



**Design of a Robotic Transcranial Magnetic
Stimulation System**

by

XIANG YI

April, 2012

Submitted to School of Mechanical & System Engineering in fulfilment of the
requirements for Degree of Doctor of Philosophy

Newcastle University

United Kingdom

ABSTRACT

Transcranial Magnetic Stimulation (TMS) is an excellent and non-invasive technique for studying the human brain. Accurate placement of the magnetic coil is required by this technique in order to induce a specific cortical activity. Currently, the coil is manually held in most of stimulation procedures, which does not achieve the precise clinical evaluation of the procedure. This thesis proposes a robotic TMS system to resolve these problems as a robot has excellent locating and holding capabilities. The proposed system can track in real-time the subject's head position and simultaneously maintain a constant contact force between the coil and the subject's head so that it does not need to be restrained and thus ensure the accuracy of the stimulation result.

Requirements for the robotic TMS system are proposed initially base on analysis of a serial of TMS experiments on real subjects. Both hardware and software design are addressed according to these requirements in this thesis. An optical tracking system is used in the system for guiding and tracking the motion of the robot and inadvertent small movements of the subject's head. Two methods of coordinate system registration are developed base on DH and Tsai-lenz's method, and it is found that DH method has an improved accuracy (RMS error is 0.55mm). In addition, the contact force is controlled using a Force/Torque sensor; and a combined position and force tracking controller is applied in the system. This combined controller incorporates the position tracking and conventional gain scheduling force control algorithms to monitor both position and force in real-time. These algorithms are verified through a series of experiments. And it is found that the maximum position and force error are 3mm and 5N respectively when the subject moves at a speed of 20mm/s. Although the performance still needs to be improved to achieve a better system, the robotic system has shown the significant advantage compared with the manual TMS system.

Keywords—Transcranial Magnetic Stimulation, Robot arm, Medical system, Calibration, Tracking

ACKNOWLEDGEMENTS

I would like to sincerely acknowledge my supervisor, Dr. Robert Bicker, for his support, encouragement and guidance. I have not only learned from him the technique aspects related to my work but also how to manage my research plan. I also wish to appreciate my second supervisor, Prof. Stuart Baker, for his support, guidance and constructive comments on my research.

I also sincerely thank all my other colleagues at Newcastle University, especially all members of the robotics lab, for their moral support.

This thesis is dedicated to my family.

I would like to express my deep gratitude to my parents, Chang Liu, for their endless support, love during my study.

Table of Contents

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
GLOSSARY OF TERMS	xxi
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Aims and Objectives	5
1.3 Hypothesis	5
1.4 Outline of thesis	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Principles and devices of TMS	7
2.2 Diagnostic application of different patterns of TMS	10
2.2.1 Single-pulse TMS	10
2.2.2 Paired-pulse TMS	13
2.2.3 Repetitive TMS (rTMS)	14
2.3 Therapeutic applications	15
2.4 Localization of coil placement	17
2.4.1 Non-stereotactic navigated localization strategy	17
2.4.2 Stereotactic image-guided localization strategy	20
2.4.3 The pose of the coil	22
2.5 Robot assisted surgical system	23
2.6 Existing robot assisted TMS systems	30
2.7 Relevant technologies	34
2.7.1 Tracking technologies	34
2.7.2 Visual servo control	37
2.7.3 Coordinate systems registration	39

2.7.4 Trajectory planning	46
2.7.5 Position and force control of robot manipulators.....	48
2.8 Safety issues	51
2.8.1 Safety of TMS	51
2.8.2 Safety of robot application	51
2.9 Summary	54
CHAPTER 3 PRELIMINARY TMS EXPERIMENTS AND DATA COLLECTION	55
3.1 The TMS system	55
3.1.1 Magstim 200 stimulator	57
3.1.2 Surface electromyography (EMG) recording.....	60
3.2 Preliminary TMS experiments	63
3.2.1 Experimental procedure	63
3.2.2 Results of the experiment.....	64
3.2.3 Discussion of the preliminary TMS experiments.....	69
3.3 Practical considerations.....	71
3.3.1 Contact force between the coil and subjects' head	71
3.3.2 Effects of the inevitable movement.....	73
3.3.3 Discussion	76
3.4 Summary	78
CHAPTER 4 SYSTEM DESIGN AND PRELIMINARY EVALUATION	79
4.1 System concept design	80
4.1.1 System architecture	80
4.1.2 Proposed control system	81
4.2 Preliminary evaluation on Polaris Spectra optical tracking system	83
4.2.1 Overview of Polaris Spectra system	84

4.2.2 The utilization of the tool	87
4.2.3 Polaris Spectra calibration.....	91
4.2.4 Utilization of Quaternions to parameterize the rotation.....	94
4.3 Robot manipulator arm	96
4.3.1 General characteristic of Staubli TX60 robot arm	98
4.3.2 General characters of Staubli CS8C controller	98
4.3.3 Ethernet communication	99
4.4 Force sensor systems.....	104
4.4.1 ATI force/torque sensor system	105
4.4.2 Pneumatic collision sensor.....	108
4.5 Definition of the tool centre point.....	111
4.5.1 TCP definition by teaching a reference point	112
4.5.2 TCP definition using sphere equation.....	114
4.5.3 TCP definition using the tracking system	114
4.7 Summary	116
CHAPTER 5 COORDINATE SYSTEM REGISTRATION.....	117
5.1 Hand-eye calibration	118
5.2 Implementation of hand-eye calibration	119
5.3 Solutions for the Shiu-Ahmad homogenous matrix equation $AX = XB$.	123
5.3.1 Calibration using Tsai-Lenz method.....	123
5.3.2 Calibration using DH method	126
5.4 Calibration experiment.....	129
5.4.1 Method of error checking.....	129
5.4.2 Experimental procedure of the calibration using Tsai-Lenz and DH method.....	131
5.4.3 Results and conclusion	132

5.4.4 Practical consideration	135
5.5 Coordinate systems registration in TMS system.....	137
5.5.1 Definition of the subject's head coordinate system	138
5.5.2 Transformation between passive tool and head coordinate systems.....	138
5.6 Summary	139
CHAPTER 6 DESIGN OF A COMBINED POSITION AND FORCE CONTROLLER	140
6.1 Position tracking control strategy.....	141
6.2 The combined force and position controller	143
6.2.1 The conventional hybrid controller	143
6.2.2 The combined force and position controller	145
6.3 Introduction of experimental plan.....	146
6.4 Preliminary testing of the position tracking method	147
6.5 Testing of the combined position and force controller	153
6.5.1 Head motion simulator	153
6.5.2 Experimental design and results	155
6.5.3 Practical considerations.....	179
6.6 Summary	187
CHAPTER 7 SOFTWARE DESIGN	189
7.1 QNX real-time operating system	190
7.2 VAL3 Robot Language	190
7.3 Design of the software system	192
7.3.1 Overview of the software system.....	193
7.3.2 Serial communications in QNX operating system.....	197
7.3.3 Ethernet communication using socket	199
7.3.4 Calibration module.....	201

7.3.5 Hybrid controller module	204
7.3.6 Software Design of Image module.....	212
7.4 Summary	218
CHAPTER 8 CONCLUSION AND FUTURE WORK	219
8.1 Conclusions	219
8.2 Future work	223
8.3 Presentations and publications	225
REFERENCES.....	226
Appendix 1	239
A 1.1 Determining the segment angle.....	240
A 1.2 Tools specifications.....	242
A 1.2.1 Tool 1	242
A 1.2.2 Tool 2	243
A 1.3 Robot characteristics	244
A 1.3.1 Robot specification.....	244
A 1.3.2 Motion range and demesions (Unit : mm)	245
A 1.3.3 Work envelope (Unit : mm)	247
A 1.3.4 Twist (Unit : mm).....	248
A 1.3.5 Components locations	249
A 1.3.6 Signal connections	250
A 1.3.7 MCP and WMS	252
A 1.3.8 Three demension of the robot controller	252
A 1.4 Geometrical information of the coil	254
A 1.5 Quaternions	255
A 1.6 Dimensions of components of force sensor system	258
A 1.6.1 Dimension of DAQ power supply.....	258

A 1.6.2 Dimension of the transducer	259
A 1.6.3 Properties of the transducer	260
A 1.7 Method of creating a tool definition file for the passive tools	260
Appendix 2	267
A 2.1 System architecture of QNX	267
A 2.1.1 QNX Neutrino Microkernel	268
A 2.1.2 Inter-process Communication	269
A 2.1.3 Threads scheduling.....	271
A 2.1.4 Photon Application Builder (PhAB)	272
A 2.2 Building an application using PhAb	273
A 2.3 VAL3 programming	274
A 2.3.1 Variable types in VAL3 programming.....	274
A 2.3.2 Programming with VAL3.....	275
A 2.4 Automatic ‘hot-spot’ finder module.....	278
A 2.5 Detailed information of the system and port status.....	280
A 2.6 Code for parsing the raw data	280
A 2.7 ‘C’ structure of the Analyze head file	281

List of Figures

Figure 1. 1 Magnetic coil and Current flow [Hallett, 2000]	2
Figure 1.2 Image-guided TMS. The stimulation site is displayed on the images [Brainsight, 2010].	3
Figure 1.3 Effect of coil orientation on MEP amplitude. Each panel shows averages of 10 MEP responses from the active first dorsal interosseous muscle; baseline activity was maintained constant throughout by visual feedback of mean rectified EMG. Different coloured traces show responses with the coil in the locations shown by the photographs. A, roll. B, pitch. C, yaw. [Data gathered in the Baker laboratory].	4
Figure 2.1 Illustration of Direction of Current Flows in a Magnetic Coil and the Induced Current in the Brain.....	9
Figure 2.2 Magnetic Coil Shape Determines the Pattern of the Electric Field.....	10
Figure 2.3 Example of a silent period to TMS. The effect of TMS on a motor area that represents a voluntarily contracted hand muscle. Responses are recorded from the left first dorsal interosseus muscle (FDI) [Kobayashi and Pascual-Leone, 2003].	12
Figure 2.4 Representative examples of MEPs produced in the APB muscle of one subject by rTMS trains at 150% of the threshold intensity and frequencies of 1, 5, 10 and 20 Hz. The numbers on the left (1,10 and 20) indicate the number of pulses during the rTMS train[Pascual-Leone et al., 1994].....	15
Figure 2.5 Top view of 10-20 system illustration. Nineteen standard positions in the conventional 10–20 system are shown (red circles). Nasion and inion are indicated as Nz and Iz [Okamoto et al., 2004]......	18
Figure 2.6 Profile view of 10-20 system illustration.	19
Figure 2.7 Surgical Tool Navigator TM with workstation, 3D camera system and subject with the magnetic coil.[Herwig et al., 2001b]	20
Figure 2.8 Depictions of fMRI and TMS, showing (a) anatomical MRI with coil representation, where the coil is centered above the primary visual cortex and in the inset the TMS coil is highlighted with vitamin E pills, and (b) fMRI BOLD activity evoked from single-pulse TMS on the primary visual cortex [Wagner et al., 2007].	21

Figure 2.9 MEPs of the left masseter muscle evoked by contralateral TMS in a representative subject. For each of the 8 different coil orientations, 3 MEPs were superimposed. The greatest and most consistent amplitudes were found at 120° and 300° , while clearly smaller MEPs were found at 30° and also between 210° and 255° [Guggisberg et al., 2001].....	23
Figure 2.10 Timeline of surgical robots [Gomes, 2011].....	24
Figure 2.11 Overall organization of the surgical system [Kwoh et al., 1988].....	25
Figure 2.12 Da Vinci robotic system(Courtesy of Intuitive Surgical Inc., Mountain View, CA).	25
Figure 2.13 ROBODOC system (ROBODOC [®] , Curexo Technology Corporation, CA, US).....	26
Figure 2.14 NeuroMate (CUREXO Technology Corporation, CA, US).....	27
Figure 2.15 The Pathfinder surgical robot (left) and the view of the camera surrounded by the light ring mounted at the end of the arm (right) [Morgan et al., 2003].	28
Figure 2.16 CyberKnife robotic radiosurgery system (courtesy of Accuray, Sunnyvale, CA).	29
Figure 2.17 Close loop control of the versatile interactive surgical assistance system [Wahrburg et al., 2004].	30
Figure 2.18 The iTMS robotic system with (A) the computer for processing and controlling planned positioning of the TMS coil, (B) the Neuromate robot arm, (C) the added TMS coil holder, and (D) the 6th axis for coil rotation [Lancaster et al., 2004].	31
Figure 2.19 The six main components for robot-guided TMS: (a) TMS coil, (b) robot, (c) robot controller, (d) computer, (e) Polaris tracking system, (f) headband [Finke et al., 2008](left) and Application setup[Matthaus et al., 2006](right).....	32
Figure 2.20 The CAD model of the robot [Lebosse et al., 2007].	32
Figure 2.21 The kinematics scheme of the robot [Lebosse et al., 2007].....	33
Figure 2.22 Polaris Spectra position sensor (c), the active markers (b) and the passive markers (a).	36
Figure 2.23 Camera-robot configuration used in visual servo control system [Kragic, 2001]	37

Figure 2.24 Dynamic position based look and move structure. ${}^c X_o$ is the current pose of the camera with respect to the object, ${}^c X_o^*$ is the desired pose.....	38
Figure 2.25 Position based visual servo (PBVS) structure	39
Figure 2.26 Registering the virtual head to the real head using landmarks and surface points. Left: Surface points and landmarks from the real head. Middle: Virtual head model. Right: VR model registered to the real head [Matthaus et al., 2005].....	40
Figure 2.27 Architecture of the functional brain mapping system [Ettinger et al., 1998]	40
Figure 2.28 Laser scan of a subject's head. Collected laser points are shown as white curves [Ettinger et al., 1998].	41
Figure 2.29 Registration process[Ettinger et al., 1998].	42
Figure 2.30 An optical tracking system is localizing the position of three rigid bodies: one attached to the bone of the patient and two attached to the end-effector parts for double check on the position of the end-effector[Burgner et al., 2008].	44
Figure 2.31 Transformation of coordinates [Wahrburg and Kerschbaumer, 1998]...45	
Figure 2.32 Coordinate systems [Knappe et al., 2003].	46
Figure 2.33 The RobaCKa system, consisting of a CASPAR robot (1),the robot controller (2), a Polaris infrared navigation system (3), the subject (4), a “sensor PC unit (5) [Korb et al., 2003].	47
Figure 2.34 Example of hybrid control [Yoshikawa, 2000].	49
Figure 2.35 Vision-force system [Leite and Lizarralde, 2009].	50
Figure 2.36 Setup of impact test using robot arm and dummy [Haddadin et al., 2009].	53
Figure 2.37 Classification of undesired contact scenarios between human and robot[Haddadin et al., 2009].	54
Figure 3.1 Principles of transcranial magnetic stimulation. The current in the coil generates a changing magnetic field (B) that induces an electric field (E) in the brain. The picture at the upper right depicts motor cortex stimulation and the trajectory of the pyramidal axons. At the microscopic level, the electric field E affects the transmembrane potentials and may lead to local membrane depolarization and subsequent neural activation. [Pascual-Leone et al., 2002].	56

Figure 3.2 Major experimental devices used in the experiment.	57
Figure 3.3 The major units of a typical magnetic stimulator.	57
Figure 3.4 Two stimulator connected via a Bistim unit.	58
Figure 3.5 Single circular coil and figure of 8 shaped coils along with their induced electric field profiles.	59
Figure 3.6 Calculated induced electric field 10mm against distance from the coil centre. The horizontal axis shows the radial displacement from coil centre, and the vertical axis shows the amplitude of the electric field [Pascual-Leone et al., 2002].	60
Figure 3.7 Detection of EMG signals.	60
Figure 3.8 Disposable snap electrodes used in the application. One side is attached on subject's skin, and the other side is snapped on a cable and connected to the circuit.....	61
Figure 3.9 Differential amplifier configuration of electrodes.....	61
Figure 3.10 Layout of the amplifier, opto-isolator and filter.	62
Figure 3.11 An example of the averaged TMS response.	65
Figure 3.12 Results of single pulse stimulation in the passive state. Averaged rectified EMG from 1DI and MEP recruitment for a Chinese (a, c) and Caucasian (b, d) subject following single pulse stimulation at different intensities. Vertical dashed lines show the response region. Filled circles show responses significantly different from zero (t-test, $P < 0.05$). Error bars show standard errors. (e)-(i), box plots showing results across the subject population for online threshold, offline threshold, I_{50} (the stimulus intensity producing a half- maximal response), parameter k (indicating how quickly responses grew with increased intensity) and MEP latency. The boxplots show the maximum, minimum, 1 st quartile, median and 3 rd quartile.....	66
Figure 3.13 Results of single pulse stimulation in the active state. Averaged rectified EMG from 1DI and MEP recruitments are shown from a Chinese (a, c) and a Caucasian (b, d) subject following single pulse stimulation at different intensities. Vertical dashed lines show the response region. Filled circles show responses significantly different from zero (t-test, $P < 0.05$). Error bars show standard errors. (e)-(g), population data for active threshold, I_{50} and parameter k . The boxplots show the maximum, minimum, 1 st quartile, median and 3 rd quartile.....	68

Figure 3.14 Results of silent period measurement. Averaged rectified EMG from 1DI is shown from a Chinese (a) and a Caucasian (b) in the active state. The vertical dotted lines indicate the silent period offset at different intensities. (c)-(d), relationship between silent period offset and intensity, for the subjects shown in (a,b). Overlain line is the best fit straight line, with slope calculated as shown above each plot. (e)-(g), boxplots of population data on silent period at threshold and threshold+10%, and the slope of the silent period offset-intensity relationship. The boxplots show the maximum, minimum, 1st quartile, median and 3rd quartile.....69

Figure 3.15 Measurement of the contact force by attaching a force sensor on the coil. 71

Figure 3.16 Contact force recorded from 8 subjects respectively. Error bar show the maximum and minimum force during the measurement. 72

Figure 3.17 Averaged response on 1DI using different contact force between the coil and subject's head. Vertical lines show the response region (data gathered from Baker's lab). 72

Figure 3.18 Different amplitudes of the muscle response obtained due to inevitable movements during the experiment. Vertical lines show the response region, n indicates 10 stimuli are delivered..... 73

Figure 3.19 Passive tool is attached on the forehead of the subjects. 74

Figure 3.20 Distribution of the relative distance from the original position. (a) shows distances when the subjects trying to maintain their position. (b) shows distances when the subjects trying to relax their muscles or disturbed by outside environment. 75

Figure 3.21 Distribution of the speed when subjects moving their heads. 76

Figure 3.22 Preliminary sequence of operating the system. 78

Figure 4.1 Conceptual layout of the system. The subject sits on a chair, a robot arm holds the coil in contact with the subject's head. A position sensor and force sensor are used for the data recording.....80

Figure 4.2 Proposed control system..... 81

Figure 4.3 Coil and force transducer mounted on the robot. 82

Figure 4.4 Content of pre-operative acquisition mode..... 83

Figure 4.5 Polaris Spectra optical tracking system. (a) Position sensor (front view); (b) Passive tool (Passive markers affixed on the rigid body)[NDI, 2005].	84
Figure 4.6 Infrared light floods the surrounding area[NDI, 2005].	84
Figure 4.7 Passive markers reflect infrared light back to the position sensor[NDI, 2005].	85
Figure 4.8 Position sensor receives light from marker reflections[NDI, 2005].	85
Figure 4.9 Determining a marker position	86
Figure 4.10 Global coordinate system of Polaris spectra.	86
Figure 4.11 Pyramid volume.	87
Figure 4.12 Phantom markers and real markers.	88
Figure 4.13 Segments A and B form one segment pair, segments C and D form another segment pair. A and C, and B and D are like segments, so segment pair AB and segment pair CD are like segment pairs. In order to track the two tools with like segment pairs together, segment angles α and β must differ by at least 2 degrees.	90
Figure 4.14 Two four-marker tool.	91
Figure 4.15 Newwall machine with digital readout system.	92
Figure 4.16 Relative positions of the position sensor and the tool.	93
Figure 4.17 The process of inducing <i>gimbal lock</i> . The red frame is the tracking system frame and the blue one is the passive tool frame.	95
Figure 4.18 FTP ‘client-server’ connection via Ethernet.	101
Figure 4.19 connection between QNX and CS8C controller via Ethernet.	102
Figure 4.20 Timing test for data exchange.	103
Figure 4.21 Time intervals with 4ms setup.	104
Figure 4.22 Applied Force and Torque Vectors on Transducer.	105
Figure 4.23 Outline of the force system[ATI, 2010].	106
Figure 4.24 Differential Connections to a Data Acquisition System[ATI, 2010].	106
Figure 4.25 Steps to obtain the force and torque.	107
Figure 4.26 Force and torque data without loading.	108
Figure 4.27 Pneumatic collision sensor (RAD's Ultimatic™, Module U4618).	108
Figure 4.28 Compliance adjustment features [RAD, 2009].	109
Figure 4.29 Cross section through the collision sensor [RAD, 2009].	110

Figure 4.30 Calibration of the pneumatic collision sensor. (a) Configuration for calibrating the regulator and pneumatic sensor. (b) The relationship between the pressure and the compressive force.....	111
Figure 4.31 Assembly of robot end-effector.	111
Figure 4.32 Left hand law is used to defined coordinate system (view from beneath of the coil). The X-axis of coil frame aligned with the handle of the coil.....	112
Figure 4.33 Transformations after teaching the reference point.	113
Figure 4.34 TCP of the coil make a contact with the reference point (the corner)..	113
Figure 4.35 Definition the TCP with a pointer.	115
Figure 4.36 Transformations used for defining the TCP.	115
Figure 4.37 Calibration results using the three methods.....	116
Figure 5.1 Hand-eye configuration of a visual servo system [Dornaika and Horaud, 1998].....	118
Figure 5.2 Finding the mounting position of a camera by solving a homogeneous transform equation of the form $AX = XB$ [Shiu and Ahmad, 1987].	119
Figure 5.3 All coordinate systems and transformations used in the calibration process.	120
Figure 5.4 Robot moves from position i to position j , matrix X is the desired transformation.	121
Figure 5.5 Transformations of the system when robot moves from position i to j	122
Figure 5.6 Two robot position i and j , matrix Y is the desired transformation.....	127
Figure 5.7 All coordinate systems and transformations in the calibration process..	130
Figure 5.8 Experimental setup. The position sensor is mounted on a movable stand, and a passive tool is mounted on the robot end-effector.....	132
Figure 5.9 RMS error in rotation and translation with two difference form equations respectively.	134
Figure 5.10 Distribution of the error in translation R_{trans} (mm) and rotation R_{rotas} (radians).....	135
Figure 5.11 Rotational and translational errors with different inter-position rotation angle	136
Figure 5.13 Coordinate systems and transformations in the robotic TMS system ..	137

Figure 5.14 Transformations between passive tool and head coordinate system	139
Figure 6.1 Changes in transformations when subject moves from position 1 to position 2.....	142
Figure 6.2 Control structure of the tracking system.....	143
Figure 6.3 Conceptual organisation of conventional hybrid controller [Raibert and Craig, 1981].	144
Figure 6.4 Combined position and force controller.	145
Figure 6.5 Structure of the test rig.	148
Figure 6.6 Changes of trajectories of both passive tools with different proportional gains. Blue curves are the trajectory of the tool mounted on the X-Y table and red curves are that of the tool on the robot arm.	150
Figure 6.7 Changes of trajectories of both passive tools with different integral gains. Blue curves are the trajectory of the tool mounted on the X-Y table and red curves are that of the tool on the robot arm.	151
Figure 6.8 System performance at different speeds. The error bars show the standard derivation.....	152
Figure 6.9 Test rig for the combined position and force controller.	153
Figure 6.10 Staubli robot moves down and makes contact with the ball.....	154
Figure 6.11 Relationship between the force and deflection in the Z axis. Error bars show the standard deviation.	155
Figure 6.12 Effect of proportional gain on force step response.	156
Figure 6.14 Effect of integral gain on force step response.	158
Figure 6.15 ITAE of different K_{FI} . Error bar is the deviation.	158
Figure 6.16 PUMA moves the ball in Z axis. (b) is a simplified picture from (a)...	159
Figure 6.17 Force control with constant gains $K_{FP} = 0.0012$ $K_{FI} = 0.002$	159
Figure 6.18 Force control with larger constant gains.....	160
Figure 6.19 Conventional gain scheduled system.....	161
Figure 6.20 Force response of the system with different K_c	164
Figure 6. 21 ITAE and IAE using different K_c	165
Figure 6.22 System response on the moving ball with different gains.	166
Figure 6. 23 ITAE using different K_c	168

Figure 6.24 Force control in Z axis using gain scheduling method.	168
Figure 6.25 PUMA moves the ball in XY plane.	169
Figure 6.26 Behaviour of position and force controller with the position disturbance input.	170
Figure 6.27 Comparison of changes of trajectories between the Staubli robot and ball. The error display on the picture shows the mean error during the movement and in steady state respectively.	170
Figure 6.28 Profiles of trajectories when the disturbance is a circle. The error display on the picture shows the mean error during the movement and in steady state respectively.	171
Figure 6.29 Trajectories when the system deals with 100mm displacement in X and Y axes. The error display on the picture shows the mean error during the movement and in steady state respectively.	172
Figure 6.30 PUMA moves the ball in 3 direction simultaneously.	173
Figure 6.31 Combined position and force control with a desired force of 2N. The error display on the picture shows the mean error during the movement and in steady state respectively.	173
Figure 6.32 Circular path in 3D space.	174
Figure 6.33 Position and force response to the 3D circle disturbance.	175
Figure 6.34 Trajectories when the system deals with 100mm displacement in X,Y and Z axes. The error display on the picture shows the mean error during the movement and in steady state respectively.	176
Figure 6.35 TCP of the PUMA robot.	177
Figure 6.36 System behaviour when there is an orientation disturbance input. . All rotation axes are with respect to coil frame.	178
Figure 6.37 Trajectories of the Staubli robot followed by a rotation input about the Z axis. All rotation axes are with respect to coil frame.	179
Figure 6.38 System force response using different K_{FI}	180
Figure 6.39 System behaviour when the Staubli robot tracks the movement of the ball with a fast response time.	181
Figure 6.40 ITAE of force response with different speeds. Error bars show the standard deviation.	182

Figure 6.41 System behaviour with different desired force.	183
Figure 6.42 Position errors with 5N contact force. The error display on the picture shows the mean error during the movement and in steady state respectively.	184
Figure 6.43 ITAE value of different desired forces. Error bars show the standard deviation.	185
Figure 6.44 ITAE value of different update rate.	185
Figure 6.45 System behaviour with natural data.	186
Figure 7.1 Software architecture.	193
Figure 7.2 System operating sequence.	194
Figure 7.3 Flow chart of the application running on the CS8C.	196
Figure 7.4 TCP socket flow diagram.	200
Figure 7.5 Calibration module: robot position collection and solution.	201
Figure 7.6 Control window of calibration module.	202
Figure 7.7 Structure of force sensing control application.	205
Figure 7.8 Software structure of position sensor system.	207
Figure 7.9 Flow chart for configuring the Polaris system.	208
Figure 7.10 An example of the data packet returned by the position sensor.	209
Figure 7.11 State machine of the tracking module.	211
Figure 7.12 DICOM file structure.	213
Figure 7.13 Image module structure.	214
Figure 7.14 MRI image viewer. The picture in the viewer is a 2D image, and the yellow point indicates the current stimulation position of the head. The coordinates of the point (the origin is the left upper corner of the image) are (230,45,16). The Z-coordinate depends on the slice number of the image that is loaded.	215
Figure 8.1 The CAD model of the robots. (a) is the model designed for TMS proposed by Lebosse et al.[2007]. (b) is the model designed by Ramrath et al.[2007] for Stereotactic Surgery.	224

List of Tables

Table 3.1 Main specifications of the robotic TMS system	77
Table 4. 1 Robot comparison.....	97
Table 4.2 Five-layer TCP/IP protocol.....	99
Table 4.3 A sample Ethernet frame.....	100
Table 6.1: An example of gain scheduling for force control.....	162
Table 6.2: Gain scheduling for force control	163
Table 6.3 ITAE and IAE of the system force response with different K_c when the ball is stationary	165
Table 6.4 System responses with different gains	167

GLOSSARY OF TERMS

1DI	First dorsal intercosseous
EEG	Electroencephalography
EMG	Electromyogram
fMRI	Functional magnetic resonance image
MEP	Motor evoked potential
MVC	Maximum voluntary contraction
PET	Positron emission tomography
SP	Silent period
SPECT	Single photon emission computed tomography
RMS	Root mean square
TCP	Tool centre point
TMS	Transcranial magnetic stimulation
$\{H_a^b\}$	Homogeneous transformation of a frame relative to b frame

CHAPTER 1

INTRODUCTION

Transcranial magnetic stimulation (TMS) has been used for more than 20 years and is a powerful, painless and non-invasive method for studying various aspect of human neurophysiology. It is an excellent research tool in brain physiology that has the potential of being developed into a systematic diagnostic or therapeutic tool in studying neurological and psychiatric disease such as: depression, Parkinson's disease, stroke and so on. Since TMS has significant advantages that other techniques do not have, many neuroinstitutes and centres are studying it. However, most of the studies focus on the neuroscience and little progress has been made in automating the applied process of TMS stimulation. An automate TMS system is not only essential to improve the accuracy of the TMS response, but also important to use TMS technique in clinical routine.

1.1 Background

The principle of TMS is to stimulate neural elements through an electric field induced by a rapidly changing magnetic field. The stimulation device, which consists of a coil of wire is placed over the scalp and a brief high-current pulse is generated in the coil so that the magnetic field passes through the skull and induces an electric field (Figure 1. 1). The induced current flow activates neurons in the cerebral cortex

[Day et al., 1987, Noirhomme et al., 2004, Edgley et al., 1990]. The effect on the motor cortex, particularly the corticospinal tract has been extensively studied using this technique. A focal muscle twitch produced by TMS is called motor-evoked potential (MEP), and it can be recorded using electromyography (EMG). TMS is also widely used for probing the function of many different parts of the cerebral cortex, to excite, inhibit and assess aspects of excitability [Hallett, 2000, Kujirai et al., 1993, Ziemann et al., 1998]. Changes in cortical function can be promoted by repetitive TMS (rTMS) [Siebner and Rothwell, 2003]. Because the magnitude and the direction of rTMS-induced plasticity depend on specific stimulus configurations, rTMS has been used for measuring the corticospinal excitability and treatment of neuropsychiatric disorders such as major depression [Couturier, 2005], Parkinson's disease [Cantello et al., 2002], tinnitus [Kleinjung et al., 2005] and aphasia [Martin et al., 2004].

