards, F. Deligianni, and G. Yang. "Freehand Cocalibration of Optical and Electromagnetic Trackers for Navigated Bronchoscopy." MIAR 2004 – 2<sup>nd</sup> International Workshop: Medical Imaging and Augmented Reality. Beijing, China, August 2004, Springer LNCS 3150: 320-328.
- [Clangron, 1985] S.P. Clangron and A.J. Collins. "Perturbation Theory for Generalized Procrustes Analysis." Journal of the Royal Statistical Society. Series B. Vol. 47. No. 2. 1985: 277-284.
- [Curtis , 1993] W.D. Curtis, A.L. Janin, and K. Zikan. "A Note on Averaging Rotations." Virtual Reality Annual International Symposium. 1993: 377-385.
- [Deif, 1982] Deif, Assem. Advanced Matrix Theory for Scientists and Engineers. Kent, England: Abacus Press, 1982.
- [Deif, 1986] Deif, Assem. Sensitivity Analysis in Linear Systems. Berlin, Germany: Springer-Verlag, 1986.

- [Dorst, 2005] Leo Dorst. "First Order Error Propagation of the Procrustes Method of 3D Attitude Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 27. No. 2. February 2005: 221-229.
- [Farin, 2000] Farin, Gerald and Hansford, Dianne. *The Essentials of CAGD: A Practical Guide*. Natick, MA: A K Peters, 2000.
- [Farin, 2002] Farin, Gerald. *Curves and Surfaces for CAGD: A Practical Guide*. San Francisco, CA: Morgan Kaufmann, 2002.
- [Fitzpatrick , 1998] J. Michael Fitzpatrick, Jay B. West, and Calvin R. Maurer, Jr. "Predicting Error in Rigid-Body Point-Based Registration." *IEEE Transactions on Medical Imaging*. Vol. 17. No. 6. October 1998: 694-702.
- [Fitzpatrick , 2001] J. Michael Fitzpatrick and Jay B. West. "The Distribution of Target Registration Error in Rigid-Body Point-Based Registration." *IEEE Transactions on Medical Imaging*. Vol. 20. No. 9. September 2001: 917-927.
- [Frantz, 2001] Donald Frantz, Stephen Leis, Stefan Kirsch and Christian Schilling (Northern Digital). "System for Determining Position and/or Orientation of One or More Objects." U.S. Patent No. 6,228,785 B1. September 11, 2001.
- [Frantz, 2003] D. D. Frantz, A.D. Wiles, S.E. Leis, and S. R. Kirsch. "Accuracy Assessment Protocols for Electromagnetic Tracking Systems." *Physics in Medicine and Biology*. Vol. 48, July 2003: 2241-2251.
- [Frantz, 2004] D. D. Frantz, S. R. Kirsch, and A.D. Wiles. "Specifying 3D Tracking Accuracy: One Manufacturer's Views." *BVM* 2004.
- [Glossop, 2002] N.D. Glossop, K. Cleary, and F. Banovac. "Needle Tracking Using the Aurora Magnetic Position System." *CAOS* 2002. June 2002.
- [Goodall , 1991] Colin Goodall. "Procrustes Methods in the Statistical Analysis of Shape." *Journal of the Royal Statistical Society. Series B*. Vol. 53. No. 2. 1991: 285-339.
- [Govari, 2002] Assaf Govari (Biosense). "Electromagnetic Position Single Axis System." U.S. Patent No. 6,484,118 B1. November 19, 2002.
- [Hummel , 2002] J. Hummel, M. Figl, C. Kollmann, H. Bergmann, and W. Birkfellner. "Evaluation of a Miniature Electromagnetic Position Tracker." *Medical Physics*. October 2002. Vol. 29, No. 10: 2205-2212.
- [Ikits, 2001] M. Ikits, J.D. Brenderson, C.D. Hanse, J.M. Hollerbach. "An Improved Framework for Electromagnetic Tracking Devices." *VR 2001 – IEEE Virtual Reality*. March 2001: 63-70.
- [Kanada, 2003] K. Kanada, et. al. "A Rapid Method for Magnetic Tracker Calibration Using a Magneto-Optical Hybrid Tracker." *MICCAI 2003 – Medical Image Computing and Computer-Assisted Intervention*. November 2003. Springer LNCS 2879: 285-293.
- [Kindratenko, 2000a] V. Kindratenko and A. Bennet. "Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems." *EGVE 2000 – Eurographics Workshop on Virtual Environments*. June 2000.
- [Kindratenko, 2000b] V. Kindratenko. "A Survey of Electromagnetic Position Tracker Calibration Techniques." *Virtual Reality: Research, Development, and Applications*. Vol. 5. No. 3. 2000: 169-182.
- [Kirsch, 2003a] Stefan R. Kirsch, Hans R. Child, and Christian J. Schilling (Northern Digital). "Gain Factor and Position Determination System." U.S. Patent No. 6,625,563 B2. September 23, 2003.