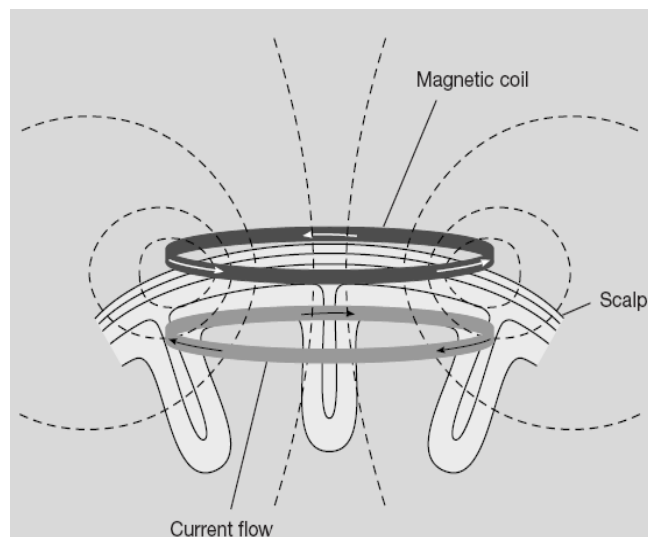


Figure 1. 1 Magnetic coil and Current flow [Hallett, 2000]

There are two key issues for the placement of the stimulation coil. The first one is how to acquire the optimal stimulation position and orientation. The second is how to maintain the predefined position and orientation during the stimulation session. Currently, there are two methods used to acquire the stimulation site:

- The coil is initially positioned at a start position known to be reasonably good for eliciting MEP in the target muscle, by an experimenter according to his experience. The experimenter then manipulates the coil position around the

start position until the maximum EMG response is found. This position is defined as the stimulation site. The 10-20 EEG system is used for the regions where there is insufficient response can be triggered to indicate correct positioning [Herwig et al., 2003].

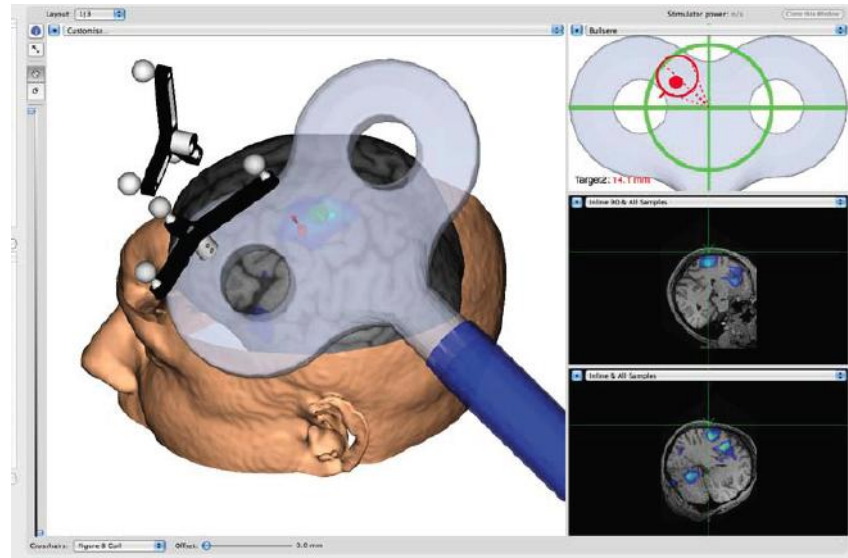


Figure 1.2 Image-guided TMS. The stimulation site is displayed on the images [Brainsight, 2010].

- Medical image technologies (MRI, fMRI and CT etc.) are used to precisely locate the stimulation site by registering the subject's head to the image data. The image data can be segmented to extract the skin surface, for use in registration, and the internal structures, for use in visualization [Ettinger, 1997, Lancaster et al., 2004]. The location of the coil is digitized relative to the internal structures so that image data can co-register with the actual subject. The images can then be used to indicate the stimulated tissue (Figure 1.2).

Once the optimal site has been found, it is necessary to fix the coil in the position, as the position and orientation of the coil critically affects the cortical activity produced by TMS [Tings et al., 2005] (Figure 1.3). However, almost all of these systems use hand-held stimulator coils and inadvertent small movements may lead to imprecise responses. Recently, to solve this problem, devices such as face masks, helmets or

bite bars have been used to ensure the coil is fixed relative to the head. But this is uncomfortable for the subject, especially in a long duration experiment.

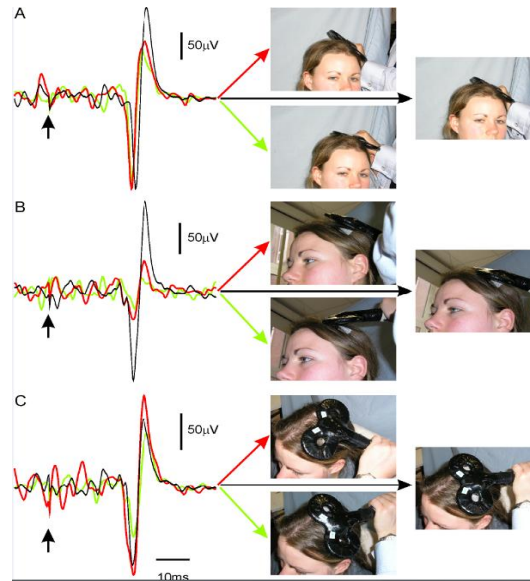


Figure 1.3 Effect of coil orientation on MEP amplitude. Each panel shows averages of 10 MEP responses from the active first dorsal interosseous muscle; baseline activity was maintained constant throughout by visual feedback of mean rectified EMG. Different coloured traces show responses with the coil in the locations shown by the photographs. A, roll. B, pitch. C, yaw. [Data gathered in the Baker laboratory].

Two major issues have been identified from the current TMS system, as follows:

- Response variability is excessive using a hand-held coil. Although the stimulation site can be precisely located using image-guide TMS system, the position and orientation is difficult to maintain by a human experimenter. The hand-held coil may be tilted or moved away from the target region during the experiment, which results in a loss of accuracy. The coil can be manually moved on the head of the subject to follow pre-planned landmarks or a visual navigation system, however, it is impossible to achieve a precise motion manually. Furthermore, because of variability, more measurements for diagnosis must be carried out to obtain an averaged and reliable figure, which takes much more time.
- The subject is restrained when rigid fixation is employed, which is rather uncomfortable. It is inconvenient for clinical use, and experimental duration

is also severely curtailed. More importantly, the response to TMS has been shown to change severely if the subject is tense [Kiers et al., 1993].

1.2 Aims and Objectives

To overcome the problems in current TMS technology and to make TMS of the cerebral cortex more reliable, a prototype robotic TMS system will be designed and constructed. Since robots can provide excellent locating and holding capabilities, the proposed system will position and hold the stimulus coil over a fixed location and at a fixed orientation on the non-fixed head of the subject. In addition, both small head movements and contact force between the subject and coil will be tracked in real time by the system during the TMS experiments.

To satisfy this aim, several objectives have been identified:

- Carry out a comprehensive review of current TMS techniques to establish the current state of the art.
- Design a prototype robotic TMS system.
- Implement a registration system to link all coordinate systems together (coil position/orientation, head position/orientation).
- Develop a method to monitor both force and position feedback.
- Develop an appropriate real-time control system to link all the functions.
- Design a test rig to test the presented algorithms and assess their performance.

The development of the prototype TMS application system should naturally lead to an improved performance, and allow evaluation of this technique in experimental and clinical use.

1.3 Hypothesis

The hypothesis of this work is: is it possible to develop a robotic TMS system to conduct TMS experiments, which can position and hold the stimulus coil over a fixed location and at a fixed orientation on the unconstrained head of the subject; and can the robot track both small head movements and contact force between the

subject and coil? It is proposed that the use of the system will provide more accurate experimental results for researchers and be more comfortable for the subjects.

1.4 Outline of thesis

This thesis is organised as follows:

A brief overview of the principles, devices, diagnostic and therapeutic applications of TMS is reviewed in Chapter 2, along with the background to existing robotic surgical systems and robotic TMS systems. In addition, previous and recent research that is relevant to this application is addressed.

Chapter 3 focuses on the preliminary TMS experiments and data collection using the current hand-held TMS. The major components of the TMS system along with the method of electromyogram (EMG) signal collection are introduced; the movement and contact force recorded from real subjects during the experiment are also specified.

The design of the robotic TMS system; including the architecture and control system developed in Chapter 4, along with the configuration and evaluation of the key components of the system, including the Polaris Spectra tracking system, Staubli TX60 robot arm and an ATI F/T force sensor.

Chapter 5 discusses the coordinate system registration techniques and includes a comparison between two different methods. The position tracking and combined position and force tracking algorithms as well as the assessment of these algorithms are addressed in Chapter 6. Chapter 7 presents the design of the software system which is used to realise and link all the functions of the system.

Finally, the conclusions along with a summary achievement and suggestion of further work are provided in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the relevant literature of Transcranial Magnetic Stimulation (TMS) and establishes the current state of the art. Although the project focuses on developing the robotic system, the principles, devices, diagnostic and therapeutic applications are reviewed first to help to establish a more specific concept and better understanding of TMS, which should be considered when developing the robotic system. The following section discusses the localization of the coil placement and methods used in current hand-held TMS system, which can also be utilized in a robotic system. Both robot assisted surgical and TMS systems are then reviewed to identify the advantages and disadvantages of the existing systems, followed by more specific descriptions on relevant technologies including the tracking technologies, registration and trajectory planning. Finally, the safety issues are presented. Other relevant literature will be discussed as background to the work in the following chapters.

2.1 Principles and devices of TMS

The technique of TMS was firstly introduced by Barker et al. [1985]. As a unique non-invasive and painless tool for the electrical stimulation of human brain and other neural tissue, TMS has been widely used as a research tool in brain physiology. It

can influence most parts of the brain, but most studies have been carried out on the motor cortex where a focal muscle twitch can be produced, which is called motor-evoked potential (MEP). There is also the likelihood of it being developed into a systematic diagnostic or therapeutic tool in neurological and psychiatric disease.

A technique named transcranial electrical stimulation (TES) has many similarities with TMS. Merton and Morton [1980] found that the motor areas of the human brain can be stimulated through the intact scalp by a brief high-voltage electric shock. This also leads to a relative muscle response (MEP). Due to the activation of pain receptors as a result of electrical stimulation, the subject feels extreme pain. This is the main reason why TES is only used in some selective areas while TMS has been more widely adopted.

Electromagnetic induction is the basic principle of TMS. If a strong and brief pulse of current passes through a coil which normally is placed over the scalp, a changing magnetic field is generated and the magnetic pulses pass through the scalp and skull without attenuation. Consequently, a current field will be induced perpendicular to the magnetic field in the conductive tissue of the brain (Figure 2.1). The currents that flow in the brain can excite or inhibit activity in a small area of brain below the coil. The intensity, focus, location and frequency of the delivered stimuli are four main factors that can significantly affect the result of TMS on the targeted region.

The current loops with the highest density are adjacent to the circumference of the coil itself, and become weaker near the centre of the coil, and there is no current flow at its centre [Hallett, 2007]. The large TMS currents can depolarize neurons that depend on the 'activation function', which causes trans-membrane current to flow and can be described as the spatial derivative of the electric field along the nerve [Kobayashi and Pascual-Leone, 2003, Barker, 1999]. Direct stimulation of only the superficial cortex is possible with present technology since the magnetic field declines rapidly with increasing distance from the coil. Fortunately, deeper brain structure can still be affected by TMS due to the massive interconnections and redundant cortical and subcortical loops [George et al., 2002, Alexander et al., 1986].

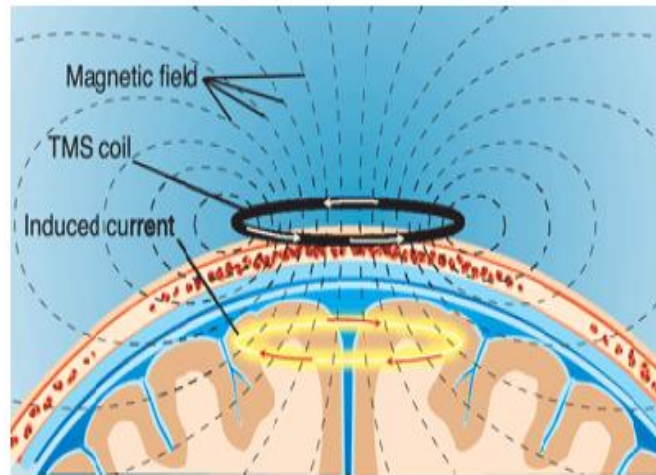


Figure 2.1 Illustration of Direction of Current Flows in a Magnetic Coil and the Induced Current in the Brain.

The magnetic coil is the only component that is directly in contact with the subject during the stimulation procedure. The focus of the magnetic field and the induced current distribution are mainly dependent on the shape of the magnetic coil. Therefore, design and selection of the coil is extremely important. Currently, single circular or figure of eight shaped coils are most commonly used (Figure 2.2). The former provides a more powerful and widely distributed electric field, which is essential to the study of central motor conduction times [Rossini and Rossi, 1998]. Because the maximal current is produced at the intersection of two loops, the latter produces a more focal density. Many researchers are exploring unique designs of the coil, concentrating mainly on increasing the focal ability and or specified stimulation. One variant coil of the flat figure of eight shape is the cone-shaped coil; the key difference is the latter has two loops angled which increases the power at the intersection [Wang et al., 2007]. The H-coil (Hesed Coil) is likely to have the ability of deep brain stimulation and without the need of increasing the intensity to extreme levels[Zangen et al., 2005].



Figure 2.2 Magnetic Coil Shape Determines the Pattern of the Electric Field.

2.2 Diagnostic application of different patterns of TMS

There are three main types of stimulation in TMS: single-pulse stimulation, a pair of stimuli separated by a time interval to the same or different brain areas, and a train of repetitive stimuli (rTMS) at various frequencies. Single-pulse TMS can depolarize neurons and evoke measurable effects and is mainly used for measuring motor threshold, central conduction time, silent-period duration, and MEP amplitude. Paired-pulse TMS can be used for the examination of intra-cortical inhibitory and facilitatory mechanisms and inter-hemispheric interactions. rTMS can modify excitability of the cerebral cortex at the stimulated site and also at remote areas along functional anatomical connections [Kobayashi and Pascual-Leone, 2003].

2.2.1 Single-pulse TMS

Single-pulse TMS is used to refer to arrhythmical stimulation not more than once every few seconds [Wassermann, 1998], and can be used to measure the motor threshold. According to [Rossini et al., 1994], motor threshold is defined as the lowest intensity required to elicit small MEP of more than $50\mu\text{V}$ peak-to-peak amplitude in at least 50% of successive trials, in resting or activated (slightly contracted) target muscles. Because the single-pulse TMS can cause MEP, the threshold can be recorded as the lowest TMS intensity by applying a single-pulse to the motor cortex. Motor threshold is likely to reflect membrane excitability and provide information on a central core region of neurons in the muscle representation [Ziemann et al., 1996]. Diseases that can influence the corticospinal tract often have

a higher motor threshold, for example: multiple sclerosis and brain or spinal-cord injury. While the amyotrophic lateral sclerosis show lower motor threshold [Chistyakov et al., 2001, Hess et al., 1987, Boniface et al., 1991, Mills and Nithi, 1997].

A second function of single-pulse TMS is measurement of central conduction time which is defined as the latency difference between the MEPs induced by stimulation of the motor cortex and those evoked by spinal motor root stimulation. It is calculated equal to the latency of the MEP induced by stimulation of the spinal motor root from that of the response to motor-cortex stimulation [Zentner, 1991]. Several studies applied F-wave measurement instead of spinal root stimulation to measure central conduction time [Rossini et al., 1994]. Central conduction time of patients with diseases such as multiple sclerosis, stroke, and amyotrophic lateral sclerosis is often delayed. In addition, central conduction time can be used to provide supporting information of diagnosis and disease prognosis but the time is still not specific to any particular disease [Escudero et al., 1998, Rossini et al., 1994].

According to [Kobayashi and Pascual-Leone, 2003], electromyographic activity can usually be arrested for a few hundred milliseconds after the MEP when a single-pulse TMS is applied to a subject's motor cortex (Figure 2.3). This silent period is normally defined as the time from the end of the MEP to the return of voluntary electromyographic activity. The silent period also refers to the time interval from stimulus delivery to the return of voluntary motor activity due to the difficulty of defining the end of the MEP [Triggs et al., 1992]. The first part of the silent period is due to the inhibition of spinal mechanisms (a smaller part, about 50-60ms), while the cortical inhibitory mechanisms contribute to the second part [Chen et al., 1999, Inghilleri et al., 1993]. The silent period duration was found to be abnormal in patients with diseases such as amyotrophic lateral sclerosis [Desiato et al., 1999], stroke, dystonia [Chen, 1997b] and Parkinson's disease [Priori et al., 1994]. Investigation has been carried out on patients who had showed hemiparesis and a long duration of the silent period due to serious stroke [Classen et al., 1997]. The result revealed that for patients who had impaired movement initiation, inability to

maintain a constant force, and impaired movement of individual fingers the silent period decreased with clinical improvement.

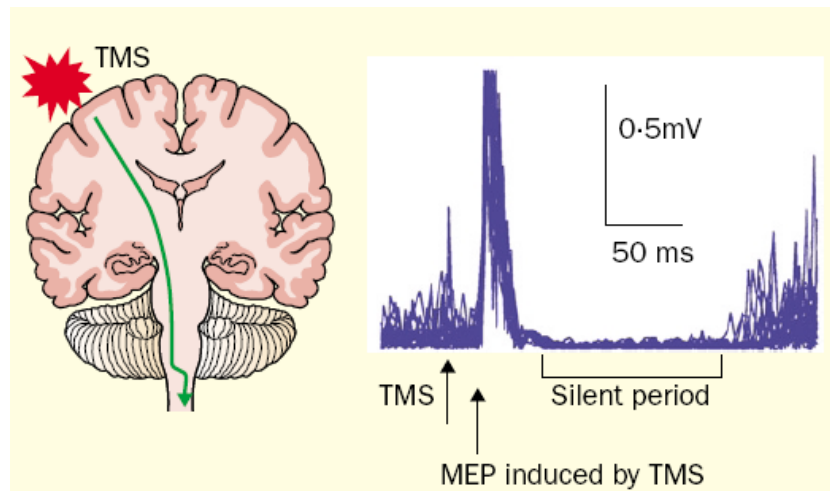


Figure 2.3 Example of a silent period to TMS. The effect of TMS on a motor area that represents a voluntarily contracted hand muscle. Responses are recorded from the left first dorsal interosseus muscle (FDI) [Kobayashi and Pascual-Leone, 2003].

Single-pulse TMS has also been used for precise mapping of motor cortex representations. This can be evaluated by stimulation over numerous different positions of the scalp and recording MEPs from different related muscles. Mapping the motor cortex with single-pulse TMS is an application of mapping with activation and it can determine the location of the optimal position for stimulation, and the centre of gravity (an amplitude-weighted representative position on the motor map) [Chen, 2000]. Some diseases can alter the motor representation, and any changes of location and excitability can finally affect the motor maps [Ridding and Rothwell, 1997], thus the map can be used for demonstrating what and where the problem is. According to [Cohen et al., 1991] and [Cicinelli et al., 1997] motor maps can be altered by conditions such as congenital mirror movements, amputations, spinal cord injury, hemispherectomy and stroke.

Another application of single-pulse TMS is measurement of MEP amplitude which is associated with the central motor conduction. This feature can be used to examine the integrity of the corticospinal tract, the excitability of motor and so on. Single-pulse TMS is also used to measure the recruitment curve which is normally used for

assessing neurons that are less excitable or located further from the centre of the activation induced by the stimulation in the spatial space [Hallett, 2007].

2.2.2 Paired-pulse TMS

Paired-pulse TMS is another significant technique for studying cortical excitability, and can be used to examine the intracortical inhibitory and facilitatory mechanisms. There are short intracortical inhibition and facilitation, and long intracortical inhibition. The former has been most widely used to demonstrate the interneuron influences. The technique is conducted by initially providing a subthreshold conditioning stimulus, which aims to initiate the intracortical influence, after this a suprathreshold test stimulus is given at a short interval. The effects of this technique depend on the intensity and interval. Inhibition is caused when the intervals less are than 5ms, and facilitation occurred with the intervals between 8-30ms. The long intracortical inhibition is given by two suprathreshold stimuli at intervals of 50-200ms. The difference between the short and the long inhibition is by increasing the pulse intensity, the former attempt to increase while the latter decline [Sanger et al., 2001]. The paired-pulse TMS technique has been applied to several psychiatric diseases, and although the results are not certain, it has shown its potential for both diagnosis and treatment. Studies have found that in subjects with neuropsychiatric disorders such as depression and schizophrenia, there are changes in the response curve. Similarity, the response curves of subjects with Parkinson's disease or dystonia showed similar abnormalities.

Another variation of paired-pulse TMS is to apply two stimuli at different regions of the brain. Ferbert et al.[1992] found that a decline of the cortical excitability occurred in the opposite region between 7-15 ms after the second stimulus. According to research on subjects with cortical myoclonus or stoke [Hanajima et al., 2001, Shimizu et al., 2002], this interaction cannot be observed, or there are some changes in it. It is thus possible to use this method to study the interactions in the brain and some movement disorders.

2.2.3 Repetitive TMS (rTMS)

Repetitive TMS is to apply TMS pulses repetitively and rhythmically to a single scalp site [George et al., 2002, Wassermann, 1998]. rTMS can be classified as ‘fast’ or ‘slow’ rTMS. In the latter the frequency is limited to 1Hz, whilst in the former are in use frequencies more than 1Hz up to a maximum stimulus rate of 60Hz are in use at present. However, the faster the frequency is, the greater the risk, and therefore, the stimulus frequency must be restricted to ensure the safety of the subject. The difference between the rTMS and repeated single pulse TMS is the stimulus interval, which normally has a delay of several seconds.

rTMS can be used temporarily to disrupt the processing of a cortical area, which make it a valuable tool for basic studies. Furthermore, long term plastic changes may also be induced following rTMS, which is the principal property used in the application. The after-effects of rTMS depend on the duration, in particular intensity and frequency of the stimuli, and according to [Pascual-Leone et al., 1994], different MEPs can be induced with variable intensities and frequencies (Figure 2.4). Normally, stimulation with high frequency and at high intensity attempts to increase rather than decrease cortical excitability. For example, Wu et al. [2000] did research using a 30-pulse rTMS at a fixed intensity (120% MEP threshold) and two different frequencies (5Hz and 15Hz). The result showed that the intra-cortical inhibition was reduced, while intra-cortical facilitation was enhanced at 15Hz. Although the inhibition also declined at 5Hz intensity, it is much shorter than that of 15Hz. On the other hand, excitability can be significantly reduced with low frequency rTMS. A significant inhibition was induced by applying a rTMS at 0.9 Hz and 115% MEP threshold for 15 minutes [Chen, 1997a]. With respect to intensity lower than the resting threshold but at high frequency, longer duration of rTMS is required for causing long term effect [Maeda et al., 2000]. Chen [1997a] reported that no change was found during one hour of 0.1 Hz rTMS. This indicates that no long term effect will be induced if the intensity and frequency is lower. According to the different long term effects induced inside and outside the motor cortical area by rTMS described above, it is possible to apply it to treat neurological disorders.

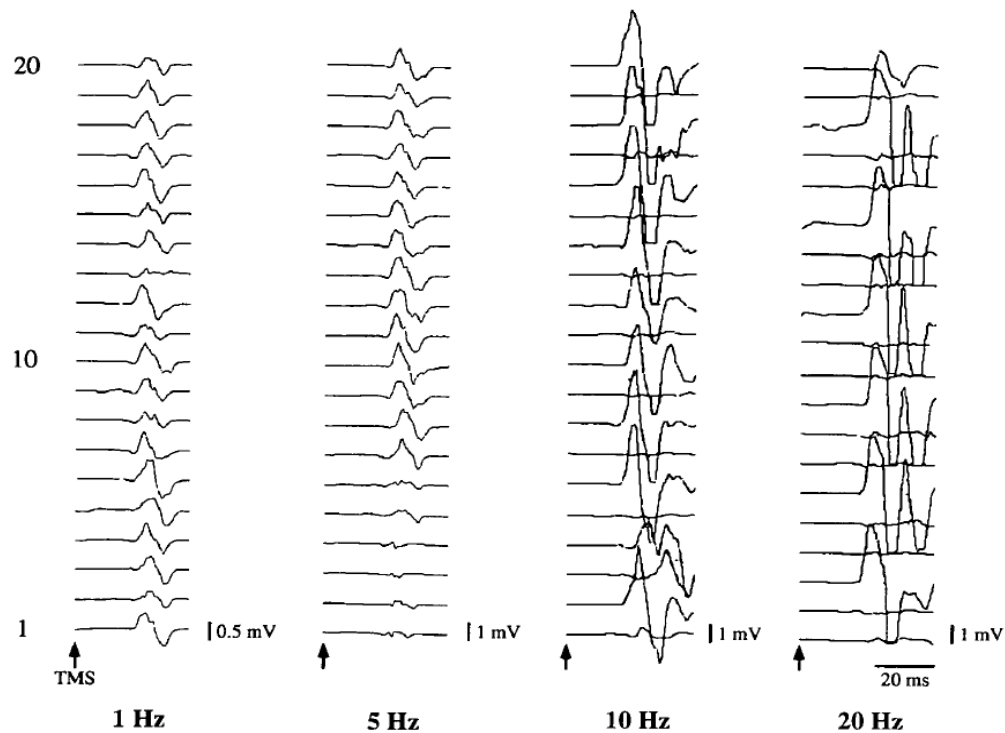


Figure 2.4 Representative examples of MEPs produced in the APB muscle of one subject by rTMS trains at 150% of the threshold intensity and frequencies of 1, 5, 10 and 20 Hz. The numbers on the left (1,10 and 20) indicate the number of pulses during the rTMS train[Pascual-Leone et al., 1994].

2.3 Therapeutic applications

TMS has been used to treat depression for several years. Low frequency rTMS was first applied over the right prefrontal cortex. However, most studies apply high frequency rTMS over the left prefrontal cortex for treating depression, which is firstly conducted by George et al.[1995]. In this study, the Hamilton Rating Scale (HRS) scores [Hamilton, 1960] for these patients declined significantly after the treatment. According to [Figiel et al., 1998], after 5 daily treatments in 21 of the 56 subjects, the HRS scores decreased by at least 60%. These studies have shown that high frequency rTMS can achieve a significant improvement in depressive symptoms. On the other hand, low frequency rTMS can also improve depression. After 10 treatments, 6 of the 14 patients experienced a greater than 50% decline in the HRS scores [Feinsod et al., 1998]. Menkes et al.[1999] obtained a similar result in their study. Kimbrell et al.[1999] reported that high frequency rTMS can achieve better results on subjects with decreased cerebral metabolism while the low frequency rTMS has better effect on hyper-metabolism. According to this result, the

most appropriate method can be chosen for treating different types of patient, and that it can prove to be more effective with longer treatment period.

Although the physiological basis is still uncertain, rTMS may be useful for treating Parkinson's disease since TMS can lead to a faster reaction time in subjects. The rationale for this would be rTMS increases the excitability so that more thalamo-cortical drive can be produced which is deficient in Parkinson's sufferers [Ben-Shachar et al., 1999]. Pascual-Leone [1994] found that reaction time and hand function in five subjects with Parkinson's disease had been improved. Studies have also found that the effect of rTMS therapy can be long term and be more substantial if the subject accepts a treatment over a longer period. However, several studies did not find any positive results [Tergau et al., 1999, Ghabra et al., 1999]. More attention should be paid to these results when rTMS is used on subjects with Parkinson's disease.

rTMS stimulation is also a therapeutic technique to promote plasticity of the brain in the damaged area, since plastic changes lead to the recovery from stroke. Khedr et al. [2005] conducted a study that employed rTMS and sham stimulation (Sham TMS is typically administered by tilting the coil 45–90 ° off the scalp, with one or two wings of the coil touching the scalp [Lisanby et al., 2001]) respectively on two groups of subjects with acute ischemic stroke. Positive results were found by comparing the disability scales measured before and after 10 days treatment and rTMS improved patients' scores more than sham stimulation. Subjects with chronic hemiparetic stroke were treated by high-frequency rTMS (Kim et al., 2006). The MEP amplitude, movement accuracy and movement time were recorded, they found that the MEP amplitude increased and the plastic change was positively associated with an enhanced motor performance accuracy.

Physiological studies found that a decrease in intracortical inhibition causes dystonia, which suggest that rTMS could have the potential for treating this condition since rTMS delivered over the primary motor cortex at 1Hz can induce an increase in inhibition [Hallett, 1998]. Siebner et al. [1999] conducted a study on subjects with writer's cramp using rTMS delivered on the motor cortex at 1Hz. Six of the sixteen subjects studied had a marked improvement of deficient intracortical inhibition and

temporary improvement in handwriting, whereas two patients noted a moderate improvement. However, the reason why some patients showed a clinical improvement and others did not is still an open question.

TMS is also used for the treatment of epilepsy [Fregni et al., 2006, Cantello et al., 2007], tinnitus [Kleinjung et al., 2005], aphasia [Martin et al., 2004] and mania [Grisaru et al., 1998], although most studies on these topics are limited and the effect of treatment are normally mild.

As presented above, TMS is a sophisticated tool used in neuroscience and various cortical activities can be induced by different types of TMS. Although there is still much work to be done before TMS can be used routinely as a therapeutic tool, it offers new insight and a method to treat many neurological and psychiatric diseases that have concerned science for a while, however, there are several factors that limit the application of the technique.

2.4 Localization of coil placement

The main issue of TMS is coil placement. This can be split into two particular problems: where to place the coil and how to maintain the position of the coil. The relation between scalp and underlying brain structures should be carefully studied before stimulation since TMS is non-invasive and the structural information under the scalp is uncertain. There are several methods for finding the target site at present, which can be classified into two main strategies: stereotactic image-guided navigation strategy and non-stereotactic navigated localization strategy.

2.4.1 Non-stereotactic navigated localization strategy

Non-stereotactic navigated localization strategy is the most commonly used method at present. The international 10-20 electroencephalography (EEG) system and the ‘standardized coil positioning’ procedures are two methods that are used most by researchers. Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity

over a short period of time, usually 20-40 minutes, recorded from multiple electrodes placed on the scalp.

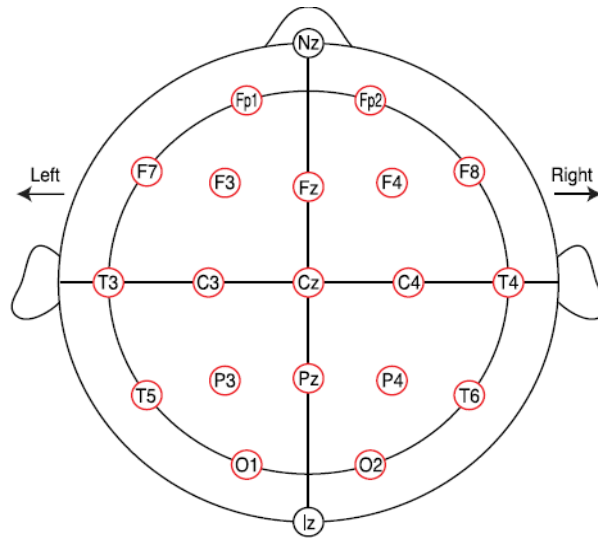


Figure 2.5 Top view of 10-20 system illustration. Nineteen standard positions in the conventional 10–20 system are shown (red circles). Nasion and inion are indicated as Nz and Iz [Okamoto et al., 2004].

The international 10-20 system is a specific system of electrode placement recommended by the International Federation of Societies for Electroencephalography and Clinical Neurophysiology (IFSECN). In a TMS application, this system can be used to guide the placement of the coil rather than electrode. Specific measurements from bony landmarks (i.e. inion, nasion and the preauricular point) are used to generate a system of lines [Jasper, 1958], which run across the head and intersect at intervals of 10% or 20% of the length of the lines (hence called 10-20 system) of their total length (Figure 2.5).

Lobe and hemisphere location are respectively marked by a letter and a number. The letters F, T, C, P and O stand for Frontal, Temporal, Central, Parietal and Occipital respectively. Note that no central lobe exists, the "C" letter is used for identification purposes only. The even numbers (2,4,6,8) refer to electrode positions on the right hemisphere, whereas odd numbers (1,3,5,7) refer to those on the left hemisphere. Two anatomical landmarks are used for the essential positioning of the coils: the nasion is the point between the forehead and the nose; and the inion is the lowest point of the skull from the back of the head and is normally indicated by a prominent

bump (Figure 2.6). Taking the dorsolateral prefrontal cortex (DLPFC) that has been studied the most for example, C3 would be the position to place the coil.

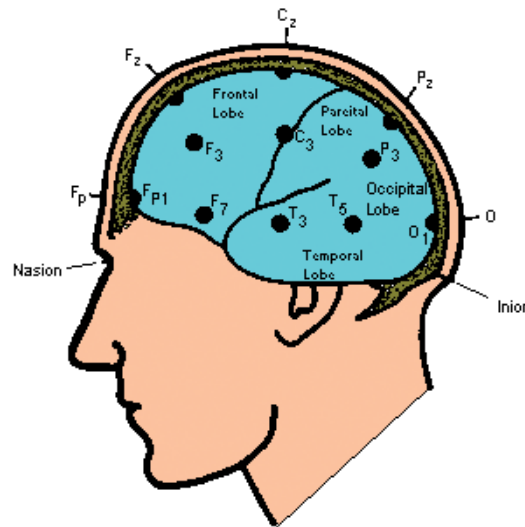


Figure 2.6 Profile view of 10-20 system illustration.

A second non-stereotactic navigated localization method was proposed by George et al. [1995] and Pascual-Leone et al. [1996]. The method is called the “standard” coil position procedure and applied by most researchers in this field [Herwig et al., 2001a]. The procedure of Pascual-Leone et al. [1996] presented for localizing the DLPFC is the motor cortex is first localized by producing a visible twitch in the first dorsal interosseus muscle (FDI) of the right hand; and an EMG recording is made to record the activation signal of the muscles. The second step is to move the coil centre 5cm anterior to the optimum scalp position from which a maximum EMG is recorded. Researchers have developed several similar methods for localizing other sites in the brain such as: the somatosensory cortex [Koch et al., 2006] and the premotor cortex [Lee and van Donkelaar, 2006].

The two methods described above are currently the most commonly used in TMS since they are both convenient and low cost. However, they are used regardless of the individual variations in the brain structure, and furthermore, there are some regions of the brain where there is insufficient feedback to indicate correct positioning. To overcome these problems, stereotactic image-guided navigation system has been developed.

2.4.2 Stereotactic image-guided localization strategy

Several research groups have developed a method which uses magnetic resonance imaging (MRI) to guide the coil into position. There are two main components: a MRI acquisition system for obtaining the MRI data and a tracking system (also called 3D digitizer) for coordinate registration, recording the position of the coil and the subject during the stimulation procedure. Figure 2.7 shows a commonly used navigational system (Surgical Tool Navigator™, Zeiss, Oberkochen, Germany).



Figure 2.7 Surgical Tool Navigator™ with workstation, 3D camera system and subject with the magnetic coil.[Herwig et al., 2001b]

There are four main steps of using this method: data acquisition including the MRI data and 3D space data of the coil and subject; followed by segmentation of the MRI data (the data is segmented into scalp surface for registration and internal cortex structure for visualization); this is followed by registering the MRI scalp surface to the 3D scalp points (obtained by the tracking system); and finally, visualization of the relative stimulation position. This method has achieved good accuracy and reliability, typically the accuracy of neuro-navigation in neurosurgery is $0.55 \pm 0.29\text{mm}$ with fixed reference frames [Kaus et al., 1997]. Schönfeldt-Lecuona et al. [2005] conducted research on accuracy of frameless stereotactic positioning of TMS, they found that the method of coil positioning has high stability and repeatability. The mean Euclidean distance between the landmark position coordinate before and after a single TMS session was 1.6mm and increasing to 2.5mm on average, after repeating two different TMS sessions. However, the results may be affected by several factors. The accuracy of the system depends on the technical limitations of

the equipment (one parameter is slice thickness), and in addition, reliance on the accuracy of registration between the subject and the image.

Recently, several research groups combined functional MRI (fMRI) with TMS for localization [Boroojerdi et al., 1999, Sparing et al., 2008, Neggers et al., 2004]. Compared to other non-invasive imaging methods, fMRI can monitor changes in the brain and provide the best quality insight into the regions of the brain influenced by TMS (Figure 2.8). fMRI uses blood oxygenation level dependent (BOLD) so that it has better spatial and temporal resolution than most other non-invasive techniques. Bohning et al.[1998, 1999] initially obtained the BOLD signal during the TMS procedure and found that there are no limitations on the number of scans by fMRI. Previous studies combining fMRI and TMS can achieve precise anatomical location. Spatial correspondence of TMS and fMRI can achieve accuracies of 2.3 ± 0.8 mm, and this method of visualizing the effects of TMS have proven powerful and fruitful [Wagner et al., 2007]. One uncertainty of fMRI-TMS is basic physiology of TMS-induced effects on the BOLD-signal [Siebner et al., 2003].

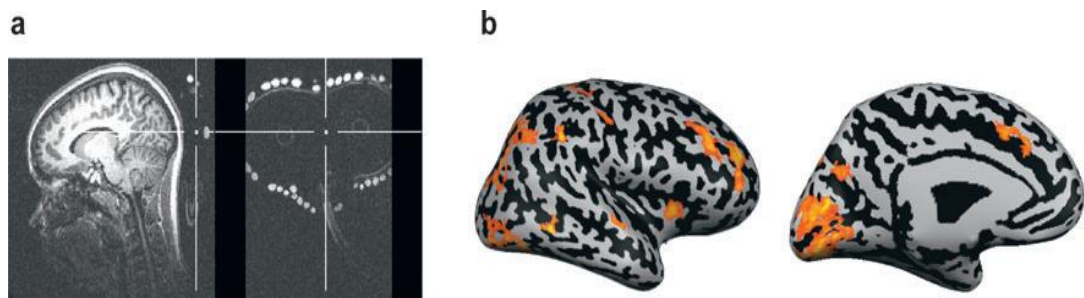


Figure 2.8 Depictions of fMRI and TMS, showing (a) anatomical MRI with coil representation, where the coil is centered above the primary visual cortex and in the inset the TMS coil is highlighted with vitamin E pills, and (b) fMRI BOLD activity evoked from single-pulse TMS on the primary visual cortex [Wagner et al., 2007].

There is a frameless MRI-based neuro-navigation system combined with positron emission tomography (PET) [Wassermann et al., 1996b]. Recently, researchers have combined TMS with other non-invasive methods such as: EEG, EMG and PET for studying human brain activities in order to obtain a more complete understanding of brain function and the relationship between behaviour and the brain. Each method has its unique advantages and provides a different perspective for studying the function of the brain.

2.4.3 The pose of the coil

After a target site for stimulation has been defined, the next problem is what is the best pose to place the coil over the scalp? The “pose” contains two main factors: one is the distance from the coil and cortex, the other is orientation of the coil. Orientation of the coil critically affects the cortical activity produced by TMS (Figure 2.9). The different MEPs amplitudes reflect the intensity and direction of electrical fields induced with varying orientation of the electrical fields [Fox et al., 2004]. Different models of the electrical field have been developed to predict the distribution of the field and the brain response. Tofts [1990] constructed a model for calculating the spatial distribution of induced currents, and found that the nerve fibres parallel to the skull surface can be stimulated more easily than those arranged obliquely. Perpendicular fibres are extremely difficult to stimulate. This study indicated that better efficiency would be achieved in a direction parallel to cortical columns, but normal to sulcal banks. Similar results have also been reported in [Thielscher and Kammer, 2002, Wagner et al., 2004]. Fox et al.[2004] developed a column-base model combined with PET to demonstrate the effect. They reported that the vector of the electrical field induced by TMS should be aligned with cortical columns in order to achieve optimal directional sensitivity. In addition, the effective current field can be calculated as the cosine of the angle between the field direction and that of cortical columns. According to this study, the most efficient orientation would be if the electrical field is parallel to the cortical columns.

Another key factor that affects the activity induced by TMS is the distance from the coil and motor cortex. Kozel et al.[2000] conducted a research on the relationship between the coil–cortex distance and motor threshold. The distance to motor cortex correlated strongly with the motor threshold, and a greater distance to cortex indicated a higher motor threshold. In addition, a significant reduction in effectiveness occurs if the coil loses contact with the head [Leboss éet al., 2006].

In conclusion, the coil has to make contact with the subject’s head and the orientation of the coil has to be precisely controlled in order to induce an electric field that is oriented parallel to the cortical columns.

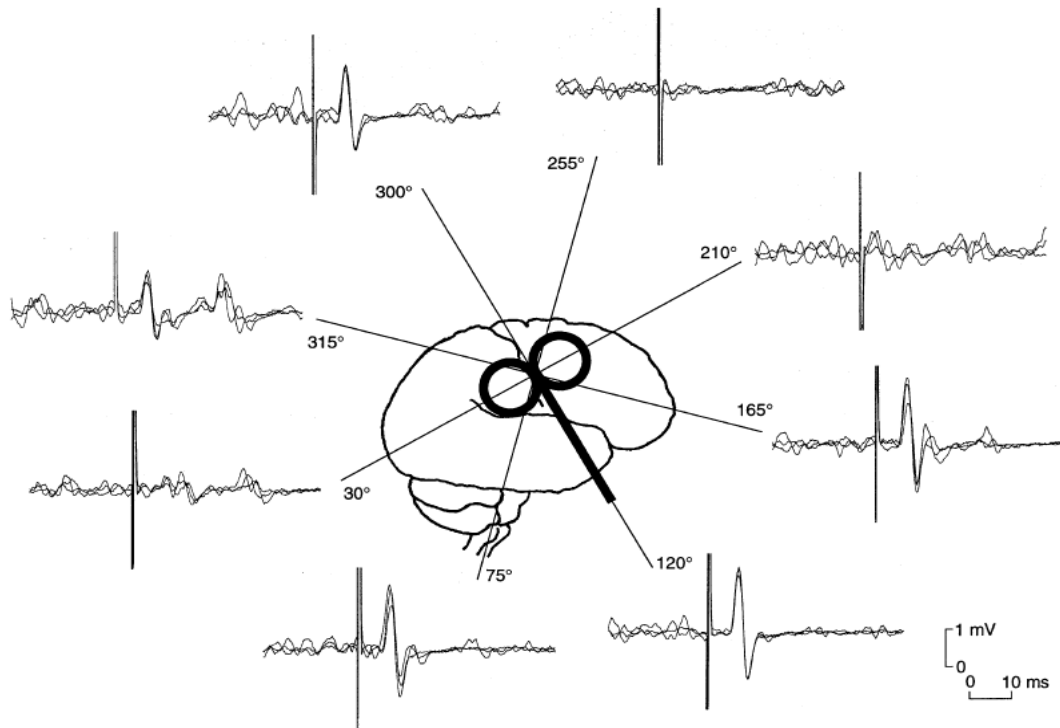


Figure 2.9 MEPs of the left masseter muscle evoked by contralateral TMS in a representative subject. For each of the 8 different coil orientations, 3 MEPs were superimposed. The greatest and most consistent amplitudes were found at 120° and 300° , while clearly smaller MEPs were found at 30° and also between 210° and 255° [Guggisberg et al., 2001].

2.5 Robot assisted surgical system

Robot assisted surgical systems have been under development for many years, although such systems still have not been routinely used in clinical procedures, they can provided some technology and experience in the development of robotic TMS system. Gomes [2011] made a summary of surgical robotics timeline (Figure 2.10).

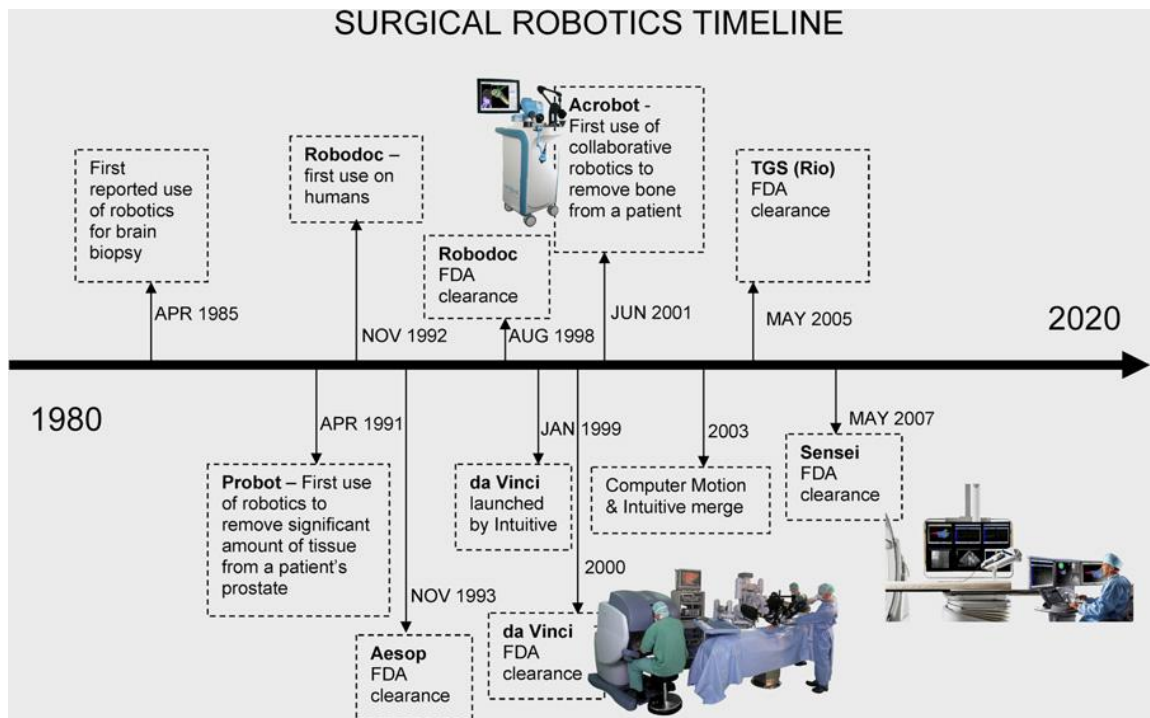


Figure 2.10 Timeline of surgical robots [Gomes, 2011]

The first clinical trial using a robot to perform a stereotactic brain biopsy was in 1985 [Kwoh et al., 1988]. The surgical system combined a CT scanner and a Puma 200 robot for Stereotactic Brain Surgery (Figure 2.11). The robot moved to the target position after the target is identified on the CT image. The surgical procedure was faster than the manually one, and furthermore, the accuracy was improved by proper calibration of the robot. The next milestone appeared six years later, a robotic device called Probot was used to autonomously remove a significant amount of tissue from a patient [Davies et al., 1991]. After this, companies started to join the field of surgical robotics. A brief introduction to several typical surgical robotic systems is given as follows.

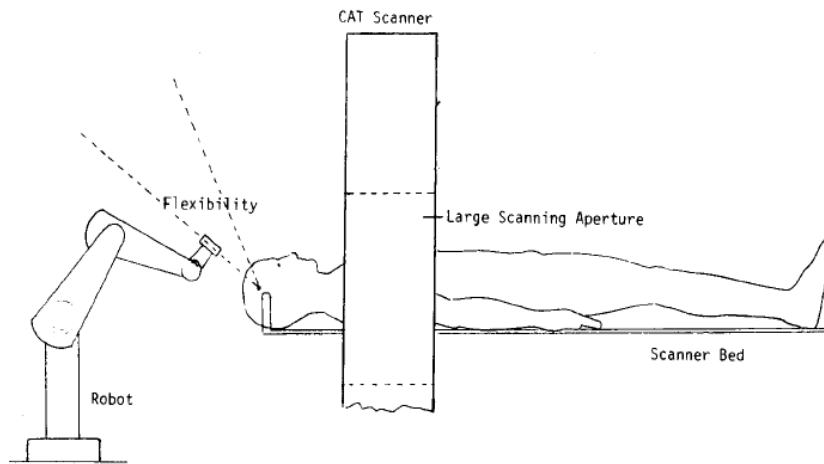


Figure 2.11 Overall organization of the surgical system [Kwoh et al., 1988].

Korb et al. [2004] divided robotic surgical systems into two groups: tele-manipulators and pre-programmed surgical robots. The former consists of a master manipulator and a minimally invasive slave manipulator, where the movement of the slave is copied from the master which is steered by a surgeon. The surgeon can monitor the position of the slave through endoscopic images.



Figure 2.12 Da Vinci robotic system(Courtesy of Intuitive Surgical Inc., Mountain View, CA).

The Da Vinci robotic surgical system (Figure 2.12) is a master/slave system consisting of a master console and a patient-side cart with instrument arms, camera arms and visualization system. The master console consists of an image processing computer; the view port where the surgeon views the image; foot pedals to control electro-cautery, camera focus, camera arm clutches, and master control grips that drive the slave robotic arms at the patient's side [Kim et al., 2002]. It translates the

surgeon's hand, wrist and finger movements at the console instrument controls into corresponding scaled down real time movements of instruments positioned inside the patient. The practice of robotic surgery is currently largely dominated by the Da Vinci system.

Unlike a tele-manipulator system, the motions of the pre-programmed surgical robots are pre-determined, where the robot executes a pre-defined trajectory. The Robodoc system (ROBODOC[®], Curexo Technology Corporation, CA, US), shown in Figure 2.13, was the first system to perform a milling trajectory using a commercial robot assisted surgical system developed by Taylor et al. [1994]. The Robodoc was developed mainly for total hip replacement which involves five main steps during the operation procedure: (1) data acquisition; (2) preoperative planning which consists of cement removal and new implant planning; (3) intra-operative validation and re-planning which is not always necessary but needed for certain cases; (4) pre-operative plans and images should be registered to the robot and the actual patient space; (5) milling the cavity. At any time during the procedure, the surgeon can stop the robot immediately if an unpredictable problem occurs.



Figure 2.13 ROBODOC system (ROBODOC[®], Curexo Technology Corporation, CA, US).

There are many robotic systems being developed for neurosurgery, which is always topical and of current interest. The NeuroMate (CUREXO Technology Corporation, CA, US) is a six-axis robot (Figure 2.14) and is the first neurosurgical robotic system approved by the U.S. Food and Drug Administration [Karas and Chiocca, 2007]. It is a surgical system that has evolved from stereotactic or functional neurosurgical applications. A typical clinical procedure of NeuroMate is similar to ROBODOC, which consists of five main steps: (1) the acquisition of images; (2) pre-planning the path; (3) registration; (4) moving to the target position; (5) the surgical tools are manipulated manually by the surgeon at this fixed position. To complete the procedure, the robot knows the position relative to the subject's anatomy. This is initially achieved by using a calibration cage, which is placed on the end-effector of the robot around the subject's head, later the registration process is completed by using ultrasonic sensor [Varma et al., 2003].

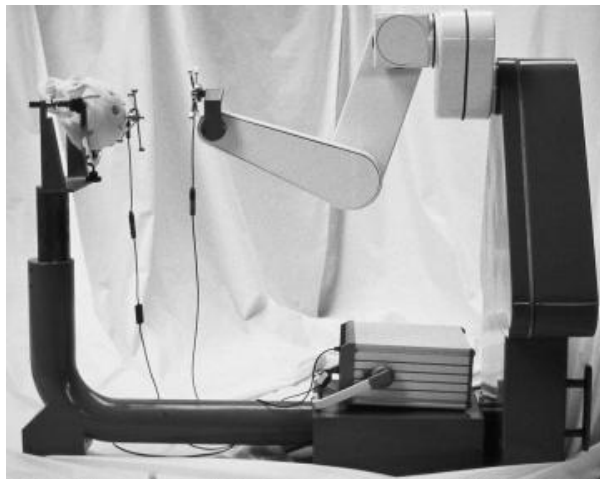


Figure 2.14 NeuroMate (CUREXO Technology Corporation, CA, US).

Pathfinder is another six-axis surgical robot produced by Armstrong Healthcare for frameless stereotaxy [Morgan et al., 2003]. A CCD camera and a tool holder are mounted on its end-effector; the former is used to locate fiducial markers on the patient, the latter for carrying the neurosurgical tools such as drill, needle and so on (Figure 2.15). The typically registration procedure of this system involves the robot moving in a pre-defined circle over the subject's head at a safe distance. The positions of the fiducial markers are located by the camera at the same time. A matching algorithm is used to register the image co-ordinates to the subject after all

the markers have been localized. The overall safety of Pathfinder is impressive as there are duplicate encoder systems. These two encoders systems are separately calculated and encoded, and the system will stop immediately if a mismatch occurs. There are several other safety features such as a foot-pedal which is triggered by the surgeon if the robot is required to move. Safety critical issues are monitored by an embedded processor.

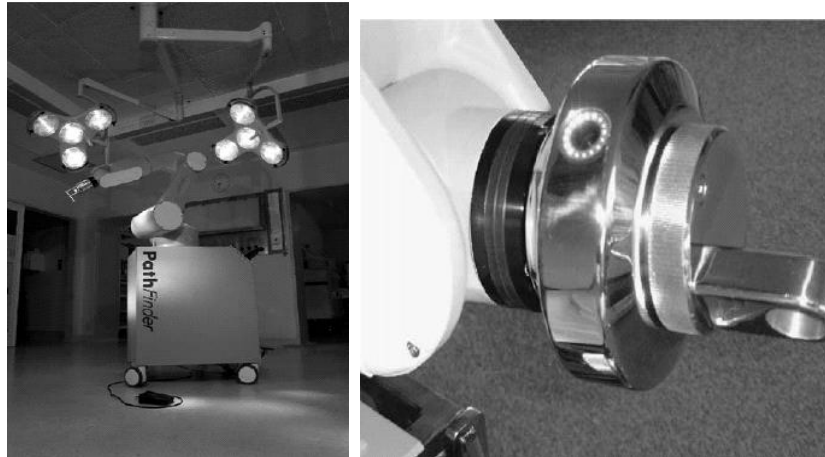


Figure 2.15 The Pathfinder surgical robot (left) and the view of the camera surrounded by the light ring mounted at the end of the arm (right) [Morgan et al., 2003].

Robot assisted radiosurgery has successfully been demonstrated for treatment of brain tumors. During the radiation delivery procedure, precise location of the tumor relative to the reference coordinate of the treatment device is essential to prevent irradiation of normal tissue. Stereotactic frames are used to locate the tumor. Adler Jr et al. [1999] introduced image-guided radiosurgery for accurate irradiation, called CyberKnife Stereotactic Robotic-radiosurgery System (courtesy of Accuray, Sunnyvale, CA) which consists of three main components: a radiation source mounted on a Kuka robot, a stereo x-ray imager and a treatment couch (Figure 2.16). Stereo x-ray imaging is employed by the system to regularly track the positions of landmarks with respect to the treatment instrument. The information is sent to the robot which is used for adjusting the focus point of the linear accelerator beam towards the tumor.

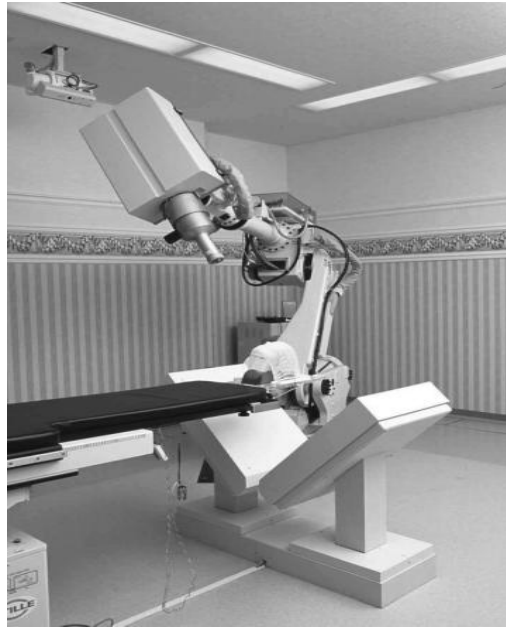


Figure 2.16 CyberKnife robotic radiosurgery system (courtesy of Accuray, Sunnyvale, CA).

A concept of a versatile interactive surgical assistance system has been introduced by Wahrburg et al. [2004]. The system combines an optical position tracking system and a manipulator. Automatic tracking of movements of the subject is carried out by the tracking system and this information is transferred to the robot to follow the movement. A key innovation of this system is the development of a versatile haptic interface which is used to support its unique operating mode. It has a force torque sensor mounted at the wrist of the manipulator, and the surgeon can move the manipulator to the desired position by gripping the handle. In this way, the system can then be operated in autonomous mode by pre-planning the motion. The surgeon can also control the arm directly depending upon the surgical requirements. Figure 2.17 shows the block diagram control loop of the system.

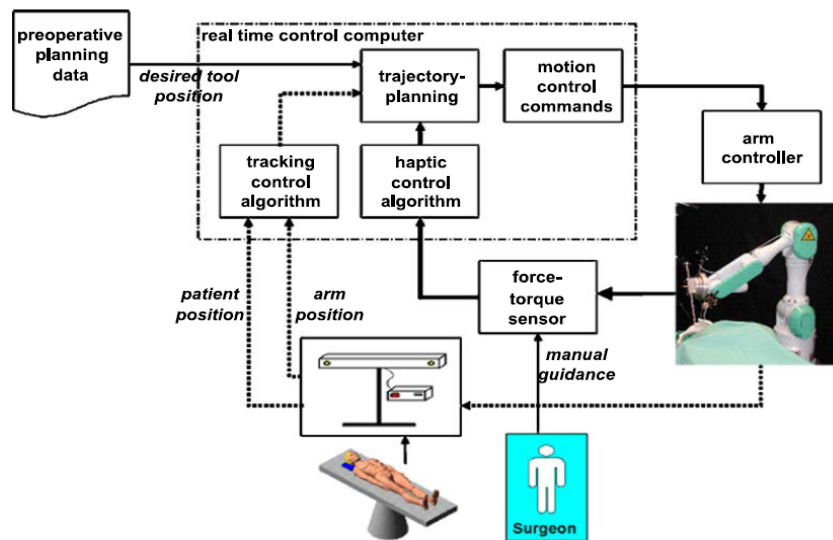


Figure 2.17 Close loop control of the versatile interactive surgical assistance system [Wahrburg et al., 2004].

2.6 Existing robot assisted TMS systems

When the coil is positioned manually, the exact stimulation at a pre-defined site is difficult to achieve, which is a major issue of existing TMS technology. It is primarily for this reason why a robotic TMS system needs to be developed which can provide accurate locating and holding capabilities, as well as tracking the head and compensating for its movements.

A basic robotic TMS system was initially constructed by Lancaster, J.L., et al. [2004] (Figure 2.18). It is an image-based robotically positioned TMS system that integrates cortical-column aiming theory [Fox et al., 2004] and exceptional planning and delivery capabilities of the irTMS system. The initial results demonstrate the achievable high level of accuracy and systematic TMS coil positioning using a robot arm. There are several important steps of the robotic system, which are as follows:

- Determining target sites and planning the TMS coil pose. Target sites are selected through MR images with overlaid co-registered statistical parametric functional images. A preferable coil pose is the one with the treatment axis of the coil passing through the targeted site, scalp-to-target distance is minimized, and orientation about the treatment axis is parallel to the cortical-column so that TMS can obtain the best stimulation efficiency.

- Registration. There are two registration procedures in this system. Firstly, registration of the head to the MR Image in order to find a relative stimulation site on the scalp for each target site in the MR Image. Second, registering the robot to the subject's head. The purpose of this is to transform the robot coordinates to the head coordinate system. A mechanical 3-D digitizer (Microscribe 3DLX, Immersion Corp., San Jose, CA) was used in this study to acquire the head surface model and record the landmarks on the head.
- Positioning the TMS coil by the robot. In this study, the position process is not fully automatic, and there is no real-time tracking scheme. In order to avoid collision, the method is to manually move the coil to a safe pose, then to use automated positioning software to move the coil in small steps to the final position.

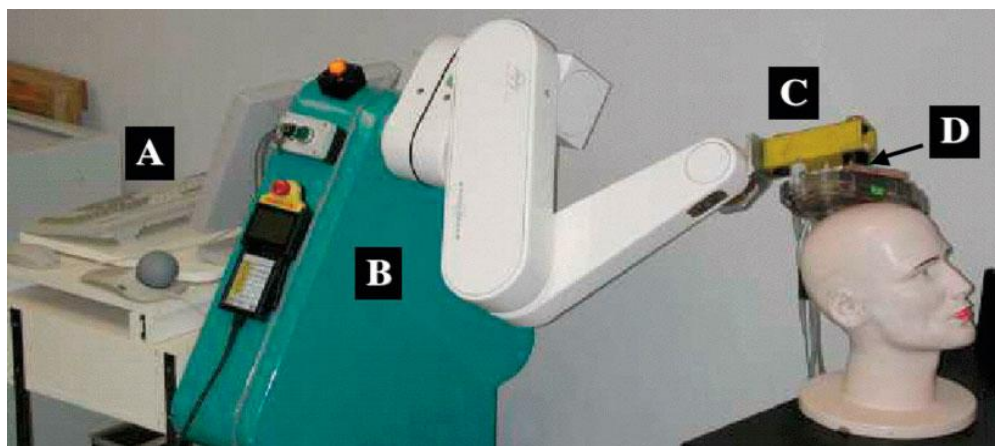


Figure 2.18 The irTMS robotic system with (A) the computer for processing and controlling planned positioning of the TMS coil, (B) the Neuromate robot arm, (C) the added TMS coil holder, and (D) the 6th axis for coil rotation [Lancaster et al., 2004].