- [Kirsch, 2003b] Stefan R. Kirsch and Christian J. Schilling (Northern Digital). "Errors in Systems Using Magnetic Fields to Locate Objects." U.S. Patent No. 6,553,326 B1. April 22, 2003.
- [Knott, 2004] P.D. Knott, C.R. Maurer, R. Gallivan and HJ Roh. "The Impact of Fiducial Distribution on Headset-based Registration in Image-guided Sinus Surgery." *Otolaryngology-Head and Neck Surgery*. Vol. 131. 2004: 666-672.
- [Kutner, 2005] Kutner, M.H., Nachtsheim, C.J., Neter, J. and Li, W. *Applied Linear Statistical Models*. Boston, MA: McGraw-Hill, 2005.
- [Lawson, 1974] Lawson, Charles and Hanson, Richard. *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice Hall, 1974.
- [Leotta, 1995] D.F. Leotta, et. al. "Three-Dimensional Ultrasound Imaging Using Multiple Magnetic Tracking Systems and Miniature Magnetic Sensors." *IEEE Ultrasonics Symposium*. 1995: 1415-1418
- [Livingston, 1997] M.A. Livingston and A. State. "Magnetic Tracker Calibration for Improved Augmented Reality Registration." *PRESENCE: Teleoperators and Virtual Environments*. Vol. 6. No. 5. 1997.
- [Murray, 1994] Murray, R.M., Li, Z.X. and Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [Murry, 2004] Herschell Murry (Polhemus). "AC & Pulsed DC Motion Trackers." <http://www.polhemus.com/ACDCpaper.htm>. 2004.
- [Netter, 2002] Frank Netter and John Hansen. *Atlas of Human Anatomy*. 3<sup>rd</sup> Edition. 2002.
- [Pennec , 1995] X. Pennec and J.P. Thirion. "Validation of 3-D Registration Methods based on Points and Frames." *INRIA Technical Report*. No. 2470. January 1995.
- [Poulin, 2002] Francois Poulin and L.P. Amiot. "Interference During the Use of an Electromagnetic Tracking System Under OR Conditions." *Journal of Biomechanics*. Vol. 35. No. 6. 2002: 733-737.
- [Schonemann, 1966] Peter H. Schonemann. "A Generalized Solution to the Orthogonal Procrustes Problem." *Psychometrika*. Vol. 31. No. 1. March 1966: 1-10.
- [Schonemann, 1970] P.H. Schonemann and R.M. Carroll. "Fitting One Matrix to Another Under Choice of a Central Dilation and a Rigid Motion." *Psychometrika*. Vol. 35. No. 2. June 1970: 245-255.
- [Sibson , 1978] Robin Sibson. "Studies in the Robustness of Multidimensional Scaling: Procrustes Statistics." *Journal of the Royal Statistical Society. Series B*. Vol. 40. No. 2. 1978: 234-238.
- [Sibson , 1979] Robin Sibson. "Studies in the Robustness of Multidimensional Scaling: Perturbational Analysis of Classical Scaling." *Journal of the Royal Statistical Society. Series B*. Vol. 41. No. 2. 1979: 217-229.
- [Sinha, 2003] T.K. Sinha, V. Duay, B.M. Dawant and M.I. Miga. "Cortical Shift Tracking Using a Laser Range Scanner and Deformable Registration Methods." *MICCAI 2003 – Medical Image Computing and Computer-Assisted Intervention*. November 2003. Springer LNCS 2879: 166-174.
- [Smith , 1990] R. Smith, M. Self, and P. Cheeseman. "Estimating Uncertain Spatial Relationships in Robotics." *Autonomous Robot Vehicles*. Springer-Verlag, NY. 1990:167-193.
- [Smith, 1987] R. Smith and P. Cheeseman. "On the Representation and Estimation of Spatial Uncertainty." *International Journal of Robotics Research*. Vol. 4. No. 4. 1987:56-68.