Another robotized TMS system was presented by a Germany group [Matthaus et al., 2005, Matthaus et al., 2006, Finke et al., 2008]. This system comprises six main components (Figure 2.19). The procedure is similar to the Lancaster system, registration of the cranium to its 3D MR imaging (MRI) data is carried out and the actual position of the head is continuously checked using an infrared optical tracking system (Polaris, NDI, Waterloo, Canada). The tracking system is also used to guide the robot-held coil to its target point. During stimulation, constant head tracking and

subsequent refinement of the robot position can be performed so that the coil will be maintained at the same position relative to the cranium (Figure 2.19).

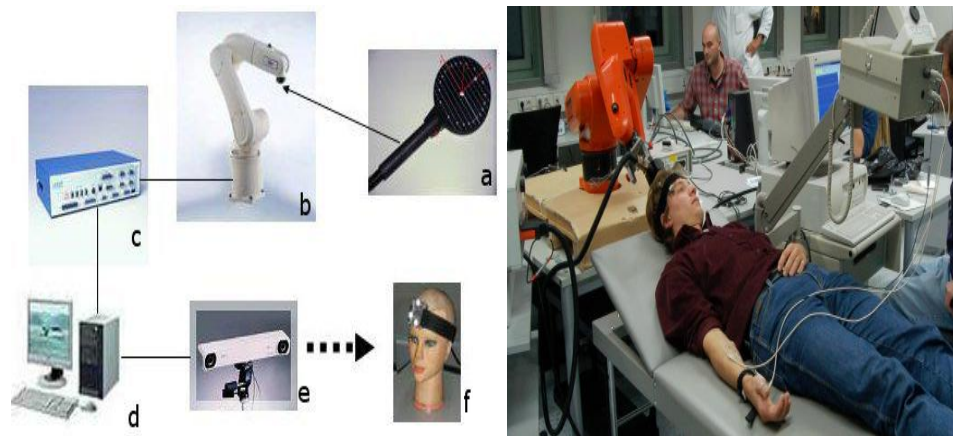


Figure 2.19 The six main components for robot-guided TMS: (a) TMS coil, (b) robot, (c) robot controller, (d) computer, (e) Polaris tracking system, (f) headband [Finke et al., 2008](left) and Application setup[Matthaus et al., 2006](right).

This system introduced a real-time tracking system to the application. This eliminates the need for head-fixation. After the target stimulation point has been chosen, the robot will position the coil to the target point base on the coordinate of the subject's head. However, the accuracy of positioning and the performance of real time compensation in this system have not been presented. Furthermore, there is no measurement of the contact force, accessory to maintain the coil contact with the subject's head. A recent paper indicates the motion compensation latency is 300ms in total, and it may vary due to the motion speed [Richter L. et al., 2010].

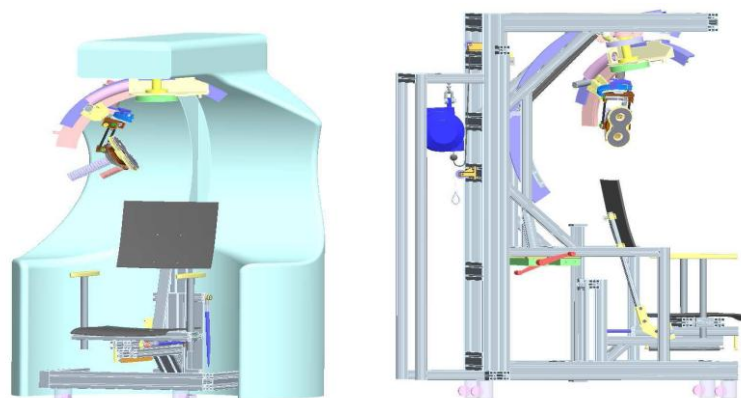


Figure 2.20 The CAD model of the robot [Lebosse et al., 2007].

A novel robotic TMS system was proposed by Lebosse, C., et al. [2007], unlike the industrial robot based system above, a sophisticated dedicated robot concept was presented (Figure 2.20). The proposed robot is a redundant serial structure with seven active degrees-of-freedom (DOF). It is consisted of three subsystems: The first is a spherical mechanism, with a 3-DOF serial structure (J1-J2-J3 in Figure 2.21). It is possible for the positioning of the coil centre around a sphere centre on the head of the subject due to the spherical shape of the workspace. The second subsystem is simply composed of an actuated prismatic joint (J4 in Figure 2.21). A compliant element is placed on the joint axis between the actuator and the coil so that the contact force can be controlled passively. Finally, the third part consists of a spherical serial wrist (J5-J6-J7 in Figure 2.21). In order to maintain the coil tangent to the head surface during the stimulation, it allows rotation around a fixed point which is the contact point between the coil and the head.

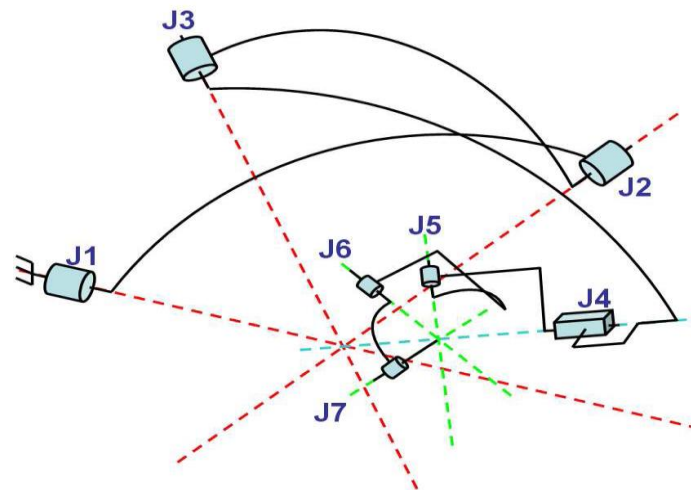


Figure 2.21 The kinematics scheme of the robot [Lebosse et al., 2007].

Although this proposed system is a non-industrial robot system, the workflow of the system is generally the same as the others. However, there are some additional tasks because more functions have been integrated. According to [Lebossé et al., 2006], the task sequence execute is as follows:

- Different MRI and fMRI images are recorded which are used to build a 3D model of the subject's head and brain. The target regions are then specified, and the frequency and duration of the stimulation period defined.
- The neuro-navigation system computes the trajectory of the coil centre on the head that best stimulates each region. A registration of the subject's head with respect to the robot and the 3D model is carried out.
- The stimulation procedure is executed autonomously accommodating head movements. In the mean time, the actual location and orientation of the coil are recorded for a post-procedure analysis.

This study proposed a concept using a novel purpose built robot, however, the performance of the system is still an open question.

2.7 Relevant technologies

Several relevant technologies that are important in developing a robotic TMS system have been identified from the previous sections, these are: (1) tracking technology; (2) visual servo control; (3) registration; (4) trajectory planning and (5) position and force control in robot manipulators. These are discussed in the following sections.

2.7.1 Tracking technologies

In order to avoid fixation of the subject's head, a tracking system should be used for guiding and subsequent refinement of the coil position and orientation. There are many types of tracking system available such as: spatial linkage systems, ultrasonic tracking system, magnetic field tracking system and optical tracking system [Glossop, 2009].

Spatial linkage systems are current used mainly in neurosurgery. Krings et al. [2001, 1997] initially introduced this system in TMS to determine the stimulation site. A linkage system normally has a multi-jointed mechanical arm with optical encoders in each joint. The encoders are used to measure the positions of the joints and this information can be used to compute the position and orientation of the device mounted on the end of the mechanical arm. However, this kind of system can only precisely fix the device mounted on the end of the arm but it is difficult to use for monitoring the motion of the subject's head.

Ultrasonic Tracking Systems are also used in many surgical procedures. An ultrasonic system normally has three microphones fixed on a rigid frame, which can receive the high frequency sound mounted on the device. The system can use the information to triangulate the position of the device. The accuracy of tracking systems based on ultrasound is approximate 2–5 mm [Kalfas et al., 1995]. However, the accuracy of this technology can be affected by ambient sounds and temperature in the surrounding environment. These systems are rarely used in current surgical or other types of navigation systems, since the deviations cannot be established precisely.

Magnetic field tracking systems [Manwaring et al., 1994, Birkfellner et al., 1998], have a clinical precision of approximately 2–4 mm. This type of system has a magnetic field generator, which induces voltages in the sensors attached to the device or subject. The position and orientation of the object can be calculated from these induced voltages. A commercial Magnetic field tracking system is the PATRIOT™ Digitizer (Polhemus, Michigan, US), which has been employed in TMS studies [Noirhomme et al., 2004, Bastings et al., 1998]. Another commercial system is the Aurora System (NDI, Ontario, CA) that is able to track the smallest sensor coils on the market. This technique is also widely used in Virtual Reality; and because sales outside the medical field are large, some components of this technique are cheap. A significant disadvantage of this type of tracking system is electromagnetic and metal interference although there have been some improvements made in recent years [Birkfellner et al., 1998, Marmulla et al., 1997].

Optical tracking systems consist of a position sensor and markers, of which there are two types, passive and active. Position sensors measure the 3D positions of both active and passive markers affixed to application-specific tools (Figure 2. 22). Active markers use infrared light emitting diodes which the position sensor detects and calculates position. The former passive markers are spheres which have a retro-reflective coating that reflects infrared light illumination directly back to the position sensor. Commercial optical tracking systems typically use either two cameras (i.e. Polaris, NDI, Ontario, CA.) or three cameras (i.e. Flashpoint, BIG Inc., CO, USA), and they can track a number of marks simultaneously. Both the Polaris and

Flashpoint systems have been employed in TMS stimulation procedure [Ettinger et al., 1998, Matthaus et al., 2006]. Khadem et al. [2000] conducted an error analysis of the Polaris and Flashpoint systems and found that both of them have jitter of less than 0.5 mm, and is indicative of the accuracy of optical tracking systems. Several surgical studies have been conducted using active markers with an accuracy of about 1 mm [Reinhardt et al., 1996, Marmulla et al., 1997]. A surgical planning and guidance system based on passive markers was developed by Colchester et al.[1996]. Video cameras were used to monitor the positions and movements of these markers, and image analysis software used for registration. The accuracy of the system is typically less than 1 mm with an intra-operative precision of approximately 2.5 mm.

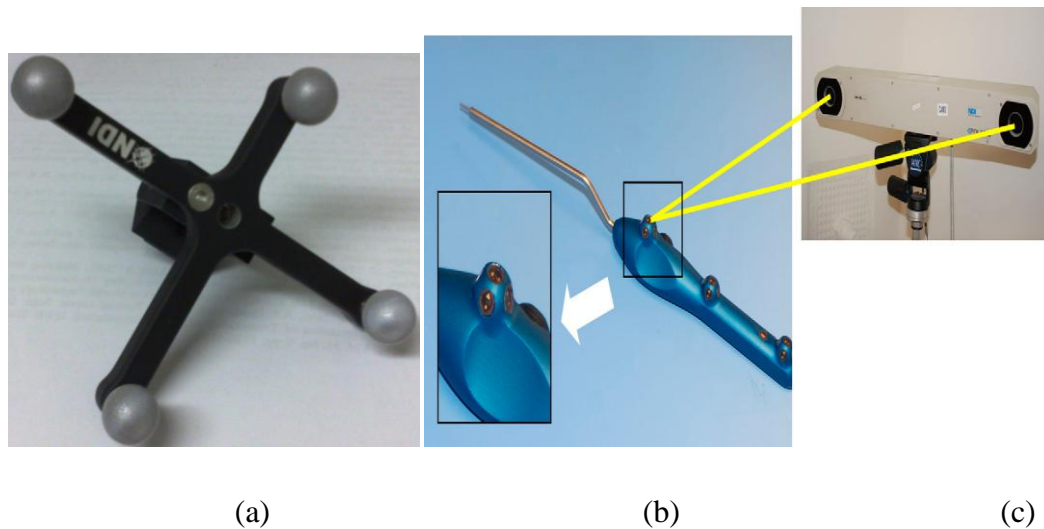


Figure 2. 22 Polaris Spectra position sensor (c), the active markers (b) and the passive markers (a).

Optical tracking and magnetic field tracking systems are more reliable and accurate. The disadvantage of optical tracking is that it requires a direct line-of-sight from cameras to targets. Therefore, compared with disadvantages of magnetic field tracking system, utilizing an optical tracking system in TMS system would be a better choice. In addition, such systems are also used to acquire the coordinate system of the scalp and coil in physical space, which acts as an intermediate of the registration system in image-guide TMS [Matthaus et al., 2005, Herwig et al., 2001b, Noirhomme et al., 2004], which is presented in following section.

2.7.2 Visual servo control

The control concept of visual servo control can be utilized in this application. Instead of using a tracking system for collecting feedback data, visual servo control using visual information that may be acquired by camera to control the motion of a manipulator or a mobile robot. The visual information includes points, lines and other features in the environment. There are several camera-robot configurations used in visual servo systems as illustrated in Figure 2.23.

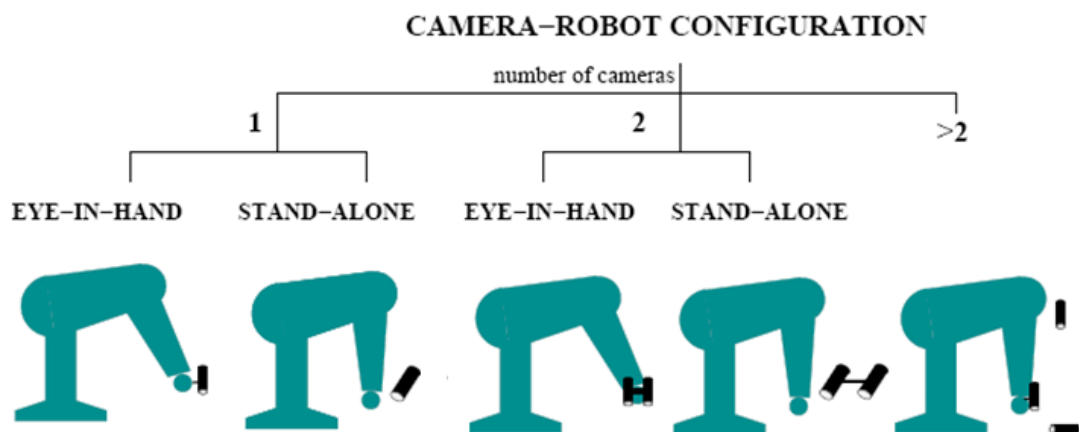


Figure 2.23 Camera-robot configuration used in visual servo control system [Kragic, 2001]

A typical visual servo control task is to continuously measure the error between the current location of the robot arm and its destination using the visual sensor. A control sequence is generated to guide the motion of the robot base on this input. According to the different application of the visual data, the system can be divided into three main types: position based, image based and 2 1/2D visual servo systems. Position based visual servo systems adapt 3D data from the scene in order to estimate the pose of the target with respect to relative coordinate systems. Image based visual servo systems use 2D images to reduce the image distance error by comparing the features of the images taken with different distances [Sjö et al., 2008]. The 2 1/2D system is a combination of the previous two methods [Kragic, 2001]. The position based system is very similar to our system requirement, where the task is represented by an error function between the current and the desired position and orientation of the robot. The motion of the robot is then generated based on this

function. There are two fundamental control structures for a position based visual servo system [Sanderson and Weiss, 1980, Hutchinson et al., 1996].

- Dynamic position based look-and-move structure (Figure 2.24). The characteristic of these systems is the visual sensor provides the information to the robot controller and the feedback from joint sensors provide the data to internally stabilize the robot [Hutchinson et al., 1996].

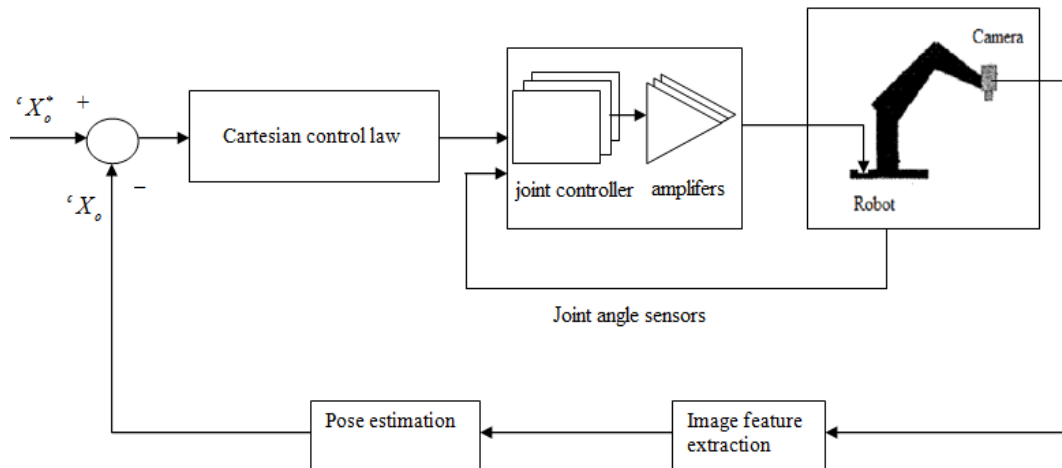


Figure 2.24 Dynamic position based look and move structure. ${}^c X_o$ is the current pose of the camera with respect to the object, ${}^c X_o^*$ is the desired pose.

- Position based visual servo (PBVS) structure (Figure 2.25). There is no joint controller in these systems, and the visual system directly provides the data to control the robot joints.

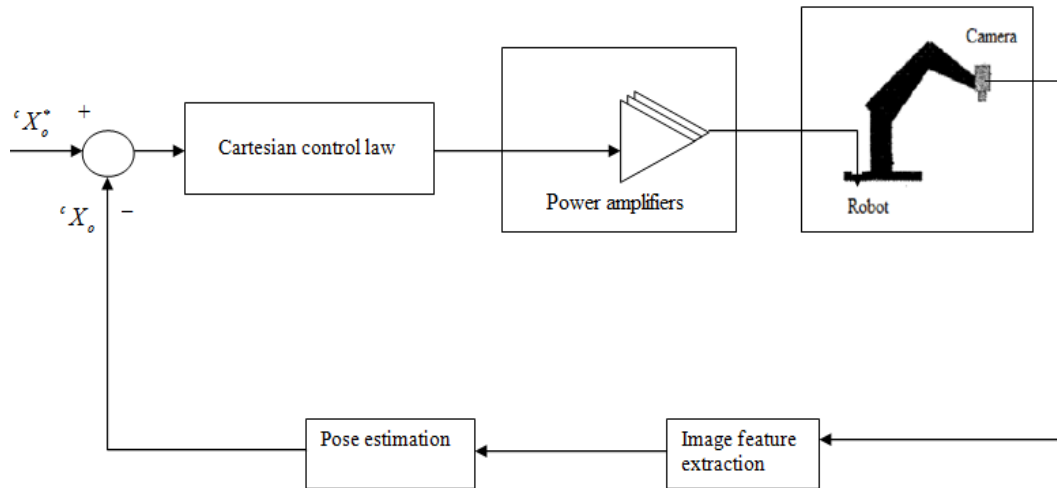


Figure 2.25 Position based visual servo (PBVS) structure

2.7.3 Coordinate systems registration

Coordinate system registration is a key step in both hand-held TMS and robotic TMS procedures. Most studies on registration are based on hand-held image-guided TMS. There have been numerous studies on registration of TMS with 3D MR images [Ettinger et al., 1996, Noirhomme et al., 2004, Neggers et al., 2004]. These studies were concerned with registering the subject's head coordinate system with respect to the MRI coordinate system. Several research groups rely on point to point correspondence of several anatomical landmarks or fiducial markers. Digitizer technologies based on optical, magnetic-field, or stereotactic frames are used for acquiring the coordinates of the markers from the physical head surface. These points are then registered to the segmented MRI scalp. Bastings, et al. [1998] applied such a method, using a magnetic-field (MF) digitizer to locate six MRI-visible labels on the subject's scalp. Surface-based methods have also been developed, which acquire hundreds of points on the surface, and the segmented MRI scalp surface is then registered with these points. Wang et al. [1994] acquired four hundred points, whereas two thousand points were acquired by Schwartz et al. [1996]. The more points that are acquired, the better the accuracy obtained. Matthaus, et al. [2005] conducted a combined landmark and surface-based method, which used a Polaris optical tracking system to acquire several hundred surface points and three landmarks attached on the surface for registration (Figure 2.26).

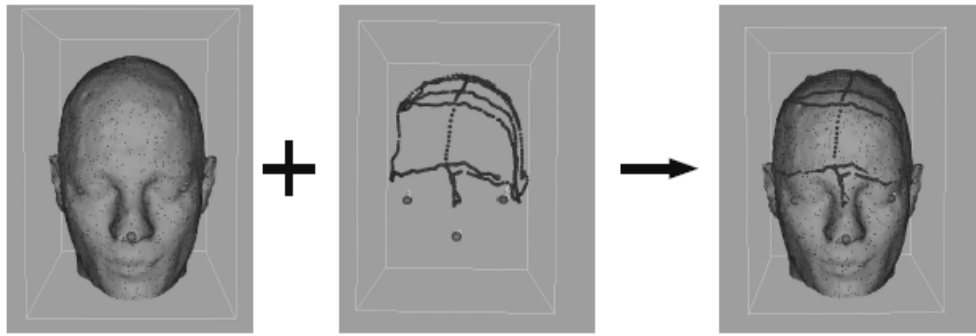


Figure 2.26 Registering the virtual head to the real head using landmarks and surface points. **Left:** Surface points and landmarks from the real head. **Middle:** Virtual head model. **Right:** VR model registered to the real head [Matthaus et al., 2005].

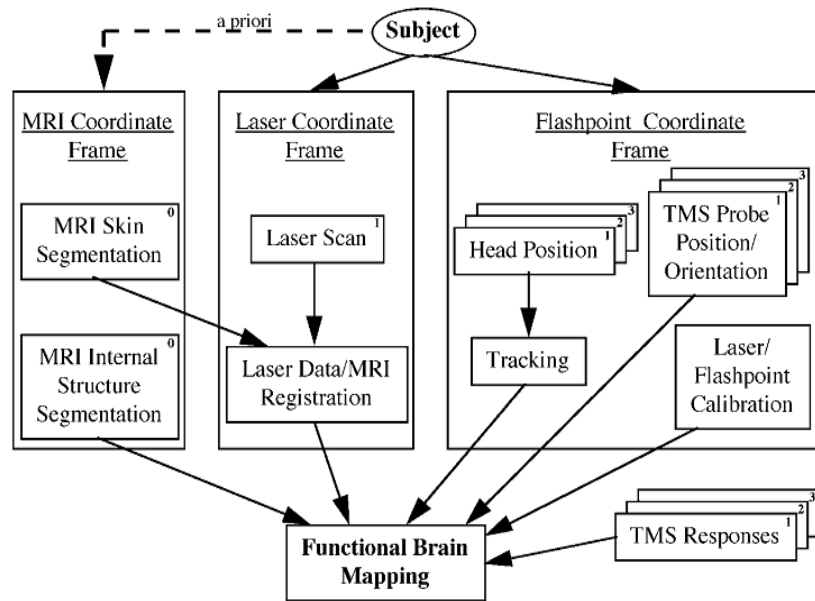


Figure 2.27 Architecture of the functional brain mapping system [Ettinger et al., 1998]

According to these studies, accurate head surface data acquired by a magnetic tracking system or optical tracking system will significantly influence the accuracy of registration. Ettinger, et al. [1998] developed a method using a laser to acquire the head contours in order to achieve a better accuracy. In this way, three coordinate frames were defined in their system: MRI coordinate frame, Laser coordinate frame, Flashpoint coordinate frame (Figure 2.27). A typical method of coordinate registration have been presented by Ettinger et al.[1996], in which a laser was used to digitize the full head surface in physical space (Figure 2.28). Flashpoint is a self-contained 3-D tracking system (BIG Inc., CO, USA) consisting of three linear

cameras which localize flashing IR LEDs. The system can track a number of LEDs simultaneously to capture the motion of the head and the coil. By fixing at least three LEDs to the head, the tracking system calculates the actual head position in space. Two LEDs are fixed to a reference frame on the magnetic coil in order to detect its position using the same cameras. Accordingly, the head's surface coordinates and the coil coordinates are registered to a single common reference system.



Figure 2.28 Laser scan of a subject's head. Collected laser points are shown as white curves [Ettinger et al., 1998].

The significant difference between Ettinger's system and other systems is the use of an additional laser scan system to acquire the physical head surface instead of using the tracking system (Flashpoint in Ettinger's system). Therefore, a registration of the Laser coordinate system and the Flashpoint coordinate system should be carried out first. An off-line calibration is required to achieve the Flashpoint to laser transform by mounting the laser scanner and Flashpoint relative to each other on a rigid bar which is attached to a movable arm. This information is then used to register the laser to the MRI coordinate system. In this process, the laser coordinates serve as an interface for the MRI and Flashpoint coordinate systems. In the other systems, the tracking system fulfils the role of the laser scanner and Flashpoint.

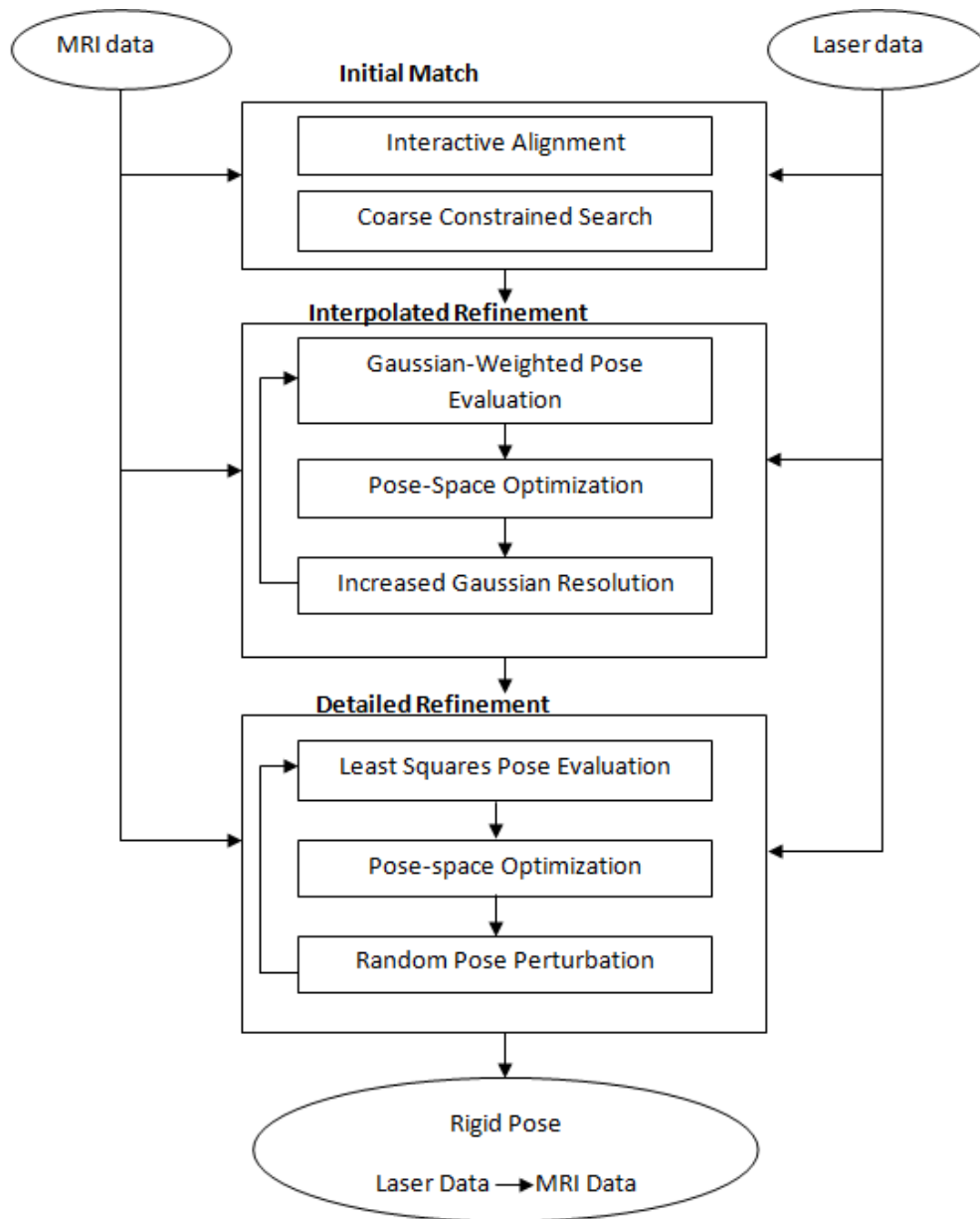


Figure 2.29 Registration process[Ettinger et al., 1998].

The key step is the registration of the MRI data with the actual subject, i.e., to align the head surface of segmented MRI data to that obtained with the laser scanner or a tracking system. Ettinger et al. [1996] divided the registration process into three main steps: initial match, interpolated refinement and detailed refinement (Figure 2.29).

The first step is to roughly align the laser data relative to the MRI data. The accuracy requirements for the initial alignment are a function of the data coverage of the laser

data [Ettinger, 1997]. An alternative method is to select pairs of points or triple points from the laser data, then roughly register them with possible points from MRI data [Grimson et al., 1996]. After this, an interpolated refinement should be carried out by minimizing an evaluation function that measures the amount of mismatch between the two data sets. Gaussian distribution [Wells, 1997] is used to weight the distance from the transformed laser points to the nearest MRI points. The evaluation function for a particular transformation is given by:

$$E_1(T) = -\sum_i \sum_j e^{-\frac{|T\ell_i - m_j|^2}{2\sigma^2}} \quad (2.1)$$

where vector ℓ_i is a laser point, vector m_j is an MRI point, σ is the standard deviation and T is the desired coordinate frame transformation. The Davidon-Fletcher-Powell algorithm (DFP) [Press, 2007] is used for minimizing this evaluation function. Finally, a rectified least-squares distance measure is used to achieve an optimal solution. The function becomes:

$$E_2(T) = \left[\frac{1}{n} \sum_i \min[d_{\max}^2, \min_j |T\ell_i - m_j|^2] \right] \quad (2.2)$$

where d_{\max} is a predefined maximum distance used to limit the effect of outliers and n is the number of points. This second objective function is more accurate, and the final values of the $E_2(T)$ error function are in the range 1.5–2.5mm.

So far, complete steps for registration of MRI data and physical head have been described. There are several studies that use Pelizzari's algorithm [Pelizzari et al., 1989], but the principle is similar [Wang et al., 1994, Noirhomme et al., 2004]. Previous studies on image-guided TMS registration are based on a manual coil placement TMS system, and did not involve robotic coordinates systems. In this study we will develop a robotic system, and the registration between the robot coordinates systems and physical head coordinate system need to be addressed.

Because the subject's head data is usually acquired using a 3D tracking system, in order to register the robot coordinate system to the subject's physical coordinate system. It is required to combine the coordinates of the tracking system with the robot coordinate system. To do this, using an optical 3D digitizer for example, rigid bodies with three or more markers need to be attached to the tip of the end-effector and the subject's head (Figure 2.30). The head and tool are in the same coordinate system since they will both be tracked by the optical tracking system. The main challenge here is to register the robot coordinate system to the end-effector coordinate system.

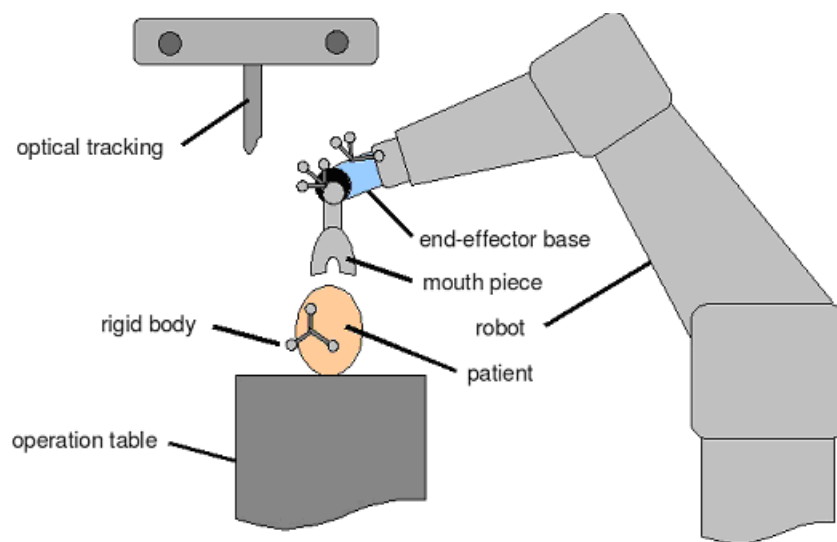


Figure 2.30 An optical tracking system is localizing the position of three rigid bodies: one attached to the bone of the patient and two attached to the end-effector parts for double check on the position of the end-effector[Burgner et al., 2008].

Several similar techniques have been adopted, in which the tool coordinate system is aligned with its origin in the flange of the robot arm tool contact point (TCP), and is selected for the transform between the robot coordinates and the coordinates of the tracking system [Wahrburg and Kerschbaumer, 1998, Knappe et al., 2003, Burgner et al., 2008]. Wahrburg et al. [1998] proposed a concept of using a robot to assist surgery on hip replacement. In this study, a homogeneous transformation matrix is used to describe the transformation between the coordinate of the tracking system and one of the robot coordinate systems (see Eq. 2.3).

$$P^{rob} = H_{dig}^{rob} P^{dig} \quad (2.3)$$

where H_{dig}^{rob} is the transformation matrix, P^{dig} is a point in the coordinate system of the tracking system, and P^{rob} is the point in the robot coordinate system. In order to obtain the relationship between the hip and the robot, another transformation matrix H_{hip}^{dig} establishes the position and orientation of hip coordinate system with reference to the digitizer system (Figure 2.31). The relationship between the hip and the robot can be defined by equation 2.4.

$$H_{hip}^{rob} = H_{dig}^{rob} H_{hip}^{dig} \quad (2.4)$$

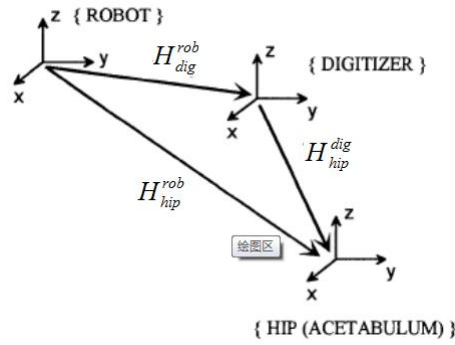


Figure 2.31 Transformation of coordinates [Wahrburg and Kerschbaumer, 1998].

A similar registration equation has also been presented in a surgical navigation robot system [Knappe et al., 2003]. This system aims to register the target site in CT-image coordinates to the robot coordinate system as shown below:

$$H_{Tool}^{Cam} = H_{Rob}^{Cam} \cdot \underbrace{H_{TCP}^{Rob}}_{\text{Robot Calibration}} \cdot \underbrace{H_{Tool}^{TCP}}_{\text{Tool-Geometry}} \quad (2.5)$$

$$H_{Tar}^{Cam} = H_{Pat}^{Cam} \cdot \underbrace{H_{Im g}^{Pat}}_{\text{Matching}} \cdot \underbrace{H_{Tar}^{Im g}}_{\text{Planning}} \quad (2.6)$$

$$H_{Tar}^{TCP} = H_{Tool}^{TCP} \cdot H_{Pat}^{Tool} \cdot H_{Im g}^{Pat} \cdot H_{Tar}^{Cam} \quad (2.7)$$

where H_{Tar}^{Cam} is the transformation matrix which indicates the *Tar* coordinate system with respect to the *Cam* coordinate system, H_{Tool}^{Cam} is the transformation matrix that indicates the *Tool* coordinate system with reference to the *Cam* coordinate system. H_{Tar}^{TCP} is the required transformation matrix that indicates the *Tar* coordinate system with reference to the *TCP* coordinate system (Figure 2.32).

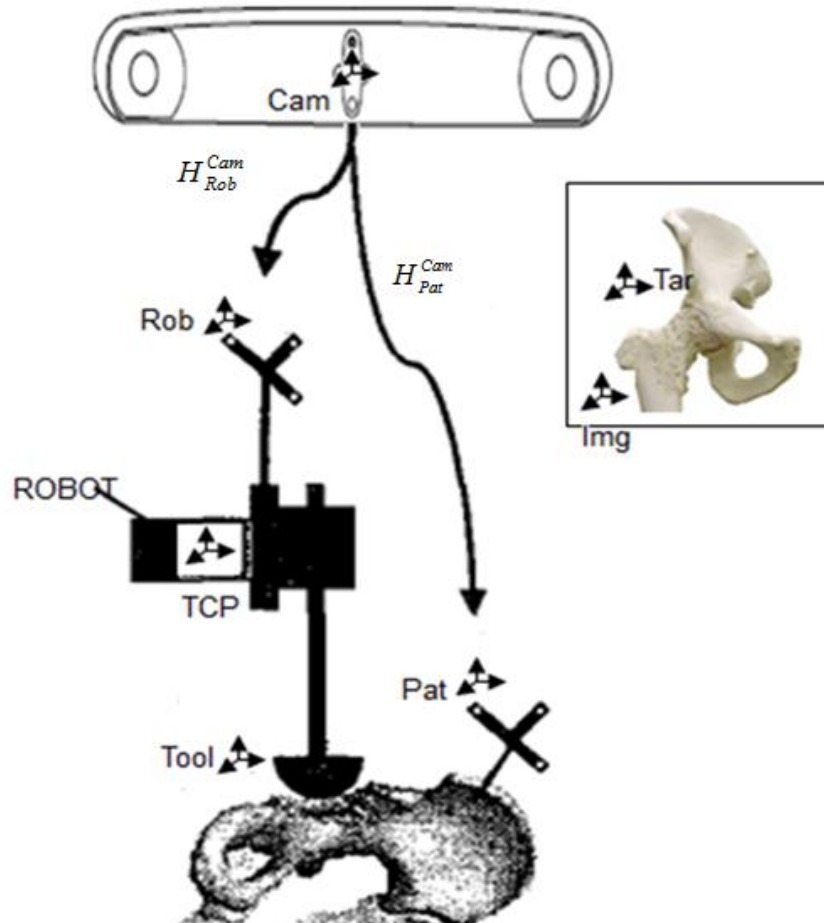


Figure 2.32 Coordinate systems [Knappe et al., 2003].

2.7.4 Trajectory planning

Trajectory planning is a big challenge in a robotic TMS system, because of the medical and safety requirements. A key issue is to regulate the contact force between the coil and the subject's head, which induces an element of risk to the subject. Furthermore, on the premise of safety, the coil/robot trajectory needs to be precisely planned and continuously adjusted under the condition of unpredictable movements

of the subject's head in order to obtain the most efficient and desirable stimulation. The application of robots in TMS is uncommon at present and as such there are no studies on methods of trajectory planning for these type systems. Several methods used in surgical robots are listed as follows.

Most surgical robots are used for simple trajectory execution. The main applications are: milling of the femoral cavities in orthopaedics [Taylor et al., 1994]; serving as tool-holding device in neurosurgery [Morgan et al., 2003] and acting as the 'third hand' for moving the camera in endoscopic surgeries [Finlay and Ornstein, 1995]. Two principal solutions to trajectory generation can be identified in robot assisted surgical system [Wahrburg and Kerschbaumer, 1998].

- A tool similar to that used manually by a surgeon is mounted on the robot. The tool approaches a pre-defined position at a desired orientation adjacent to the operating area. From this location, it can be moved to the target position manually by the surgeon at the correct orientation. The robot then remains in a locked and unpowered position and the surgical procedure is carried out by the surgeon, which makes the operation safer.

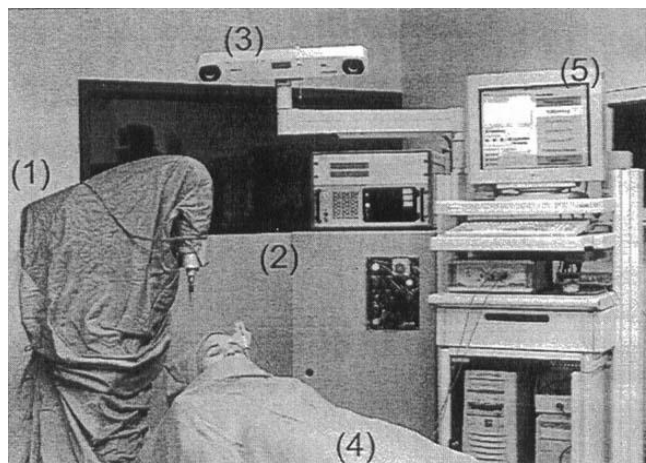


Figure 2.33 The RobaCKa system, consisting of a CASPAR robot (1), the robot controller (2), a Polaris infrared navigation system (3), the subject (4), a "sensor PC unit (5) [Korb et al., 2003].

- A second solution is the robot performs a full automatic procedure with a robot fitted tool. The tool is guided by the robot to carry out the operation. Proper motion planning software has to be integrated into the system to ensure the robot moves along the desired trajectory.

Korb, et al. [2003] developed a surgical robot system that allows precise execution of surgical interventions and milling of complex trajectories. This system consists of three main components: a Caspar robot system, a Polaris system, and a force-torque sensor (Figure 2.33). This trajectory planning method is derived from [Engel et al., 2003], and the algorithm used to maintain the robot configurations close to each joint mid-position and determines the motion based on the following criteria: “(1) reachability of the trajectory interpolation points; (2) not close to singularities; (3) no collision between robot and patient; (4) consideration of the operative approach and the patient’s position on the operating table” [Korb et al., 2003]. The planned trajectory is executed with an error of 0.66mm. However, in this system, the subject is fixed. The Polaris system is used for detecting the movement of the head, and the robot is stopped immediately once movement has occurred.

2.7.5 Position and force control of robot manipulators

In this application, the coil is required to make contact with the subject’s head, thus it is necessary to control not only the position but also the interaction force between the stimulus coil and the subject’s head. Position control is the main mode of control for robot manipulators; and there are two major methods for robot force control: hybrid position/force control [Raibert and Craig, 1981] and impedance control [Hogan, 1985]. Most of the position and force control methods are developed based on these two control structures [Wongratanaphisan and Cole, 2009, Hogan, 1987, Anderson and Spong, 1988, Yoshikawa, 1987, Ghosh et al., 2000, Kumar et al., 2011]. Impedance control controls the position and force by adjusting the mechanical impedance of the end-effector to the external force when the robot end-effector contacts with the environment. It controls the force without the explicit closure of a force feedback loop. Hybrid control is normally employed in an application to achieve more precise control [Yoshikawa, 2000]. The hybrid position and force control approach decomposes the position control in the position subspace, and force control in force subspace. There are usually two feedback loops for position and force. The hybrid controller combines them into a single command vector. Corrections based on these measurements are applied by joint actuators to make the

manipulator track the desired position and force trajectories. A two-link planar robot arm is used as an example to demonstrate the hybrid controller (Figure 2.34); and the task is to push the endpoint of the manipulator with a specified force against the flat smooth surface of the shaded object while moving the endpoint along the surface. In the constraint frame $O_c X_c Y_c$, the force is denoted as ${}^c f(t)$ and the position is denoted as ${}^c p(t)$. Thus the force and position errors in their controlled direction can be expressed as:

$${}^c p_e(t) = (I - S)({}^c p_d(t) - {}^c p(t)) \quad (2.8)$$

$${}^c f_e(t) = S({}^c f_d(t) - {}^c f(t)) \quad (2.9)$$

where ${}^c f_d(t)$ is the desired force, ${}^c p_d(t)$ is the desired position, S is the selection matrix for force controlled directions and I is the identity matrix. Letting α and β denote the position and force compensation functions respectively the hybrid control law in joint coordinate space is then given by

$$\tau(t) = \tau_p(t) + \tau_f(t) = \alpha {}^c p_e(t) + \beta {}^c f_e(t) \quad (2.10)$$

where $\tau_p(t)$ is to compensate for the position error and $\tau_f(t)$ is to compensate for the force error.

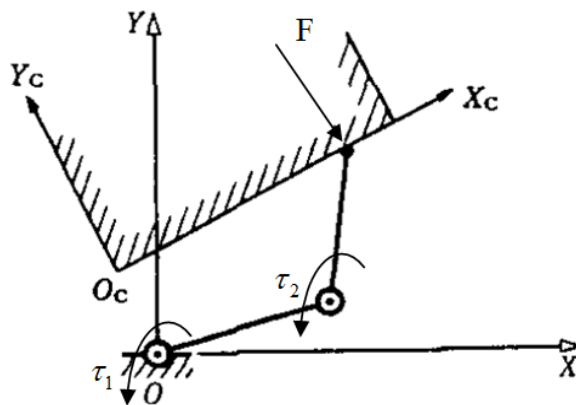


Figure 2.34 Example of hybrid control [Yoshikawa, 2000].

Several studies have been conducted to improve the response of the hybrid controller by introducing the dynamics of the manipulator to the controller [Yoshikawa et al., 1988]. The major disadvantage of this type of method is precise knowledge of the constraint surface is required in advance, to solve this problem, visual sensors and force sensors are combined for the control, with the visual sensor providing information on the contours of the environment (Figure 2.35). Nelson et al.[1995] summarized three different approaches combined vision and force sensing:

- *traded control*, a task space direction is alternately controlled using a vision sensor or force sensor
- *hybrid control*, different directions of the task space are simultaneously controlled using vision and force sensors
- *shared control*, both vision and force sensors control the same direction of the task space simultaneously.

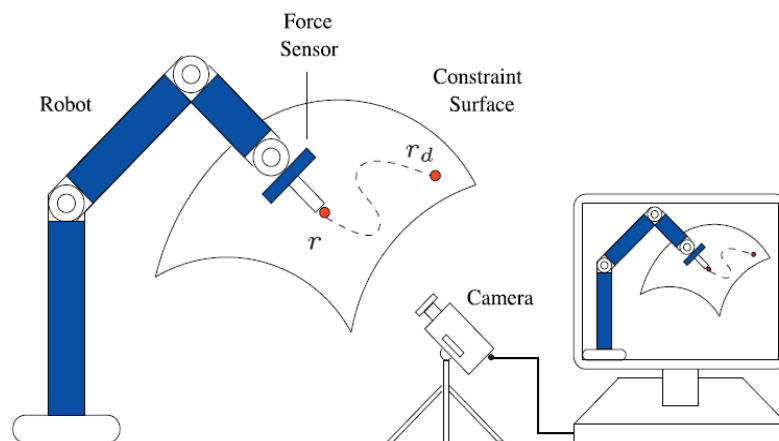


Figure 2.35 Vision-force system [Leite and Lizarralde, 2009].

Most recent work has focused on the hybrid vision and force controller. A hybrid control scheme that uses on-line estimators for the constraint geometry has been designed [Hosoda et al., 1998], with the controller only requiring prior knowledge of the robot kinematics instead of precise knowledge of the surface. Several controllers have been proposed to deal with the dynamic uncertainty, kinematics and constraint surfaces [Zhao and Cheah, 2004, Cheah et al., 2010]. Method also proposed by Leite and Lizarralde [2009] controls the robot using an un-calibrated camera. The

concept of hybrid vision and force controller can be utilized in this application since there are two external sensors employed for the control in this application. However, these proposed hybrid controllers were all used to follow a trajectory on an immovable surface, while this application has to cope with a moving object.

2.8 Safety issues

There are two safety issues which need to be considered in a robotic TMS system. One concerns TMS itself, the other involves the safety of the robot.

2.8.1 Safety of TMS

Seizure is the most critical adverse effect and is usually associated with rTMS at high frequency, high intensity, and short inter-train interval; no significant risk of single-pulse TMS has been found yet. Seizures have been produced both in patients and in normal subjects by rTMS with a stimulus frequency higher than 1 Hz [Wassermann, 1998]. To address this problem an international workshop on the risk and safety of rTMS was held in Bethesda, Maryland, June 1996. Safety guidelines were produced as a final achievement for the safety and ethical use of rTMS based on the known and potential risks of rTMS. Up to the present, no seizures have occurred by following this safety guidelines. However, researchers still have several safety concerns. According to [George et al., 2002], it is essential to note that this guideline was developed in a small subject sample using a surrogate endpoint for a seizure spread of TMS-induced motor evoked potentials (MEPs) beyond the target area of stimulation. This indicates that the guidelines apply only for stimulation of the motor cortex rather than other brain regions. Furthermore, although the intensity and frequency of stimulation can be examined, the inter-train interval is still not certain. One of the inadvertent seizures that did occur followed the safety guidelines because of an excessively short inter-train interval [Wassermann et al., 1996a].

2.8.2 Safety of robot application

As with medical robotics, a TMS robot is quite different application from an industrial robot, which indicates that the safety measures adopted for industrial robots are not sufficient for medical robots. However, most current robotic medical

or TMS systems employ industrial robots. This makes safety an issue that must be addressed to improve this field. There are no systematic approaches on the safety issues of medical robots and no specific standard safety guidelines for medical robots. But there are several standards which can be used as references, such as EN 755 (ISO 10218) a standard safety guideline for industrial robots. The FDA (Food and Drug Administration of the USA) have provided a safety standard for computer controlled medical devices and IEC 1508 is the standard for a safety-related system [Fei et al., 2001].

Taylor et al. [1991] defined several principal requirements for a medical robotic system: (1) the robot must never 'run' away; (2) the robot must never exert excessive force on the patient; (3) the robot's cutter must stay within a pre-specified positional envelope relative to the volume being cut; (4) the surgeon must be in full control at all times. According to [Davies, 1996], the use of redundant sensors, design of special-purpose robots and fail-safe techniques can be used to ensure the safety. The more autonomous the robot, however, the more safety critical is the dependency on the mechanical and software systems. Davies also catalogued some safety guidelines based on his experience. Software fault tolerance plays a very important role in medical robotic systems, and recently several studies have addressed this issue. Varley[1999] presented practical techniques for software development for the Pathfinder (Armstrong Healthcare Ltd.) medical robotic system. There are numerous other methods such as: event tree analysis [Khodabandehloo, 1996], fault tolerance algorithm [Hamilton et al., 1994], dependability principles [Dowler, 1995].

Fei et al. [2001] proposed a systematic method called hazard identification and safety insurance control (HISIC) to analyze, control and evaluate the safety of medical robots. They classified the system error into four parts: pure hardware, pure software, hardware induced by software, and software induced by hardware. The safety model developed by them consists of three parts: software, hardware and policy. The policy is the HISIC which includes seven principles: definitions and requirements, hazard identification, safety insurance control, safety critical limits, monitoring and control, verification and validation, system log and documentation. HISIC was implemented in the development of a robot for urological applications

and the initial results showed that HISIC had the potential ability to improve the safety of the system. Hawes et al. [2006] developed three software design principles for safety in cognitive systems: Concurrent modular processing, Structured management of knowledge and Dynamic contextual processing. In addition to these three principles, another three design parameters for safety in human-robot interaction: robustness, fast reaction time, and context awareness have been proposed by Giuliani et al.[2010]. These design principles have been applied to robot architectures and shown increased safety level for the human.

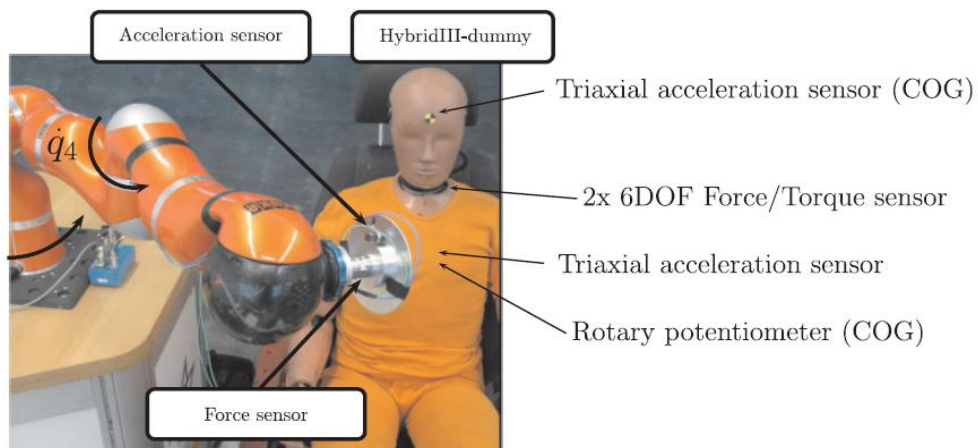


Figure 2.36 Setup of impact test using robot arm and dummy [Haddadin et al., 2009].

Haddadin et al.[2009] presented the first systematic evaluation and classification of possible injuries during physical human–robot interaction. They conducted a systematic series of tests with several industrial robots hitting crash test dummies and listed the injuries that might be caused by uncontrolled robot movements, which included fractures and shearing of limbs (Figure 2.36). They divided the contact between human and robot into five categories: unconstrained impacts, clamping in the robot structure, constrained impacts, partially constrained impacts, and resulting secondary impacts. The results showed that blunt head or chest impacts without clamping at a typical robot speed are, no matter how massive the robot is, definitely not life threatening. In the case of clamping both the head and chest can receive severe injury even leading to death, especially with increased robot mass (Figure 2.37).

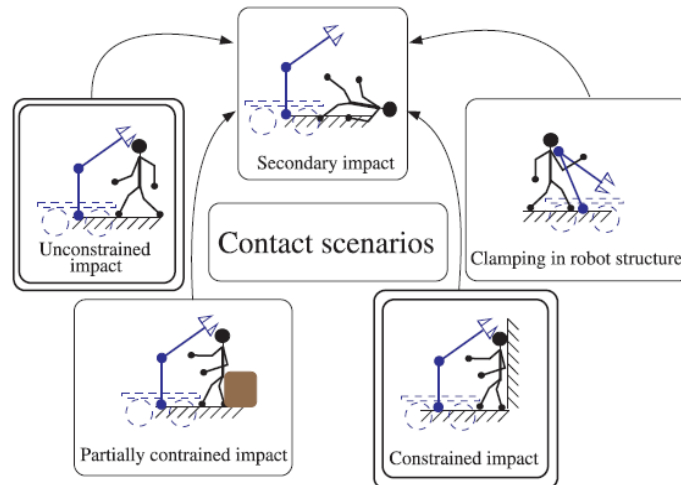


Figure 2. 37 Classification of undesired contact scenarios between human and robot[Haddadin et al., 2009].

2.9 Summary

This chapter has described the TMS technique and its therapeutic applications. The methods of coil placement and coordinate system registration in current hand-held TMS systems are also introduced. Several robotic surgical and TMS systems along with the relevant technologies used in these systems are introduced; and some of their conceptual design can be utilized for this application. Finally, the safety issues that are essential to human-robot interaction are described in both the TMS and robot sides. The next chapter will present the experimental procedures using current TMS system.

CHAPTER 3

PRELIMINARY TMS EXPERIMENTS AND DATA COLLECTION

Previous chapters have described the motivation and relevant literature and techniques behind the present work. This chapter discusses the preliminary TMS experiments and data collection using a hand-held TMS system. Magnetic stimulators are the key devices for TMS; and surface electromyogram (EMG) is a technique used for recording the muscle responses. A series of TMS experiments was carried out to get familiarized with the general conditions and procedures of TMS experiments using real subjects to obtain a better understanding of the problems experienced in a manual TMS system. The subject's movements were recorded using the Polaris Spectra system and the contact forces between the hand-held coil and the subjects' heads were also measured to provide useful data for the further development and specification of the robotic system. This chapter begins with the introduction of the TMS system.

3.1 The TMS system

The physical principle of magnetic stimulation is illustrated in Figure 3.1. It is based on Faraday's law of electromagnetic induction. A rapidly changing current field in a coil generates a changing magnetic field B which in turn induces a secondary current field E in the brain. The neuron's transmembrane voltage is stimulated by the electric field E and therefore causes local membrane depolarization and subsequent

neural activation. When sufficient neurons are activated, a macroscopic response may be observed. The response can be detected by directly observing the subject's motor behaviour (muscle twitch), or using electroencephalography (EEG), functional magnetic resonance imaging (fMRI), positron emission tomography (PET), single-photon emission computed tomography (SPECT), and surface electromyography (EMG).

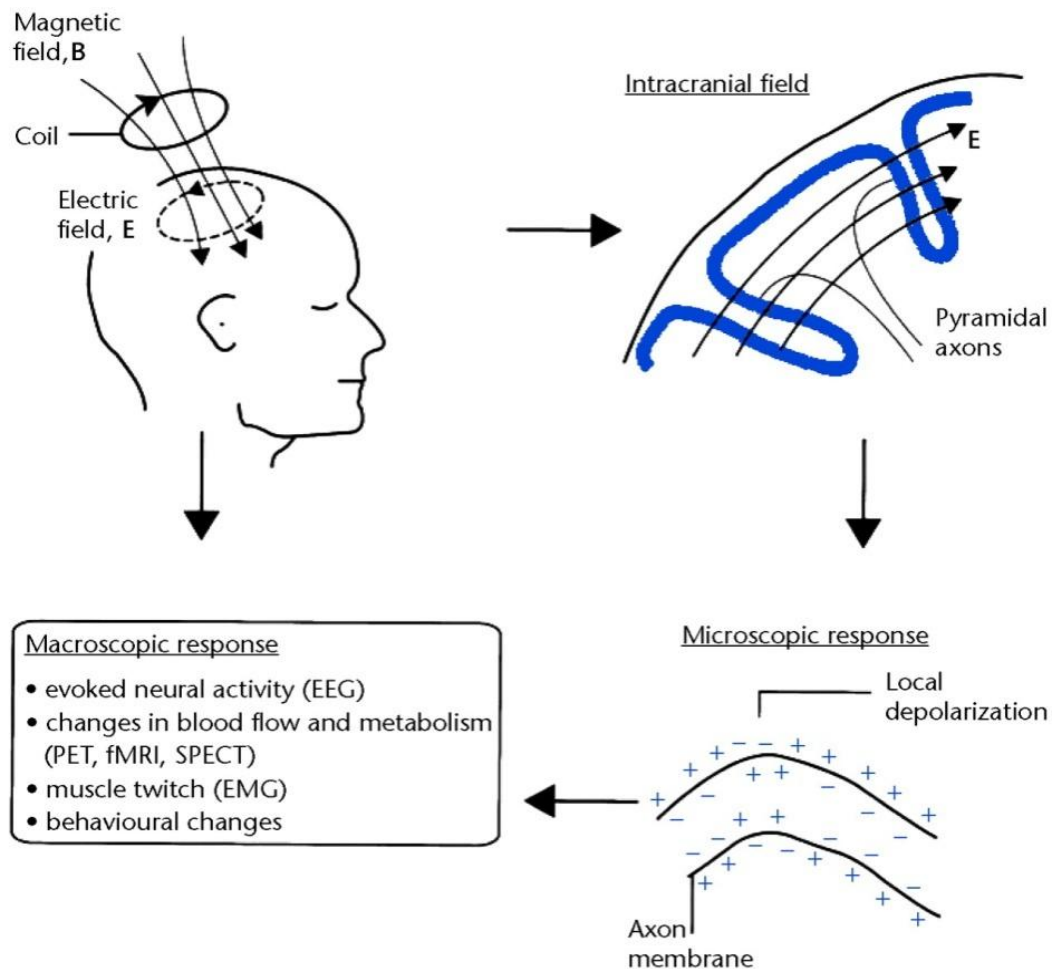


Figure 3.1 Principles of transcranial magnetic stimulation. The current in the coil generates a changing magnetic field (**B**) that induces an electric field (**E**) in the brain. The picture at the upper right depicts motor cortex stimulation and the trajectory of the pyramidal axons. At the microscopic level, the electric field **E** affects the transmembrane potentials and may lead to local membrane depolarization and subsequent neural activation. [Pascual-Leone et al., 2002].

EMG is commonly used for recording the muscle twitch during the TMS experiments since it is straightforward and convenient to apply on the selected

muscle. Major components of a TMS system are illustrated in Figure 3.2. A stimulator and a coil are used to generate the electric field and stimulate the subject's brain. A PC is used to trigger field and capture the surface EMG of muscle responses after they are amplified, filtered and converted to digital signals. There is an opto-isolator unit between the PC and subject to ensure the safety of the subject.

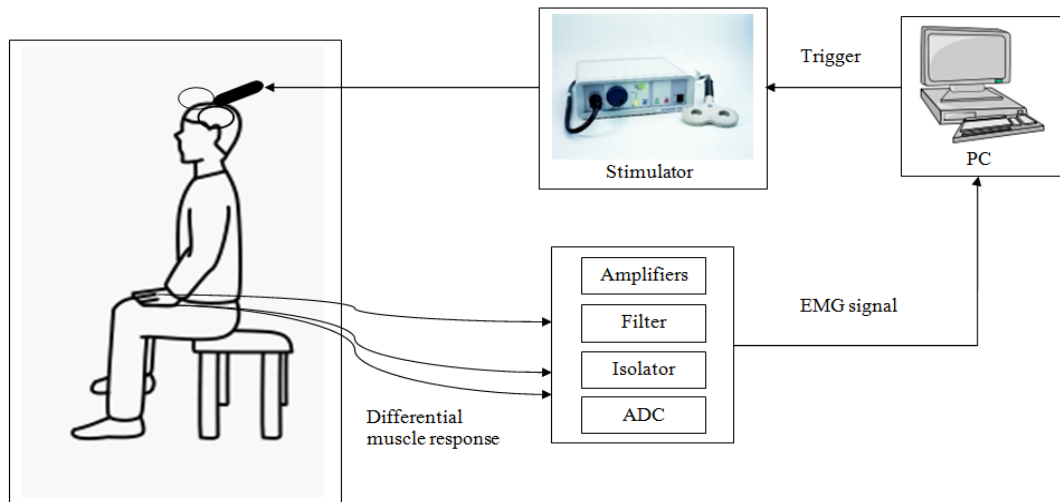


Figure 3.2 Major experimental devices used in the experiment.