- [Taylor, 2003] Russell Taylor. "Interpolation and Deformations: A Short Cookbook." Class notes from CS 600.445. October 2003.
- [Tian, 2001] X. Tian, K.V. Ling, and W.S. Ng. "Magnetic Tracker Calibration for an Augmented Reality System for Therapy." Proc. Inst of Mech. Eng. Vol. 215. No. 1. 2001: 51-59.
- [Umeyama , 1991] Shinji Umeyama. "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 13. No. 4. April 1991: 376-380.
- [Wagner, 2002] A. Wagner, K. Schicho, W. Birkfellner et. al. "Quantitative Analysis of Factors Affecting Intraoperative Precision and Stability of Optoelectronic and Electromagnetic Tracking Systems." Medical Physics. May 2002. Vol. 29, No. 5: 905-912.
- [Weisstein, 2005] Eric W. Weisstein. "MathWorld--A Wolfram Web Resource."  
<http://mathworld.wolfram.com>.
- [West, 2004] Jay B. West and Calvin R. Maurer, Jr. "Designing Optically Tracked Instruments for Image-Guided Surgery." IEEE Transactions on Medical Imaging, Vol. 23. No. 5. May 2004: 533-545.
- [Wu, 2003a] Xiaohui Wu and Russell Taylor. "A Direction Space Interpolation Technique for Calibration of Electromagnetic Surgical Navigation Systems." MICCAI 2003 – Medical Image Computing and Computer-Assisted Intervention. November 2003. Springer LNCS 2879: 215-222.
- [Wu, 2003b] Xiaohui Wu and Russell Taylor. "A Framework for Calibration of Electromagnetic Surgical Navigation Systems." IROS 2003. Las Vegas. Vol. 1: 547-552.
- [Wu, 2003c] Xiaohui Wu. "JHU-NDI Aurora Project". Johns Hopkins University Project Report. May 2003.
- [Zachman, 1997] Gabriel Zachman. "Distortion Correction of Magnetic Fields for Position Tracking." CGI 1997 – Proc. Computer Graphics International. June 1997.

## Curriculum Vita

Gregory Scott Fischer was born on July 16, 1980 in New Brunswick, New Jersey. He resided in Warren, New Jersey and went to high school at Watchung Hills Regional High School (WHRHS) where he graduated in 1998 and graduated with the award in “Science and Technology.” He then went on to receive Bachelor of Science degrees in both Mechanical Engineering and Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY in 2002. Greg Fischer began graduate studies at The Johns Hopkins University in Baltimore, Maryland during the fall of 2002. Since 2002, he has been a graduate research assistant at the NSF Engineering Research Center for Center for Computer Integrated Surgical Systems and Technology. He received the Master of Science in Engineering(MSE) degree in Electrical Engineering from The Johns Hopkins University, Baltimore, MD in 2004. Greg is currently working towards the PhD degree in Mechanical Engineering at The Johns Hopkins University.



## Publications

1. Fischer G, Zand J, Talamini M, Marohn M, Akinbiyi T, Kanev K, Kuo J, Kazanzides P, Taylor R, Intra-operative Ischemia Sensing Surgical Instruments, International Conference on Complex Medical Engineering - CME 2005, Takamatsu, Japan, 2005
2. Zand J, Fischer G, Hanly E, Shih S, Marohn M, Taylor R, Talamini M, "Ischemia Sensing Surgical Instruments", Annual congress of the Society of American Gastrointestinal Endoscopic Surgeons - SAGES 2005, Ft. Lauderdale, FL, 2005
3. DeOliviera M, Deguet A, Fischer G, Balogh E, Taylor RH, Fayad LM, Zinreich SJ, Fichtinger G, "Image Overlay for CT-Guided Hepatic Needle Insertions -- Cadaver Studies", Annual congress of the Society of American Gastrointestinal Endoscopic Surgeons - SAGES 2005, Ft. Lauderdale, FL, 2005
4. Fichtinger G, Deguet A, Masamune K, Fischer G, Balogh E, Mathieu H, Taylor RH, SJ Zinreich, Fayad LM, Image Overlay Guidance for Needle Insertion in CT Scanner, IEEE Transactions on Biomedical Engineering
5. Fichtinger G, Deguet A, Fischer G, Balogh E, Masamune K, Taylor RH, Fayad LM, SJ Zinreich, CT Image Overlay for Percutaneous Needle Insertions, Journal of Computer Assisted Surgery - JCAS
6. Fichtinger G, Deguet A, Masamune K, Fischer G, Balogh E, Mathieu H, Taylor RH, Fayad LM, SJ Zinreich, Needle Insertion in CT Scanner with Image Overlay -Cadaver Studies, Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2004, St Malo, France, Lecture Notes on Computer Science, Vol 3217, pp 795-783, Springer, 2004
7. Boctor E, Fischer G, Choti M, Fichtinger G, Taylor R, A Dual-Armed Robotic System for Intraoperative Ultrasound Guided Hepatic Ablative Therapy, International Conference on Robotics and Automation - ICRA 2004, New Orleans, LA, pp 2517-2522, IEEE, 2004
8. Prasad S, Kitagawa M, Fischer G, Zand J, Talamini M, Taylor R, Okamura A, A Modular 2-DOF Force-Sensing Instrument For Laparoscopic Surgery, Sixth International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003, Montreal, Canada, Lecture Notes on Computer Science, Vol 2878, pp 279-286, Springer, 2003