3.1.1 Magstim 200 stimulator

There are five major units in a typical magnetic stimulator: a control unit, charging system, energy storage capacitors, discharge switch and the coil (Figure 3.3).

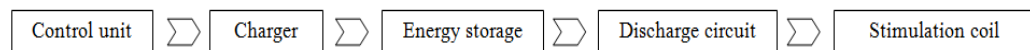


Figure 3.3 The major units of a typical magnetic stimulator.

Magsitm 200 stimulators (Magstim Company, Dyfed, UK) are used in this TMS application. Magnetic nerve stimulation can be achieved by discharging the stimulator with the stimulating coil in the vicinity of neuromuscular tissue. The magnetic field produced by the Magstim Model 200 depends on the type of coil connected. It can produce a magnetic field of 2 Tesla using a 13cm outside diameter circular coil. The power output can be set as a percentage of the maximum power

level; and the repetition rate of stimulus depends on this percentage. It takes 2 seconds to achieve a 50% power level, 3 seconds to 80% and 4 seconds to 100%. When there are two stimuli required in a short time interval, two Magstim 200 stimulators can be connected via a Bistim unit (Figure 3.4).

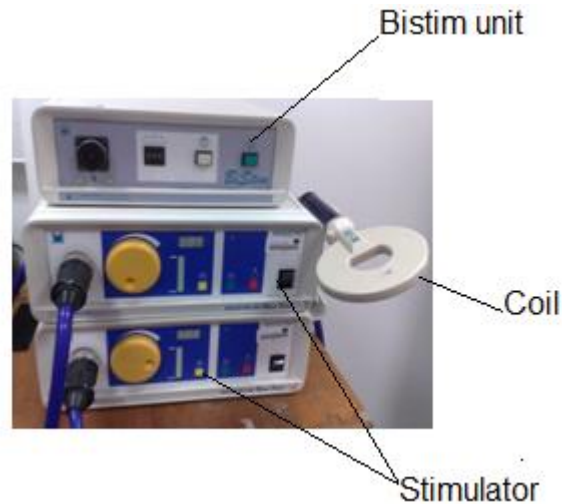


Figure 3.4 Two stimulator connected via a Bistim unit.

A PC is connected to the stimulator for triggering and recording the responses. In this way, the PC has to be synchronised so that the magnetic pulse produced by the stimulator occurs precisely at the start of the sweep of the recording instrument. The stimulator provides a trigger input socket to synchronise the stimulator to an external trigger input by application of a signal (TTL level). This input will result in the instrument being triggered when the stimulator and coil are in the ready state. In addition, a trigger output socket is provided to synchronise the stimulator to an external recording instrument. A 50ms pulse (TTL level) of either polarity is provided at this output whenever the stimulator is discharged. The trigger output of the PC should be connected to the trigger input of the stimulator, and the trigger input of the PC should be connected to trigger output of the stimulator in order to achieve the synchronisation.

Two commonly used coils, a single circular coil with 90mm mean diameter winding and a figure-of-8 shaped coil with two 70mm mean diameter winding are illustrated in Figure 3.5. The induced current is zero or near zero on the central axis of the circular coil (Figure 3.5(a)), while the induced current of a figure-of-8 shaped coil is

at its maximum under the centre of the coil where two windings meet (Figure 3.5(b)). Thus the figure-of-8 coil provides better focus, and can be used to stimulate deeper neural structures in the brain.

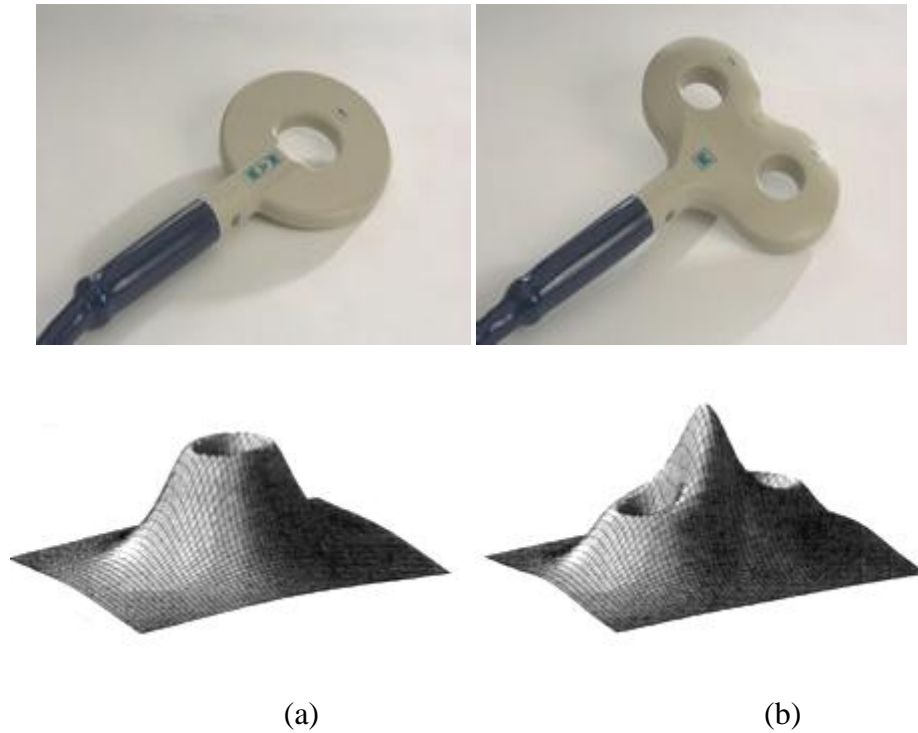


Figure 3.5 Single circular coil and figure of 8 shaped coils along with their induced electric field profiles.

The induced electric field 10mm from the coil surface is shown as a function of distance from the coil centre (Figure 3.6). The figure-of-8 shaped coil has a central peak amplitude double that of its peripheral peaks, which allows the coil to be used to stimulate neural structures under its centre. For the single circular coil, however, there is no electrical field induced in its centre, and the two peaks are symmetrical about the centre. The amplitudes of both coils vary with position, which indicates the stimulating amplitudes will change if the contact point between the coil and subject's head cannot be well maintained, which consequently affects the responses of the stimulation. Allowable errors were proposed according to this figure, the maximum allowable position error is 3mm, which loses about 2 percentage of the maximum amplitude of the maximum electrical field. The maximum allowable error of

orientation can be calculated using inverse trigonometric function accordingly, which is 1.5 degree.

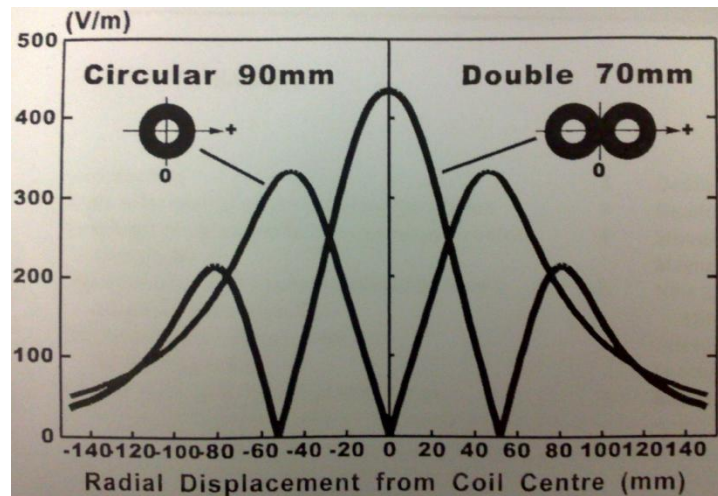


Figure 3.6 Calculated induced electric field 10mm against distance from the coil centre. The horizontal axis shows the radial displacement from coil centre, and the vertical axis shows the amplitude of the electric field [Pascual-Leone et al., 2002].

3.1.2 Surface electromyography (EMG) recording

The muscle responses of the magnetic stimulation are recorded using EMG, which is a non-invasive technique for measuring muscle electrical activity that occurs when muscle cells are electrically or neurologically activated. As shown in Figure 3.7, the EMG signals generated by the muscles are captured by the electrodes, then amplified, filtered and captured by the data acquisition module. After which, it can be recorded, processed and displayed. An opto-isolator is added to protect the subjects.

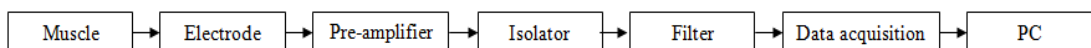


Figure 3.7 Detection of EMG signals.

Silver-silver chloride electrodes (The Natus Europe, Munchen, Germany) are placed in contact with the skin (Figure 3.8). The electrodes are either directly connected to the circuits, or indirectly connected via an extension cable.

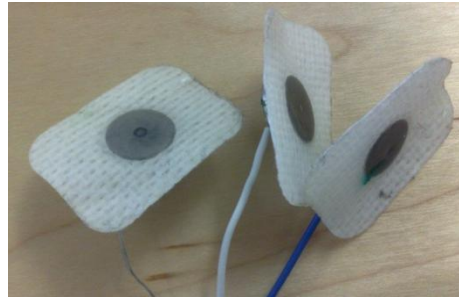


Figure 3.8 Disposable snap electrodes used in the application. One side is attached on subject's skin, and the other side is snapped on a cable and connected to the circuit.

A differential detecting configuration is employed in order to eliminate the potentially greater noise signal from power line sources. The differential amplification technique is shown schematically in (Figure 3.9). The signal is detected at two adjacent sites, and the signals are subtracted and amplified by the circuit to provide common-mode rejection of noise of the detection sites. Any signal that originates far away from the detection sites will appear as a common mode signal. According to the description, two detection electrodes and a reference electrode are needed to measure a target muscle's EMG signal. One detection electrode is directly attached on the target muscle, and the other is attached adjacent to the first one, but 5cm distance from its centre to ensure the difference. The reference electrode can be placed relatively far away from the target muscle.

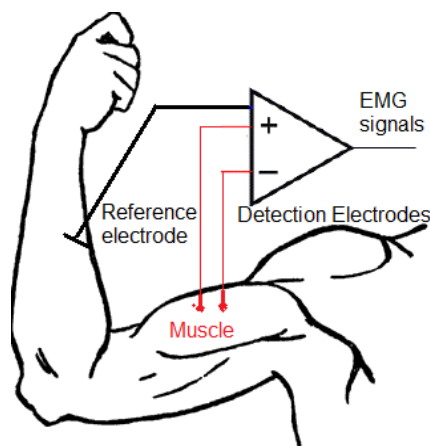


Figure 3.9 Differential amplifier configuration of electrodes.

Pre-amplifier, opto-isolator and filter modules are employed (Digitimer Ltd, UK) (Figure 3.10). The NL844 is a four channel, high impedance differential and low noise amplifier designed for use close to the human subject. The detection electrodes are connected to the 'REF.' and 'ACT.' connector of the amplifier, and the 'COM' connector is connected to the reference electrode to provide a reference potential for the amplifiers. The gain of each channel can be set to x100, x1000 and x10000 through the buttons on its rear, and there are three options of left cut off frequency: 3Hz, 10Hz and 30Hz. The amplified signals from the NL844 are connected to the four channel analogue opto-isolator NL820A. Four single-ended inputs with a common isolated terminal are provided together with positive and negative isolated supplies for powering the NL844 and the other circuits. In addition, each channel has adjustment of sensitivity by a three step switch to provide further amplification (x1, x2 and x5). The output of the opto-isolator is connected to the NL135 filter which is a 4-channel, two-pole low-pass filter module. The unit has 14 frequency settings giving repeatability over a wide range (10Hz-20KHz). The active Notch filter provides rejection of line frequency (50Hz) interference when switched in.

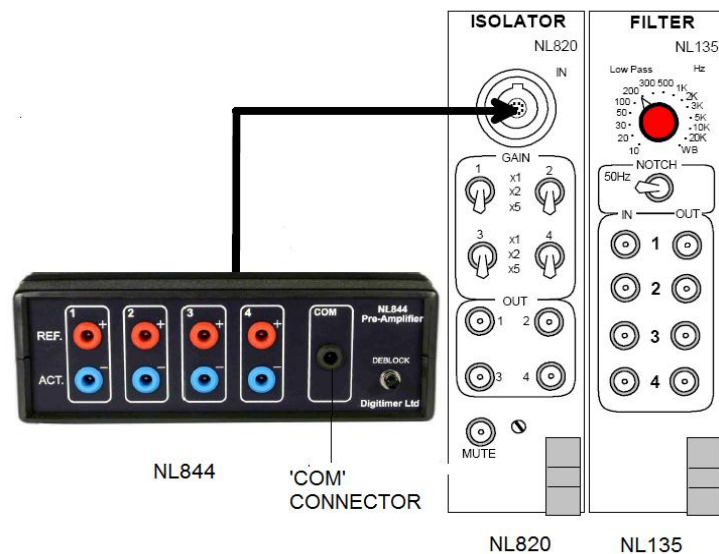


Figure 3.10 Layout of the amplifier, opto-isolator and filter.

The data acquisition module used in the application is Power1401 and the control software is Spike2 (CED, Ltd, UK). Power1401 is a high performance data acquisition interface which is used for recording waveform data and digital

information. It can also generate waveform and digital outputs simultaneously for real-time, multi-tasking experiment control. There are 16 channels of 16-bit waveform input on the unit, and normal working input range of these channels is $\pm 5V$, but this can be switched to $\pm 10V$. The maximum sampling rate is 800KHz in single channel mode, and 625KHz with multi-channels. The host interface is USB2.0 and PCI. Spike2 can also be used to trigger the stimulator during the experiments using the TTL or waveform outputs on Power1401.

3.2 Preliminary TMS experiments

Two TMS experiments were designed to familiarize the procedure and obtain an understanding of the problems experienced in setting up a manual TMS system. The first experiment was conducted to measure Chinese and Caucasian subjects' motor evoked potential (MEP) threshold, MEP recruitment and MEP latency at passive state. The second experiment was designed to measure the MEP threshold, recruitment and silent period (SP) in active state. For active state measurements, the subjects maintained a steady contraction of first dorsal interosseous (1DI) at a level of 5% of their maximum voluntary contraction (MVC); this was ensured via visual feedback of the mean rectified EMG using a computer display. Differences were also investigated between Chinese and Caucasians on various parameters measured from responses to TMS. The procedures for conducting the experiment and data analysis are introduced in the following sections.

3.2.1 Experimental procedure

Thirty two healthy Han Chinese and white European volunteers (16 in each racial group, 8 males and 8 females, age 21-26 years) consented to participate in this study, which was approved by the Local Research Ethics Committee of Newcastle University Medical School. Surface electromyogram (EMG) signals were recorded (bandpass 30Hz to 2kHz) from the right 1DI using bipolar surface electrodes. A Magstim 200 stimulator (Magstim Ltd, UK) was used for the motor cortical stimulation along with a 13cm outside diameter circular coil. Stimulus intensities were expressed as a percentage of the maximum stimulator output. The coil was placed over the vertex, with current direction optimal for left hemisphere activation.

Once the system had been set up, the procedure for the first experiment was as follows:

- Subjects sit comfortably and at rest (passive state).
- The electrodes are attached on the 1DI muscle of the right hand.
- Position for the stimulation is defined. This is achieved by positioning the coil at a standard start position known to be reasonably good for eliciting MEP in the target muscle. In this experiment, the centre of the coil is put over the vertex where it is estimated by the experimenter; and TMS intensity was increased in 5% increments until a reproducible MEP was observed at this position. After this, the coil was moved around a radius of a few centimetres until the response is largest at this stimulus intensity. This position was marked as the stimulating position.
- MEP threshold at rest is identified. The stimulus intensity achieved in the previous step was decreased in 1% steps until threshold was located (defined as a visible response to 5/10 stimuli). This value is referred to as the 'online threshold'.
- Data recording of the MEPs following different stimulus intensities. Single pulses TMS were delivered at intensities ranging from 15% below the online threshold up to a level where the MEP saturated, in 5% steps. Ten stimuli were delivered at each intensity.

The main procedure for the second experiment was similar to the first one, the only difference being in the first step, where the subject was required to sit and make a contraction of 1DI at 5% MVC (active state).

3.2.2 Results of the experiment

Stimulus markers and EMG waveform were captured continuously by computer at a 5 kHz sampling rate. Offline analysis was used to separate responses according to condition and to compile averages of the rectified EMG. Single sweep responses were measured as the area of the rectified EMG between the MEP onset and offset,

judged from the averaged response; single subject plots show the mean \pm SEM. An example of the averaged response is illustrated in Figure 3.11, the vertical red lines show the MEP onset (t_1) and offset (t_2), and the response amplitude R is given by:

$$R = \int_{t_1}^{t_2} y dt \quad (3.1)$$

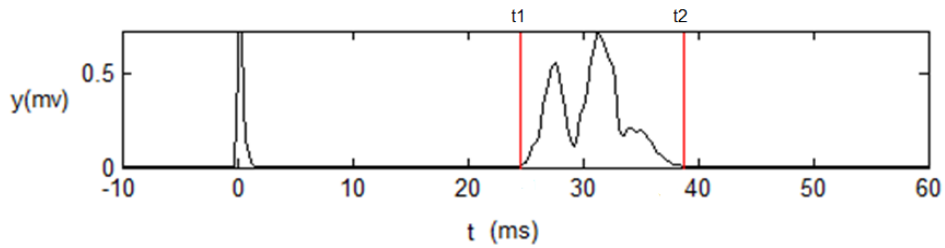


Figure 3.11 An example of the averaged TMS response.

A sigmoid curve was used to fit the relationship between response amplitude and intensity for both passive and active MEP recruitment relationships, according to the following relationship:

$$R = R_{\max} \frac{1}{1 + e^{-\frac{I - I_{50}}{k}}} \quad (3.2)$$

where R is the response amplitude at intensity I . From the parameters of this curve, we measured the intensity at which the response was half maximal (I_{50}), the maximal response (R_{\max}) and the parameter k , which is related to how quickly the response rises (the slope at intensity $I=I_{50}$ is $dI/dR=R_{\max}/4k$). Passive thresholds were estimated as the first intensity where the response was significantly different from zero; this is denoted here as the ‘offline threshold’. In the active state, the silent period duration was estimated by measuring the time at which the average returned to the baseline level. Box plots were compiled to summarize all parameters in each racial group; significant differences were assessed using the Mann-Whitney U test.

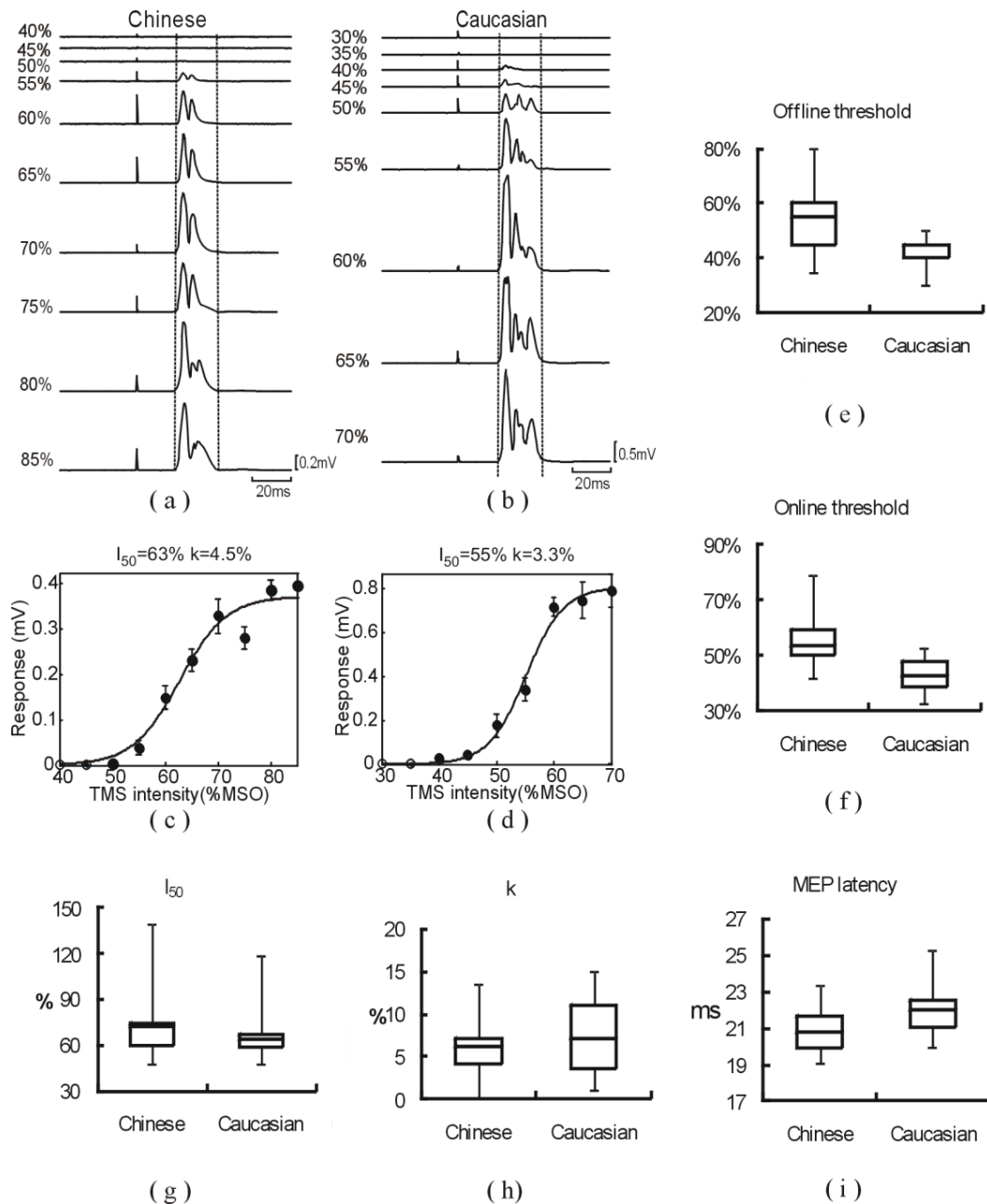


Figure 3.12 Results of single pulse stimulation in the passive state. Averaged rectified EMG from 1DI and MEP recruitment for a Chinese (a, c) and Caucasian (b, d) subject following single pulse stimulation at different intensities. Vertical dashed lines show the response region. Filled circles show responses significantly different from zero (t-test, $P < 0.05$). Error bars show standard errors. (e)-(i), box plots showing results across the subject population for online threshold, offline threshold, I_{50} (the stimulus intensity producing a half-maximal response), parameter k (indicating how quickly responses grew with increased intensity) and MEP latency. The boxplots show the maximum, minimum, 1st quartile, median and 3rd quartile.

Figure 3.12(a)-(d) show typical results of one subject from each group. Figure 3.12(a) is the averaged EMG response following single pulse TMS at different intensities in a Chinese subject; these were quantified to yield a MEP recruitment curve in Figure

3.12(c). Figure 3.12(b) and (d) present similar displays for a single Caucasian subject. For these two cases, the Chinese subject had a higher offline threshold (55%) than the Caucasian (40%); a similar trend was seen in the I_{50} (63% vs 55%). The response grew more slowly with increases in intensity in the Chinese compared with the Caucasian subject (value of parameter k of 4.5 % and 3.3 % respectively). Figure 3.12(e)-(i) present group data as box plots for each measure made from the responses at rest. Both offline threshold (Figure 3.12(e)) and online threshold (Figure 3.12(f)) showed clear significant differences between the two groups ($P=0.002$ and $P=0.0001$ respectively). By contrast, there were no significant differences in the I_{50} or k parameters between the groups (Figure 3.12(g) and (h), $P>0.05$). There were also no differences in MEP latency (Figure 3.12(i), $P>0.05$).

Figure 3.13 presents similar measurements made in the active state. Single-subject responses and the corresponding recruitment curves are shown for a Chinese subject (Figure 3.13(a) and (c)) and a Caucasian (Figure 3.13(b) and (d)). In these subjects, the Chinese had a higher threshold (35%) than the Caucasian subject (30%), and larger k (6.8 % vs 2.9 %), indicating that responses grew more slowly with increasing intensity. Across the population, there was no significant difference in active threshold between the two populations (Figure 3.13(e), $P>0.05$). Unlike the passive state results, the I_{50} in the active state did show a significant difference between Chinese and Caucasians (Figure 3.13(f), $P=0.007$). A small difference in k just failed to reach significance (Figure 3.13(g), $P=0.067$).

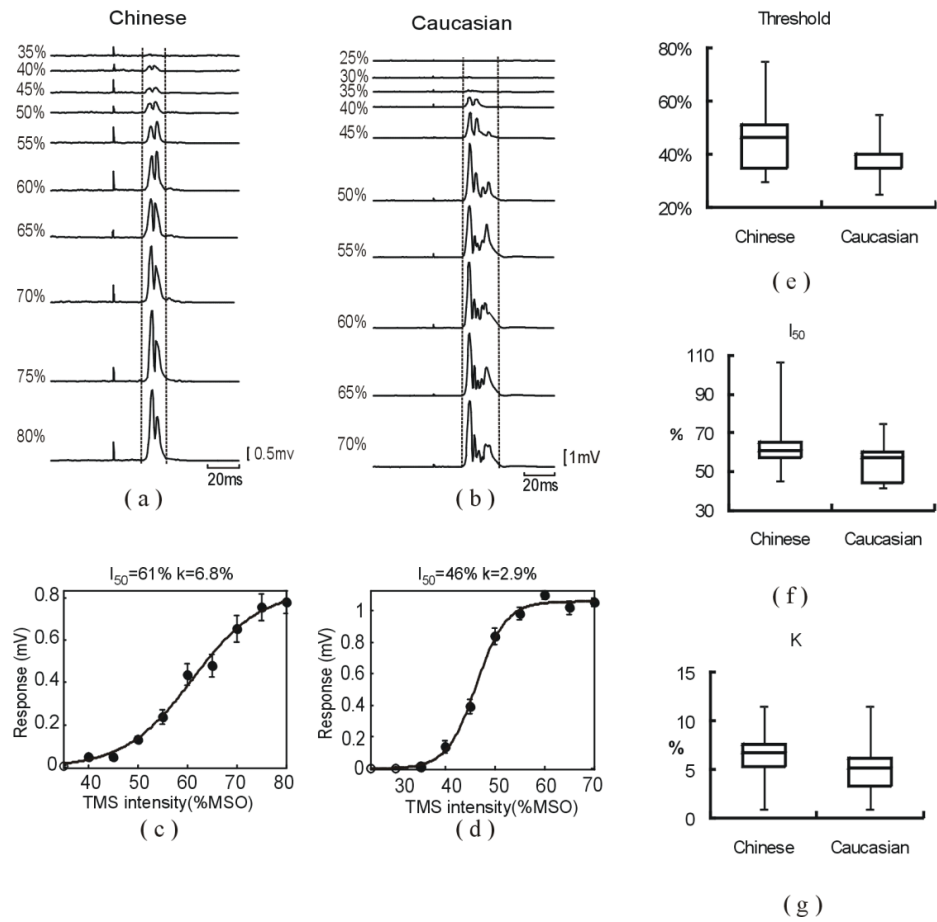


Figure 3.13 Results of single pulse stimulation in the active state. Averaged rectified EMG from 1DI and MEP recruitments are shown from a Chinese (a, c) and a Caucasian (b, d) subject following single pulse stimulation at different intensities. Vertical dashed lines show the response region. Filled circles show responses significantly different from zero (t-test, $P < 0.05$). Error bars show standard errors. (e)-(g), population data for active threshold, I_{50} and parameter k . The boxplots show the maximum, minimum, 1st quartile, median and 3rd quartile.

Figure 3.14 illustrates data on the silent period. Individual responses in Figure 3.14(a) and (b) showed a clear silent period. The estimated offset time of the silent period has been marked by a vertical dotted line, for each intensity where a silent period could be discerned. Plots of the offset latency versus intensity revealed an approximately linear relationship (Figure 3.14(c) and (d)), allowing estimation of the slope of the best-fit line. Measurement of the silent period at threshold just failed to show significant differences between Caucasians and Chinese (Figure 3.14(e); $P = 0.069$). However, at intensities of 10% maximum stimulator output above threshold, the silent period was significantly longer in Caucasians (Figure 3.14(f);

P=0.014). The slope of the silent period duration versus intensity relationship was also higher in Caucasians than Chinese (Figure 3.14(g); P=0.002).

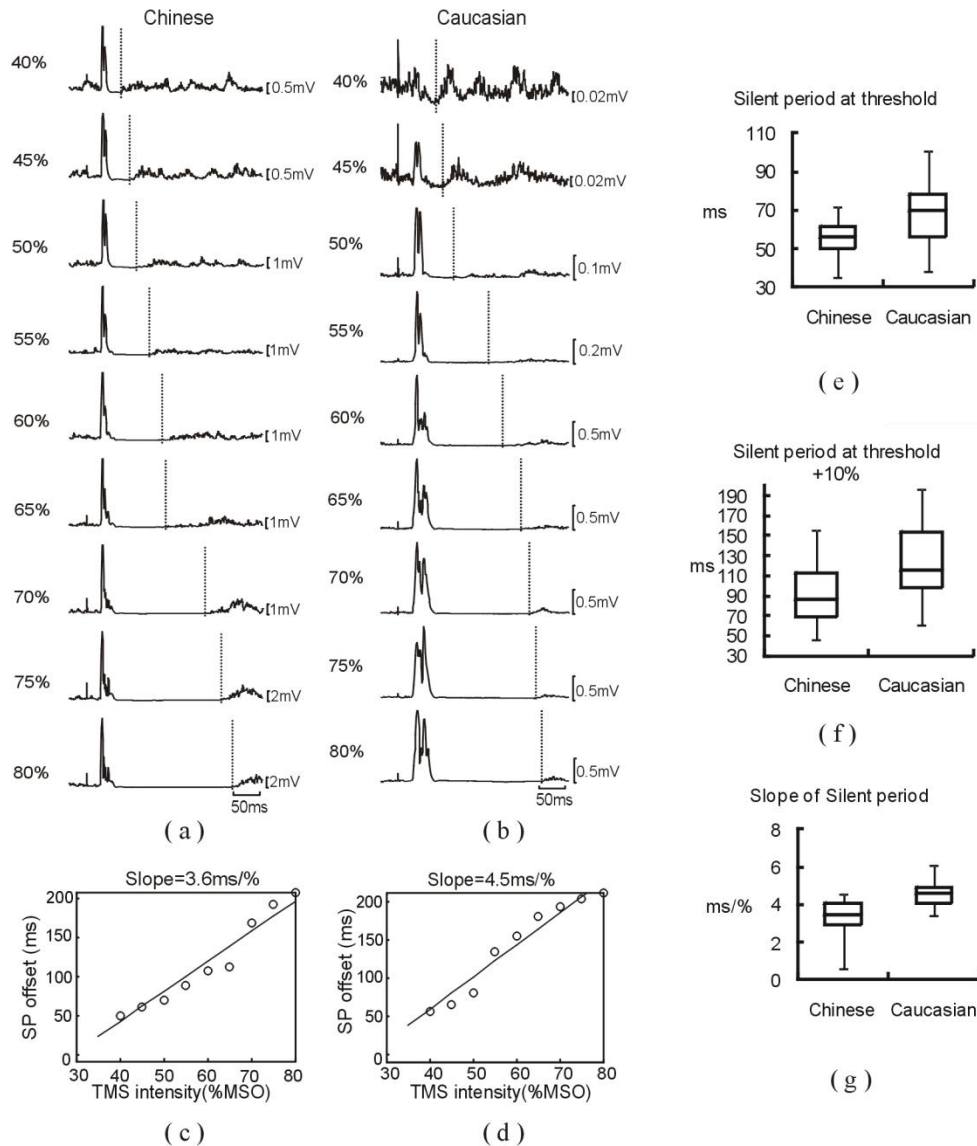


Figure 3.14 Results of silent period measurement. Averaged rectified EMG from 1DI is shown from a Chinese (a) and a Caucasian (b) in the active state. The vertical dotted lines indicate the silent period offset at different intensities. (c)-(d), relationship between silent period offset and intensity, for the subjects shown in (a,b). Overlain line is the best fit straight line, with slope calculated as shown above each plot. (e)-(g), boxplots of population data on silent period at threshold and threshold+10%, and the slope of the silent period offset-intensity relationship. The boxplots show the maximum, minimum, 1st quartile, median and 3rd quartile.

3.2.3 Discussion of the preliminary TMS experiments

This study provides the first objective evidence that there are differences between TMS measurements across different racial groups, supporting anecdotal accounts

from several laboratories. Interestingly, no differences were seen in MEP latency. This suggests that central motor conduction time (CMCT) will be comparable between the groups, and can reasonably be interpreted without reference to race. By contrast, passive threshold and silent period all appeared different. It would be unsafe to use normative data on these measures gathered from healthy volunteers in a single racial group to diagnose abnormality in a wider range of patients. Furthermore, gathering normative data in heterogenous populations without consideration of race will increase the variability in the measures and hence reduce their sensitivity to detect abnormality.

Our results appear to show two distinct differences between Chinese and Caucasian subjects. The first is a lower passive threshold in Caucasians; by contrast the active threshold was similar. The most obvious possible explanation for differences is that the skull shape is subtly different between the two groups. This could modify current flows within the brain, leading to a different effective stimulus at the same percentage of maximum stimulator output. However, such an explanation cannot account for the unchanged active threshold. If changes in current flow underlie our observations, we would expect a simple shifting to higher intensities of all measures in Chinese subjects, rather than a selective effect on the passive threshold.

In switching from the passive to the active state, there are changes in both corticospinal [Baker et al., 1995, Di Lazzaro et al., 1998] and motoneuron excitability. Our results suggest that overall excitability is different between Chinese and Caucasian subjects at rest, but that these differences disappear during an active contraction. This implies that the changes from rest to active must be more profound for the Chinese subjects. It is unclear whether these differences lie at the spinal or cortical level; further studies, for example using H reflex testing or responses to corticospinal stimulation at the cervico-medullary junction [Ugawa et al., 1991], could resolve this issue.

In this study, the relationship between stimulus intensity and silent period duration was well fitted by a straight line. A sigmoid curve may provide a more suitable fit [Kimiskidis et al., 2005], since silent period duration does not increase above a certain stimulus level [Valls-Sole et al., 1994]. However, in our experiments the

duration of the silent period did not reach a plateau for most subjects, justifying the approximation of a linear fit in this case.

In conclusion, we have shown differences between Han Chinese and White European healthy subjects on several TMS parameters. Although the causes of these differences remain to be clarified, future clinical and experimental studies using TMS should be aware of the differences.

3.3 Practical considerations

By conducting the experiment and data analysis, there are several practical considerations that were identified during the experiment. The contact force between the coil and the subjects' head affect not only the comfort of the subjects, but also the response of the stimulation. Another key issue affect the stimulation result is the inevitable movement during the experiment.

3.3.1 Contact force between the coil and subjects' head

The contact force between the coil and the subjects' head affects the subjects' comfort, and presumably the lower the contact force the more comfortable the subject will be. The 90mm diameter single circular coil and 70mm figure-of-8 shaped have weights of 1.4kg and 1.61kg respectively. If they are placed on the subjects' head, then their weight alone can cause significant discomfort to the subjects especially in a long duration experiment. However, the actual contact force is related to how the experimenter supports the coil.

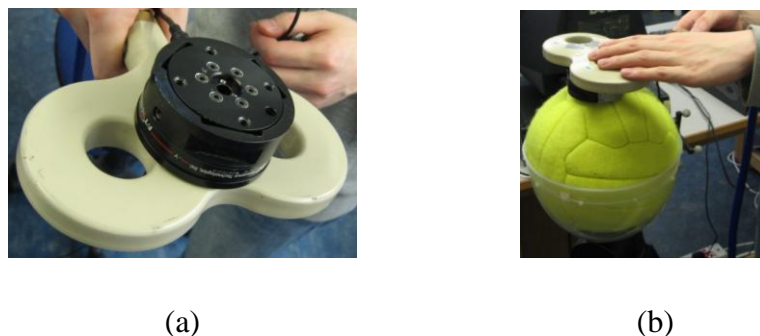


Figure 3.15 Measurement of the contact force by attaching a force sensor on the coil.

The contact force was measured by attaching a force sensor on a figure-of-8 shaped coil as illustrated in Figure 3.15(a). The handle of the coil was held in the experimenter's right hand and the other hand was placed on top of the coil to maintain its position, as shown in Figure 3.15(b). In this way, the contact force largely depends on how hard the experimenter pushes on the coil. The force data was recorded for 2 minutes on each of 8 subjects with a gentle press on the coil. The results are illustrated in Figure 3.16, the mean contact force was measured at 7.5N with a gentle push on the coil.

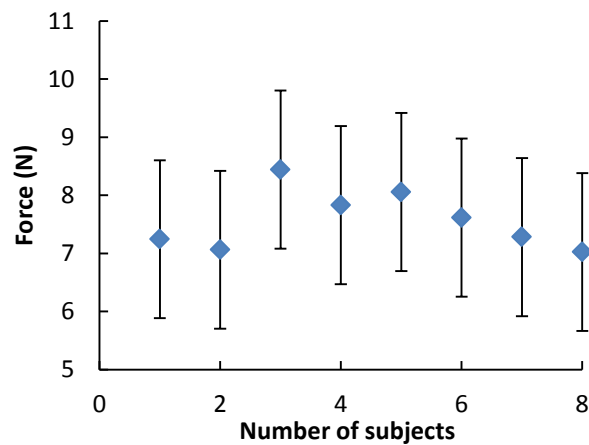


Figure 3.16 Contact force recorded from 8 subjects respectively. Error bar show the maximum and minimum force during the measurement.

Averaged responses from 1DI using different contact forces are illustrated in Figure 3.17. It can be seen that the responses of 2N and 7N have similar amplitude, however, there is loss of response when the contact force is 1N .

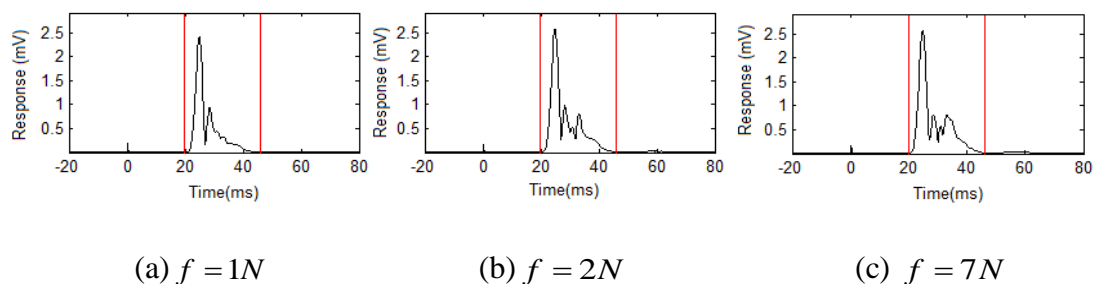


Figure 3.17 Averaged response on 1DI using different contact force between the coil and subject's head. Vertical lines show the response region (data gathered from Baker's lab).

3.3.2 Effects of the inevitable movement

The inevitable movements of both subject and experimenter during the experiment have a significant impact on the responses of the stimulation. Figure 3.18 demonstrates two different responses obtained from the same subject during the experiment. Figure 3.18(a) shows the averaged EMG response at an intensity of 60%, when the stimulation position is well maintained during the experiment, resulting in an amplitude of response up to 0.4mv. However, during movement the amplitude falls to less than 0.2mv as shown in Figure 3.18(b). This is due to the three factors that affect the TMS response of a given neural element: depth from coil; location within the induced field and orientation with respect to the induced current [Pascual-Leone et al., 2002]. These three factors can be easily influenced by the relative change in position between the coil and subjects' head caused by the inevitable movements. This implies the relative position and orientation between the coil and head should be maintained during the experiment to ensure better accuracy of TMS stimulation. In addition, according to these three factors, the optimal position and orientation of the coil, to obtain an optimal response, is to ensure that it is positioned at the shortest distance from the neural element, and it should be oriented tangentially to the surface to achieve the symmetry of the induced electrical field.

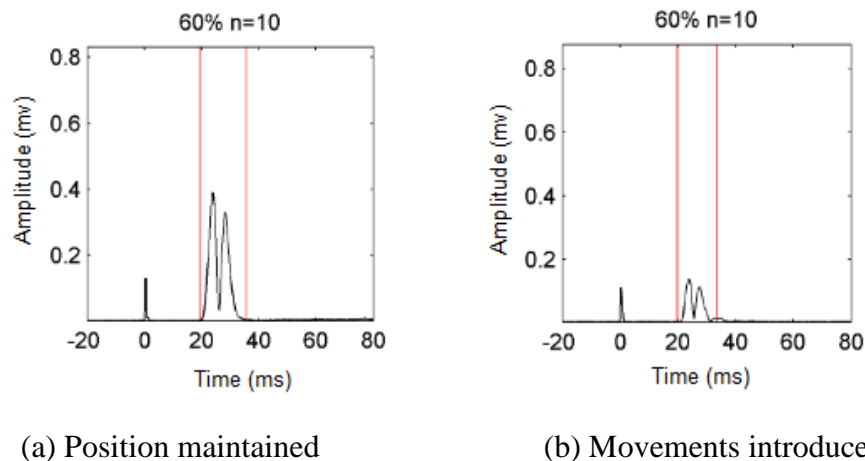


Figure 3.18 Different amplitudes of the muscle response obtained due to inevitable movements during the experiment. Vertical lines show the response region, n indicates 10 stimuli are delivered.

Movements of 8 normal subjects were recorded using the Polaris Spectra tracking system in order to quantify subject's movements during the TMS experiments. A passive tool was attached on subjects' forehead as shown on the head phantom (Figure 3.19), and the movement data was recorded by using the procedure described in Section 3.2.1.



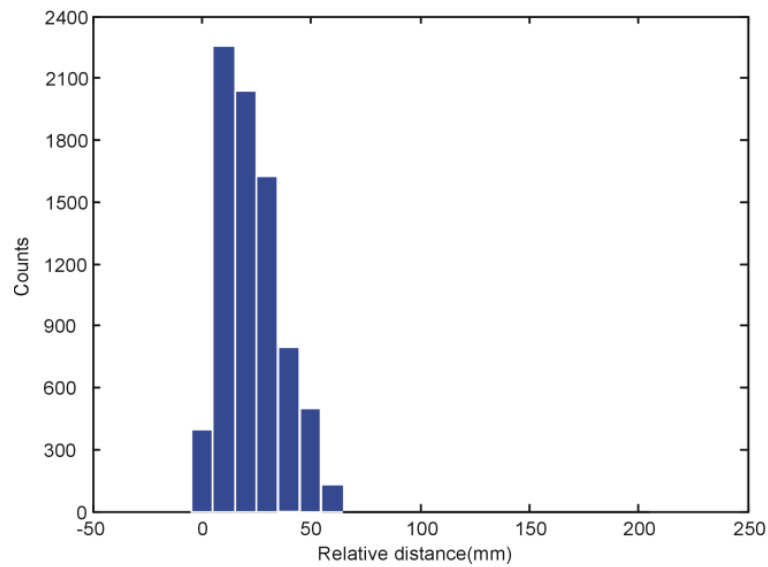
Figure 3.19 Passive tool is attached on the forehead of the subjects.

The distributions of the relative movement from the original starting positions are shown in Figure 3.20. The update rate of the sensor was set to 60Hz, and the root mean square of the relative distance D_{rms} was calculated using:

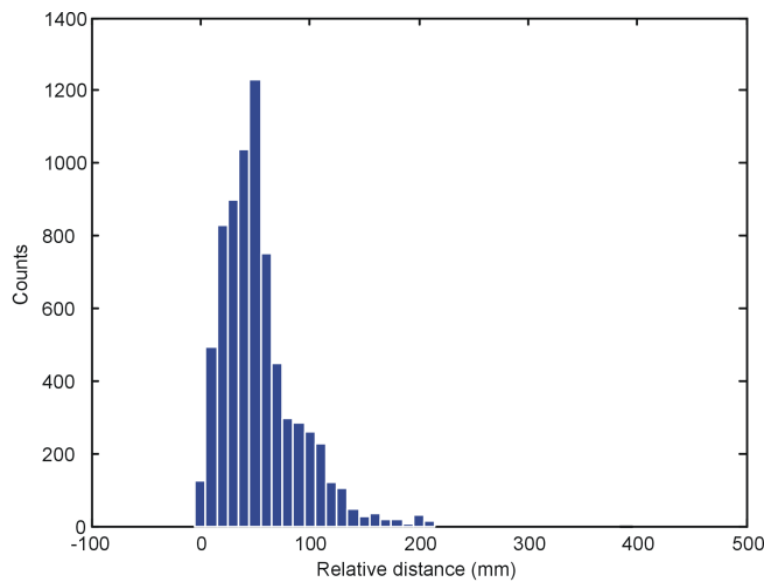
$$D_{rms} = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2} \quad (i = 2, 3, 4 \dots n) \quad (3.3)$$

where x_i, y_i, z_i are the i -th coordinates of a subject's head, x_1, y_1, z_1 is the initial position of the subject's head, n is the number of recorded positions, and the calculated distances are relative to their original positions. Figure 3.20(a) shows the results obtained during the experiment, and it can be observed that the subjects can maintain their position with small movements in a TMS experiment. The maximum displacement was recorded as 60mm; with most positions maintained between 10-20mm from their original position. Figure 3.20(b) illustrates the distribution of movement when subjects were allowed to stretch their neck or became distracted. Although these unpredictable movements can lead to motions more than 200mm on

some occasions, the changes in positions were distributed a distance of 20-60mm from their original position.



(a) Subjects trying to maintain their position



(b) Subjects do movement during the test

Figure 3.20 Distribution of the relative distance from the original position. (a) shows distances when the subjects trying to maintain their position. (b) shows distances when the subjects trying to relax their muscles or disturbed by outside environment.

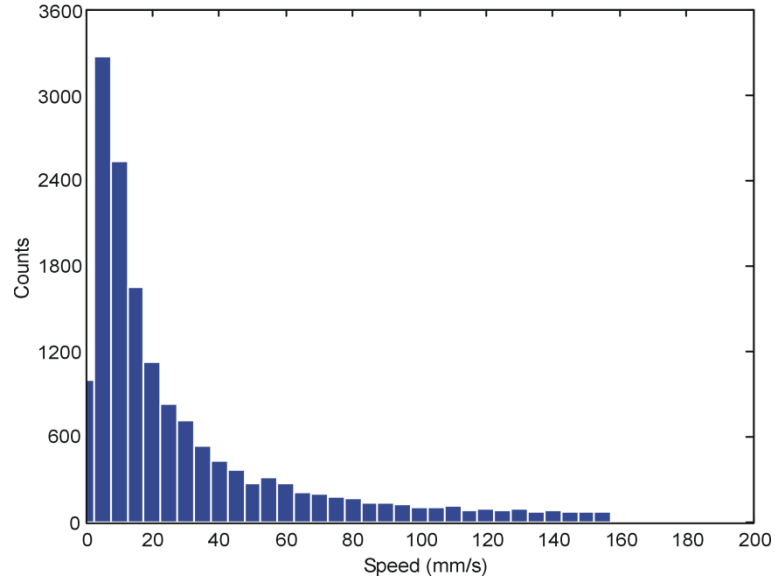


Figure 3.21 Distribution of the speed when subjects moving their heads.

The velocities of these movements were also estimated using:

$$V(x_{(i+1)T}, y_{(i+1)T}, z_{(i+1)T}) = \frac{|(x_{(i+1)T}, y_{(i+1)T}, z_{(i+1)T}) - (x_{iT}, y_{iT}, z_{iT})|}{T} \quad (i = 1, 2, 3 \dots n) \quad (3.4)$$

where x_i, y_i, z_i are the i -th coordinates of the subject's head, n is the number of recorded positions, T is the sampling time. The distribution of the speeds is demonstrated in Figure 3.21, with 70% of the speeds recorded less than 20mm/s, although a maximum of 160mm/s was measured.

3.3.3 Discussion

The preliminary experiments were carried out to obtain a better understanding of the TMS procedure. The contact force between the coil and the subjects' head is a factor that can cause discomfort especially in a long duration experiment. The contact force was found to be approximately 7.5N with a hand-held coil when there is no big movement introduced. For a robotic system, the coil is supported by the robot arm and the contact force should be maintained at a level that will cause the minimum discomfort to the subjects on the premise of the accuracy of TMS stimulation. A force of 2N has been proposed as it can ensure the quality of the stimulation and cause much less discomfort than 7N.

The inevitable movement during the experiment is a key factor that will affect the accuracy of the stimulation. The subjects' movements are normally in a 300mm cube. Most of the position changes caused by movements are less than 10mm with speeds of less than 20mm/s. These two parameters can be used for the tests on system behaviour in the future. Since there is no existing standard for a robotic TMS system, the main specifications of the robotic system is proposed according to the previous experiments and analysis as follows.

Table 3.1 Main specifications of the robotic TMS system

Parameters	Values
Tracking volume	300*300*300 cube
Contact force	Less than 7.5N which is the manual mean contact force
Maximum allowable error of position	3mm when the speed of subject is less than 20mm/s
Maximum allowable error of orientation	1.5 degree when the speed of subject is less than 20mm/s

Although the major task of the robotic system is to compensate the subject's movements and control the contact force in real time, its operation should follow the procedures of the TMS experiments. As illustrated in Figure 3.22, first of all, the robotic system should be initialized (i.e. mount appropriate coil, set the gains of the amplifier etc.), and the robot arm is at a safe position where it is easy for the experimenter to start the TMS experiment. The stimulation site should then be determined and the robot is commanded to place the coil on the stimulation site. Once the target site is defined and the coil is in position, the relative location between the coil and subject's head should be recorded. The sequences of stimuli are then triggered and the robot arm is commanded to execute the position and force tracking. Once the experiment is finished, the system is stopped and the robot arm returned to the start position.

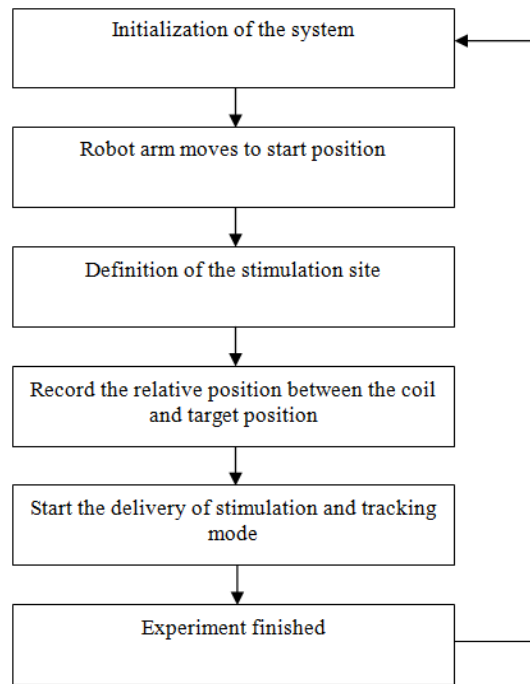


Figure 3.22 Preliminary sequence of operating the system.

3.4 Summary

This chapter describes the operation of a manual TMS system. A series of preliminary TMS experiments were carried out, and surface EMG recording and data analysis were undertaken. These experiments provided the first objective evidence that there are differences between TMS measurements across different racial groups. In addition, an understanding of the problems experienced in a manual TMS system was obtained during the experimental process. The issues of force contact and inevitable movements during the experiment are addressed. A force of 2N has been proposed as it can ensure the quality of the stimulation and cause less discomfort. The subjects' movements are normally within a 300mm cube, and most of the position changes caused by movements are less than 10mm at speeds of less than 20mm/s. The design of robotic TMS system is introduced in next chapter.

CHAPTER 4

SYSTEM DESIGN AND PRELIMINARY EVALUATION

This chapter describes the design of the robotic TMS system; including the architecture and control system. The key components of the system including Polaris Spectra tracking system, Staubli TX60 robot arm and an ATI F/T force sensor are described. The Polaris tracking system consists two main components: position sensor and passive reflective markers. The accuracy of the Polaris Spectra tracking system within a pre-defined region was evaluated using a 3-axis machine tool (Newall, UK, Ltd) which has a high accuracy digital readout system. The requirements for the design of the reflective rigid body markers were addressed, in addition to the initial tests of force sensor and techniques for defining the tool centre point. Finally, the protocol and established data exchange rate between robot and PC controller was tested.

4.1 System concept design

4.1.1 System architecture

As presented in Chapter 1, the robotic TMS system must be capable of finding the ‘hot-spot’ (optimal stimulation site) and holding a TMS coil over a fixed location and orientation on the subjects’ head. In addition, the contact force and small movements of the unrestrained head due to postural sway or the subject altering their sitting position should be tracked and corrected in real time by the robot arm. According to these requirements, a schematic layout of the system is illustrated in Figure 4.1. There are three main components required to realize these functionalities mentioned above: a position tracking system, a robot arm and a force transducer.

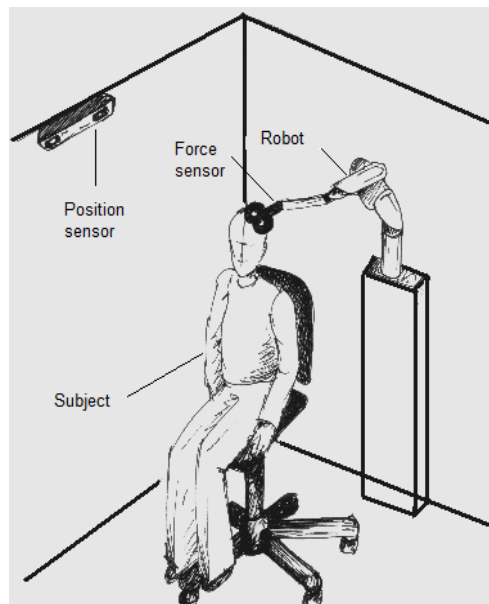


Figure 4.1 Conceptual layout of the system. The subject sits on a chair, a robot arm holds the coil in contact with the subject’s head. A position sensor and force sensor are used for the data recording.

According to the initial concept design, the system architecture was developed as illustrated in Figure 4.2. A Staubli TX60 robot arm which has six degrees of freedom is used to hold and position the coil. An ATI Mini40 F/T Force Transducer is mounted between the robot tool flange and the coil (Figure 4.3) for sensing the direction and magnitude of the contact force, the force data is acquired using a PowerDAQ PCI DAQ card (Model: PDL-MF, UEI, US). The position of the

subject's head is tracked using an optical tracking device (NDI Polaris Spectra); it is also used for guiding subsequent refinement of the coil position. A PC running the QNX Neutrino real-time operating system is used as the main force/position controller to acquire force and position sensor data, and an Ethernet link is established between the CS8C robot controller and PC controller. Further detail about these components is discussed in the following sections.

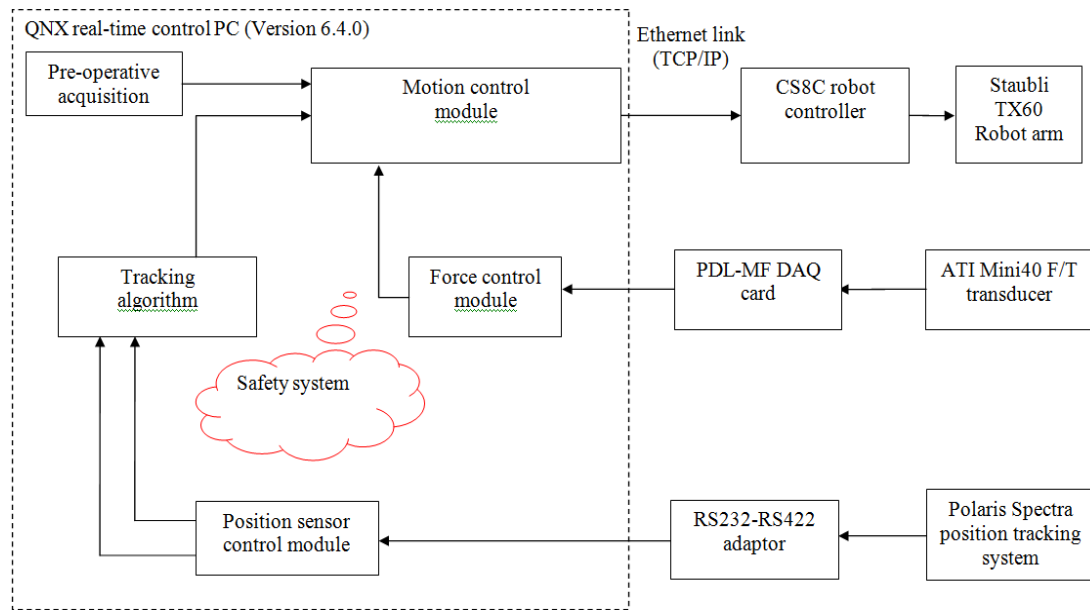


Figure 4.2 Proposed control system.

4.1.2 Proposed control system

Figure 4.2 illustrates the proposed control functional block diagram; there are four main modules running in real-time on the PC controller: tracking module, position sensor control module, force control and motion control. The work reported in this thesis focuses principally on the design of the tracking module (combined position and force tracking), motion control and position sensor control module. The force control and safety system are being developed by a colleague [Zakaria, expecting to submit in 2012].

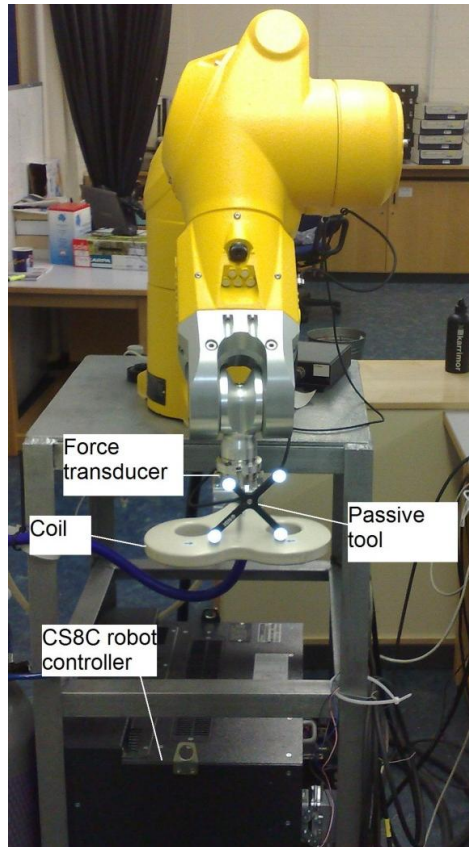


Figure 4.3 Coil and force transducer mounted on the robot.

The robotic TMS system has two modes of operation, a pre-operative acquisition mode and a tracking mode. Pre-operative acquisition provides for registration of the different coordinate systems and determination of the stimulation site, and is carried out in the sequence illustrated in Figure 4.4. The coordinate system registration includes registration of the robot and Polaris world coordinate systems, and registration between the MR image and the subject's head coordinate system. There are three ways to find the hot-spot: a manual 'hot-spot' finder is used to determine the stimulation site by manually moving the coil in the pre-operative mode and requires an experienced neuroscientist to conduct the task. A real-time force sensing algorithm is used to allow the experimenter to manually manipulate the robot arm by grasping a handle located in the head of the coil. Using this 'tele-operated' mode, the operator can search for and acquire the desirable stimulation site. An automatic 'hot-spot' finder using Bayesian inference to iteratively update the location of the 'hot-spot', and the robot moves correspondingly until it acquires the target site. This method was developed by Baker [Expecting to submit in 2012]. The MRI viewer

can also be used for pre-defining the ‘hot-spot’. Following the registration between the image and the subject’s head coordinate system, the experimenter can define the stimulation site on the image, and manually find the target on the subject’s head accordingly. In addition, the current stimulation site can be displayed during the stimulation procedure. Registration of the MRI image is addressed in Chapter 7. Following registration with the coil at the target site, the tracking mode is activated. In this mode, real-time tracking and force sensing are used to measure and regulate any change in position whilst attempting to regulate contact force, and is then used to compute the correct trajectory and guide the robot.

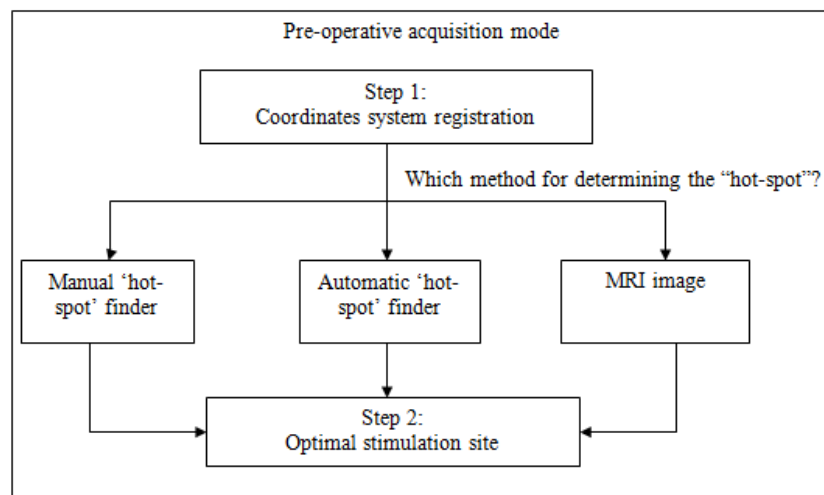


Figure 4.4 Content of pre-operative acquisition mode.

4.2 Preliminary evaluation on Polaris Spectra optical tracking system

The real time position control is a fundamental requirement of the system and an appropriate tracking system should be employed. As described in Chapter 2, there are two main types of tracking systems: electromagnetic tracking systems and optical tracking systems. The main disadvantage of electromagnetic tracking systems is that they suffers from interference, and consequently the induced magnetic field of TMS can cause distortion and impact the accuracy of the positions, and thus are unsuitable in this application. Although optical tracking systems have a drawback of line-of-sight, this can be mitigated by properly placing and fixing the tracking system. For

these reasons, an optical tracking system: Polaris Spectra (NDI, Waterloo, CA) was adopted in this system.

4.2.1 Overview of Polaris Spectra system

This section describes the major features of the Polaris Spectra system. It comprises two main components: the position sensor and the passive tool (Figure 4.5). The origin of the world coordinate system for the tracking system is located at the centre of the position sensor; a local coordinate system is defined for each tool. In this way, both position and orientation of the tool with reference to the tracking system frame can be determined.

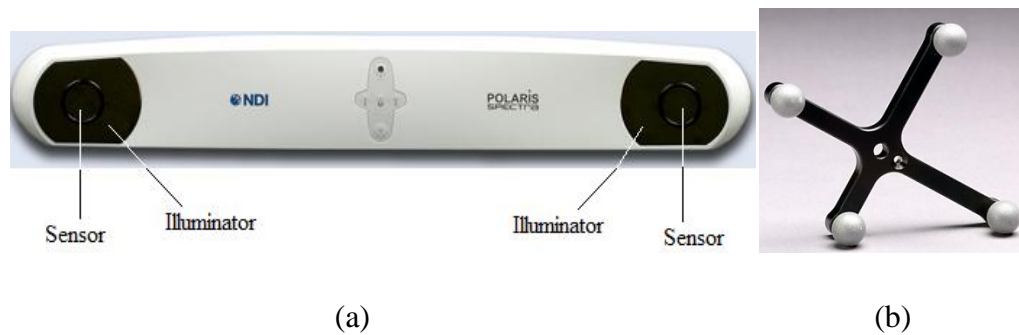


Figure 4.5 Polaris Spectra optical tracking system. (a) Position sensor (front view); (b) Passive tool (Passive markers affixed on the rigid body)[NDI, 2005].

The principle of operation of the Polaris Spectra system is as follows:

- The position sensor emits infrared (IR) from its illuminators and the light floods the surrounding area (Figure 4.6);

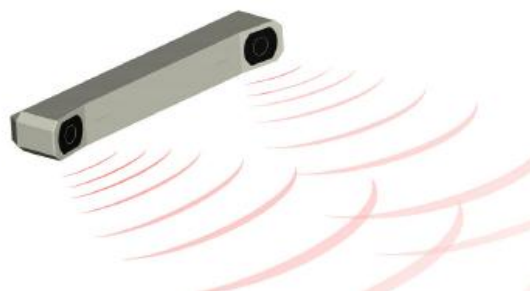


Figure 4.6 Infrared light floods the surrounding area[NDI, 2005].

- Passive markers reflect infrared light back to the position sensor (Figure 4.7);



Figure 4.7 Passive markers reflect infrared light back to the position sensor[NDI, 2005].

- The position sensor receives light from marker reflections, then triangulates the three dimensional position and orientation of a tool to provide 6 degrees of Freedom (Figure 4.8).

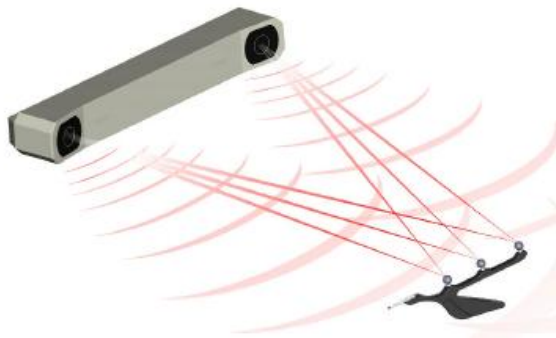


Figure 4.8 Position sensor receives light from marker reflections[NDI, 2005].

- The position sensor transmits the transformation data along with status information to the host computer.

In order to determine the position of an IR source, triangulation is used by the position sensor to derive the position of IR source with respect to the frame of the position sensor (Figure 4.9). Where the lines cross, the system considers the point to be a possible marker position. Otherwise, the point is ignored.

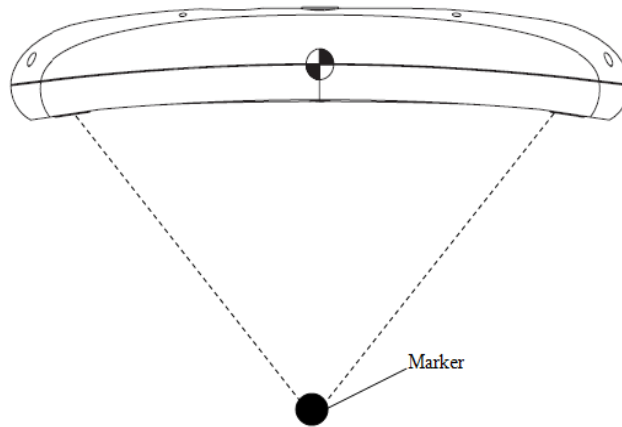


Figure 4.9 Determining a marker position

A global coordinate system is adopted for the position sensor, and the reported orientations and positions of the tools are defined with respect to this coordinate system. The origin of this coordinate system is located at the position sensor and axes are aligned as shown in Figure 4.10.

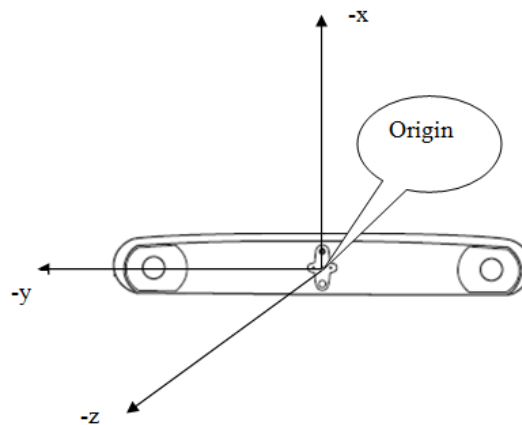


Figure 4.10 Global coordinate system of Polaris spectra.

A pyramid measurement volume is characterized for the system (Figure 4.11). The volume defines the range where data was collected and used to characterize the position sensor. The system accuracy is 0.3mm RMS (root mean square) within the pyramid volume. The accuracy of measurements reported outside the Polaris Spectra characterized measurement volume is unknown. Relative measurements can be made using a reference tool. In this way, a local coordinate system is defined for the tool and the software can calculate and report a transformation with respect to the local coordinate system.

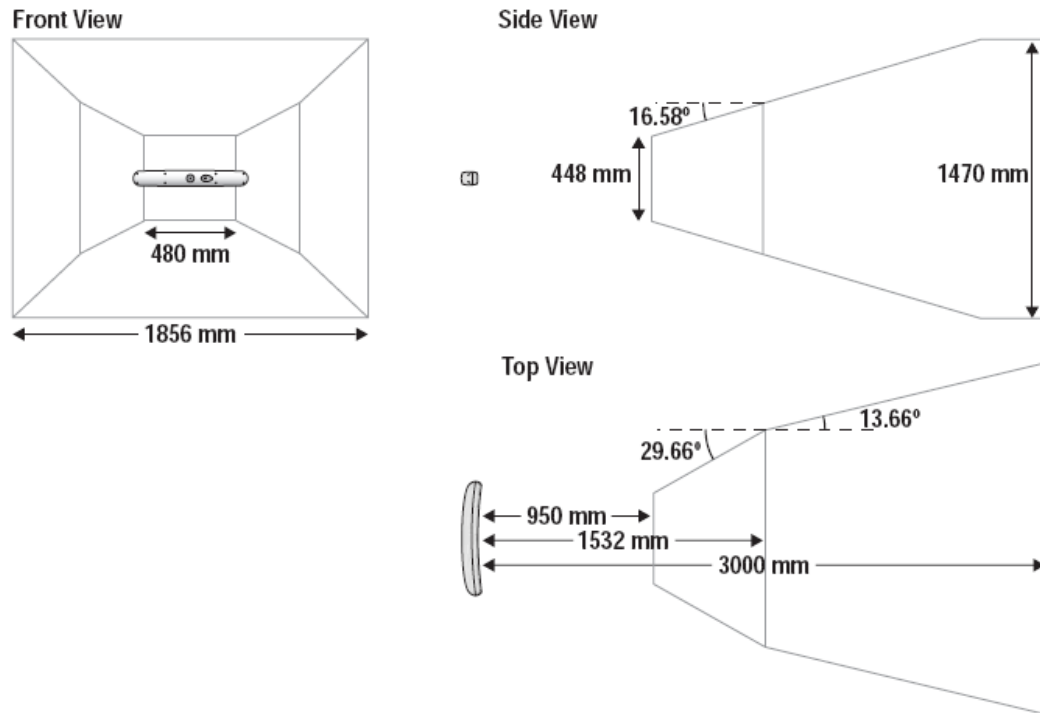


Figure 4.11 Pyramid volume.

The system uses a RS422 interface to communicate with the host computer, and it is controlled using an application program interface (API) which sets commands and parameters to configure and request information from the system. The internal tool transformation update rate of the Polaris system is up to 60Hz.

4.2.2 The utilization of the tool

Either individual markers or tools can be tracked by the system for determining the position. A tool is a rigid body on which three or more markers are fixed with no relative movement between them. Once the system has measured the positions of all the markers, it calculates the segment length (the distance between two markers) for each pair of markers, and the angle between each segment. It compares this data with the marker geometry data in each tool definition file to determine which markers belong to which tool. The unique geometry requirements allow the Polaris system to distinguish between tools. Any markers that are not part of a tool are considered stray markers. A tool is acquired once it has matched the minimum number of markers (a parameter in the tool definition file) for the tool and can calculate a transformation for the tool. Tools (Figure 4.5) are employed in this system for

determining the position rather than individual markers since more requirements have to be met for determining the position and orientation of the tools to achieve accurate measurements. There are several more reasons for this:

- The position sensor can be ‘tricked’ by phantom markers when using individual markers for an application, and this may cause a significant error. Phantom markers are the result of the calculation that the tracking system uses to determine the position of a source of IR. As mentioned previously, in order to determine the position of an IR source, the Position Sensor calculates a line between the source of IR and each sensor (Figure 4.9). Where the lines cross, the system considers the point to be a possible marker position. Phantom markers emerge when more than imaginary line crosses. This generally happens when two or more markers are in the same plane as the sensors. For example, in the case of two coplanar markers, there would be four intersections (Figure 4.12).

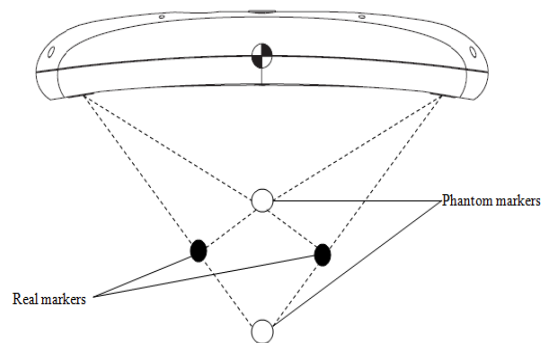


Figure 4.12 Phantom markers and real markers.

- Individual markers can be subjected to interference from external IR sources. One IR transmitter may be recognized as a marker, but it is difficult for external IR source to meet the tool’s geometry data.
- There is no orientation information of an individual marker since no local coordinate system can be assigned to individual markers.

Since more than one tool is used in the tracking scheme, tools must have different marker geometries from each other so that the position sensor can distinguish them. A tool definition file which specified the geometry of the tool's markers, the parameters and settings and some other properties must be loaded into the system before the tracking procedure is initiated. The system cannot precisely analyse the data without this information. Accordingly, different marker geometries along with corresponding definition files are required for the system.

Two tools were designed for this application, and several geometric constraints for designing tools must be satisfied:

- At least 4 markers should be used for each tool. The minimum distance between any two markers should be 50mm within the measurement volume, as the system may not be able to track the tool reliably if the markers are closer.
- The distance between two markers on a tool is defined as segment length. These lengths on a tool must differ by at least 3.5mm. This ensures that the tools have sufficient difference in geometry so that the position sensor can distinguish between them. If the lengths of two segments differ within 3.5mm, these two segments are called like segments.
- Two segments on the same tool comprise a segment pair, and if there are segment pairs on two different tools, each segment in one pair has a like segment in the other, these two pairs are called like segment pairs (Figure 4.13). Any tools with like segment pairs that will be tracked simultaneously must have segment angles that differ by at least 2 degrees. Appendix 1.1 describes how to determine a segment angle.

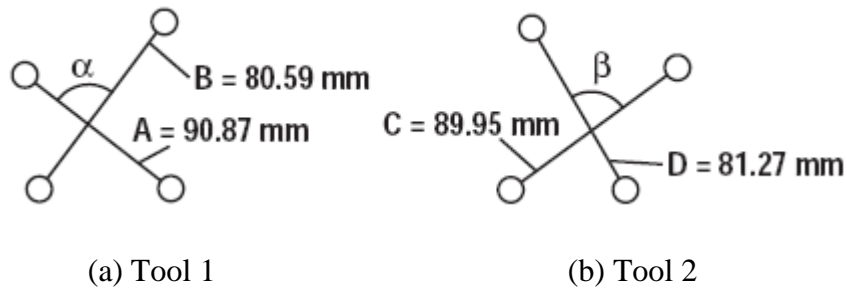


Figure 4.13 Segments A and B form one segment pair, segments C and D form another segment pair. A and C, and B and D are like segments, so segment pair AB and segment pair CD are like segment pairs. In order to track the two tools with like segment pairs together, segment angles α and β must differ by at least 2 degrees.

Once the tool has been designed, the next step is to create a tool definition file for characterizing it. This process can be accomplished using NDI 6D architecture software, which is a software application program that provides an interface to help characterize and create their tool definition files (detailed step to create a file is described in Appendix 1.7). As mentioned previously, a tool definition file contains tool geometric information mentioned above, as well as three tool tracking parameters, which significantly affect the accuracy of the measurement. Because considerable differences in accuracy can be obtained with different parameters even if the tools are identical, these parameters should be carefully considered as follows:

- **Maximum 3D error.** This parameter defines the maximum 3D error for each marker. It is the variation between the measured and actual position of a marker on a tool, and if the measured 3D error of a marker is over the defined maximum 3D error value, the marker is not used by the system to calculate the tool transformation.
- **Maximum marker angle.** This parameter defines the maximum tolerable angle between a marker and each detection camera on the position sensor. A marker ‘normal’ indicates the direction of a marker specified in the tool definition file. The marker angle is measured between the marker normal and each sensor. The default maximum marker angle for passive markers is 90°. Similarly, if the angle between the marker normal and either sensor is over the defined maximum marker threshold, the marker will not be used by the system to calculate the tool transformation.

- **Minimum number of markers.** This parameter defines the minimum number of markers that the system must use for calculating the transformation of a tool. For example, consider a four marker tool with only three markers inside the volume. If the parameter is set to three in the definition file, the system still can report the transformation of the tool. On the other hand, if the minimum number is four, the system will not calculate a transformation and it will report the tool as missing.

According to these constraints and the parameters setting requirement, two four-marker tools are designed and used in this application, as shown in Figure 4.14. (The specifications for the tools given in Appendix 1.2). The Maximum 3D error, Maximum marker angle and Minimum number of markers of the two tools have been set to 0.3mm, 90 degree and 4 respectively.

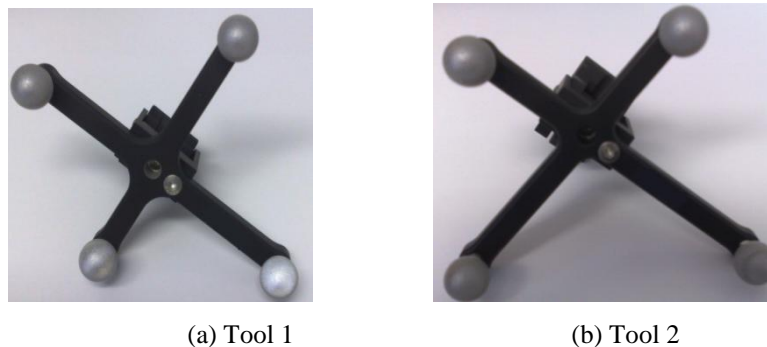


Figure 4.14 Two four-marker tool.

4.2.3 Polaris Spectra calibration

Although the user guide claims that the system volumetric accuracy is 0.3 mm RMS within the pyramid volume, this accuracy may vary because of different tool design. In order to check the accuracy and repeatability of the system with the designed tools, a calibration was carried out. The calibration was conducted by calculating the relative changes of position data. The method is to fix the tool on the spindle of a 3-axis machine tool and move the tool a distance along each specific orthogonal axis, and to record the position measured by Polaris tracking system, and then determine

the variation between them. The following steps were used to confirm the accuracy and the repeatability:



Figure 4.15 Newwall machine with digital readout system.

- A passive tool is mounted on the machine tool spindle as shown in Figure 4.15. The tool was moved throughout a 300*300*300mm working cube, which meets the requirements for the movement of the subject's head during the TMS stimulation procedure. The machine tool is fitted with a digital readout system that has a resolution of 0.001mm, which is 10 times more than the measured accuracy of the Polaris Spectra system.
- The Polaris Spectra position sensor was positioned in the workspace and the tool located in the centre of the working range (Figure 4.16). The tool was moved to pre-determined locations within the working cube. The tools are moved to 4 positions on each axis and the eight corners of the cube, and sufficient samples at each location were recorded. In this test, 30 samples

which are sufficient for the calculation were recorded at each position, and the measurement data were recorded from the Polaris Spectra and the machine tool readout at the same time.

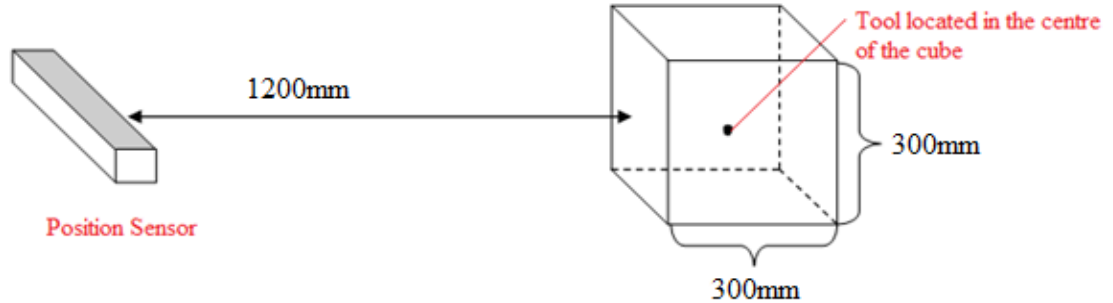


Figure 4.16 Relative positions of the position sensor and the tool.

The accuracy of the system is calculated as the root mean square (RMS) variation of mean of m readings about the true travel (k) across all the locations throughout the working cube. The following are the equations:

ACCURACY_{RMS}

$$= \sqrt{\frac{\sum_{i=1}^{n-1} \left[((x_{i+1_{average}} - x_{i_{average}}) - k)^2 + ((y_{i+1_{average}} - y_{i_{average}}) - k)^2 + ((z_{i+1_{average}} - z_{i_{average}}) - k)^2 \right]}{n-1}} \quad (4.1)$$

where $x_{i_{average}} = \frac{\sum_j^m x_{i,j}}{m}$, $y_{i_{average}} = \frac{\sum_j^m y_{i,j}}{m}$, $z_{i_{average}} = \frac{\sum_j^m z_{i,j}}{m}$ are the average measurements, n represents n locations.

RMS repeatability is calculated as the RMS variation of the m readings about the average of the 3D readings at each location. The equation for repeatability is given by:

REPEATABILITY_{RMS}

$$= \sqrt{\frac{\sum_{i=1}^n \left[\sum_{j=1}^m ((x_{i,j} - x_{i_{average}})^2 + (y_{i,j} - y_{i_{average}})^2 + ((z_{i,j} - z_{i_{average}})^2 \right]}{n * m}} \quad (4.2)$$

following calibration, the estimated RMS accuracy was found to be 0.41mm and RMS repeatability was 0.057mm for both tools.

4.2.4 Utilization of Quaternions to parameterize the rotation

The Polaris Spectra system and the Staubli robot controller use different representation of rotation. The orientation information measured by the Polaris Spectra system is represented using Quaternions [Hamilton, 1844], and Euler angles are adopted by the robot controller. Euler angles are most commonly used to parameterize 3D rotation, which is expressed as three rotations about two or three orthogonal axes such as: Z-Y-Z (roll-pitch-roll) or X-Y-Z (roll-pitch-yaw). A 3*3 rotation matrix is a typical format for representing Euler angles, which can be obtained by multiplying three independent rotational matrices according to the series of three rotations. Although it is easy to understand the process when using Euler angles to parameterize the rotation, there are several disadvantages with this method which may cause significant problems in this system.

The most serious disadvantage is *gimbal lock* [Watt and Watt, 1992], which means that one degree of freedom is lost during the rotation procedure. In this application, the passive tool is tracked with respect to the tracking system frame. In order to display *gimbal lock* clearly, assume the origin of the passive tool frame and the tracking system frame are coincident at the start position (Figure 4.17(a)). First, rotate 45 °about axis Z_p (Figure 4.17(b)), next, rotate 90 °about Y_p , so that X_p and Z_t are coincident (Figure 4.17(c)). After this step, if we rotate -30 °about X_p , the position becomes Figure 4.17(d). The complete yaw-pitch-roll rotation series is Z_p - Y_p - X_p , with the Euler angles (45, 90, -30). However, since the X_p and Z_t are coincident after the second step, the rotations of the first and the final step are about the same axis, and in this way, the same orientation can be obtained by simply rotating 75 °about Z_p . In other word, Euler angles ($Rz1$, 90, $Rx1$) and ($Rz2$, 90, $Rx2$)

has the same effect, if $R_{z1}-R_{x1}=R_{z2}-R_{x2}$. Therefore, one degree of freedom is lost and we have *gimbal lock*.

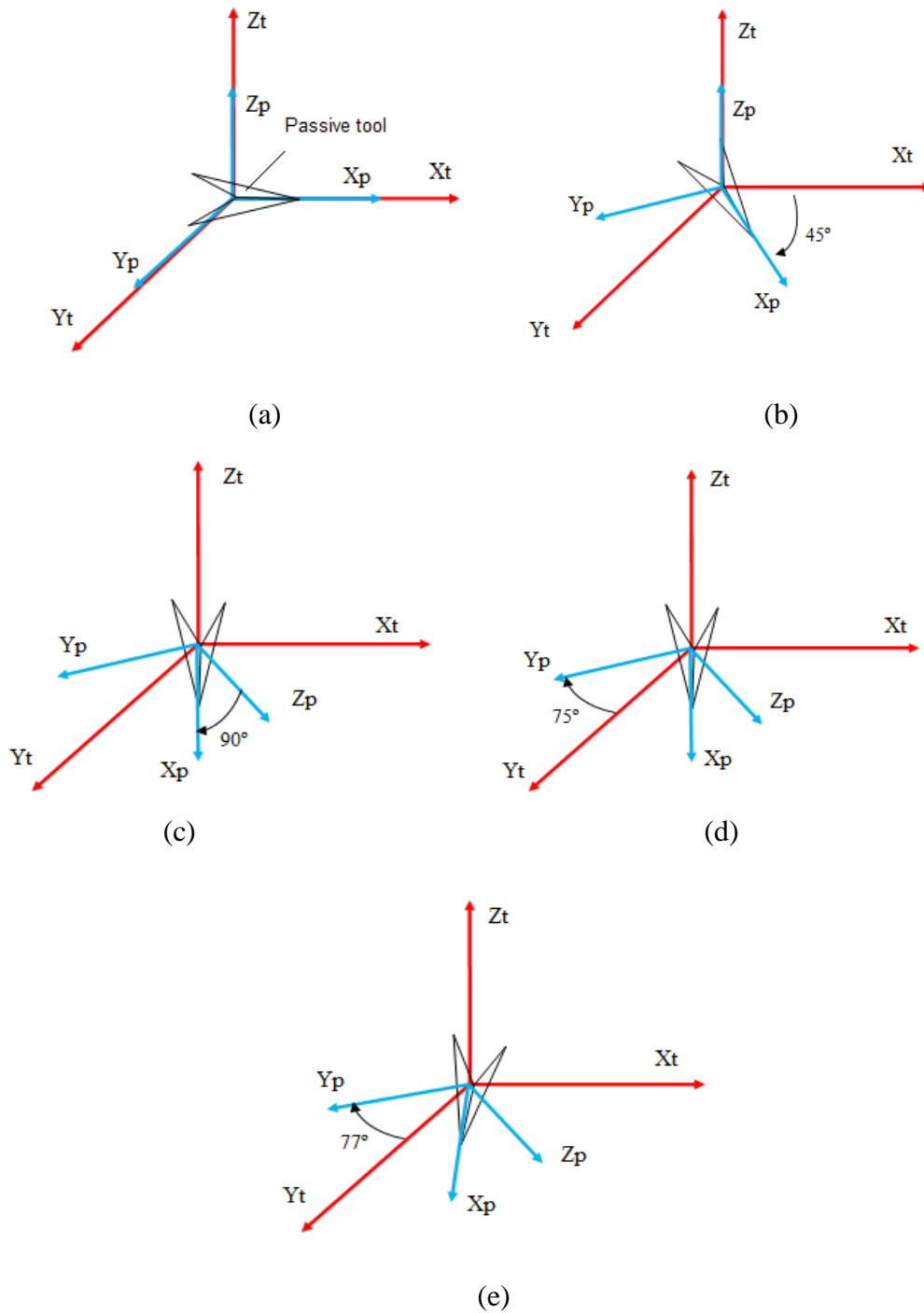


Figure 4.17 The process of inducing *gimbal lock*. The red frame is the tracking system frame and the blue one is the passive tool frame.

The consequence of *gimbal lock* is dangerous, as the Euler angles cannot represent the real orientation in space. For example, from the position in Figure 4.17(d), a rotation of 2° about the Z_p axis, so there should be an angle between X_p and Z_t axis (Figure 4.17(e)) and Euler angle becomes (47, 90, -30). However, if a rotation sequence is carried out with these Euler angles (47, 90, -30), X_p and Z_t axis are coincident. If this happened during the TMS procedure, the subject could be in danger, and thus Euler angles are not suitable for monitoring the subject's movement. For this reason, quaternions are used in the application for the calculations of rotation. (Appendix A1.5 provides an introduction to the properties of quaternions).

4.3 Robot manipulator arm

In the automated TMS application, the robot arm should be capable of supporting and positioning the stimulus coil over a fixed location and at a fixed orientation on the unrestrained head of the subject. Additionally, it should also be able to track small head movements between the subject and coil and compensate the movement within a certain range.

For the robot to meet these challenges, three key requirements have been identified:

- The payload of the robot should over 3kg (weight of coil, sensors and umbilical cords)
- Sufficient motion range with high precision of repeatability
- Real-time path control ability

Table 4. 1 Robot comparison

Model Parameters		Adept s650	<u>Staubli TX60</u>	Denso VS6556G	Kawasaki FS003N
Reach		653mm	670mm	650mm	620mm
Maximal Payload		5kg	9kg	6kg	3kg
Repeatability		±0.02mm	±0.02mm	±0.02mm	±0.05mm
Weight		28kg	51.4kg	35kg	20kg
Motion Range	Axis 1	±170°	±180°	±170°	±160°
	Axis 2	+45°,-190°	±127.5°	+135°,-100°	+150°,-60°
	Axis 3	+256°,-29°	±142.5°	+166°,-119°	+120°,-150°
	Axis 4	±190°	±270°	±190°	±360°
	Axis 5	±120°	+133.5°,-122.5°	±120°	±135°
	Axis 6	±360°	±270°	±360°	±360°
Controller		(called <u>Smartcontroller</u>) Ethernet TCP/IP interface (100Mbit/s)	(CS8C) Ethernet (100Mbit/s), field bus interface	(RC7) Ethernet	(D42) Ethernet, <u>Profibus-dp</u>
real-time path modification (ALTER)		YES (requires V+ Extensions) update rate 16ms	YES (VAL3 programme language) update rate 4ms	No	No
Noise		NO figure for this	NO figure for this	NO figure for this	NO figure for this

4.3.1 General characteristic of Staubli TX60 robot arm

Four different 6 degree of freedom robot arms were compared, and their main parameters are summarized in Table 4.1. After the comparison between these parameters, it can be found that the Denso and Kawasaki cannot meet the third requirement (real-time path control) mentioned above so that both of them are not suitable for this application. Both Staubli TX60 and Adept s650 meet all the requirements, however, the Staubli has an increased motion range and payload, moreover, it has a much faster update rate for the real-time path control(4ms vs 16ms). Thus, Staubli TX 60 was chosen for the application due to its better overall performance, although it is considerably heavier. The configuration and performance of the robot arm and control are shown in Appendix A 1.3.

4.3.2 General characters of Staubli CS8C controller

The Staubli CS8C controller (Figure 4.3) is a multi-processor system, including emergency and safety stop channels, basic inputs/outputs, field bus interface (DeviceNet, Profibus, CANopen, ModBus TCP client), Ethernet link, serial port and programming logical controller (PLC). The field bus interface and PLC are not originally installed on the controller, extension parts are needed for their usage. The processor controls the robot arm through digital power amplifiers dedicated to each axis of the arm, and serial ports, Ethernet link and field bus are available for communication with the controller. The robot is programming using ‘high-level’ VAL3 language and it also can be programmed using IEC61131-3 standard PLC languages such as: IL, SFC, FBD LD and ST when a PLC is integrated into the control system. The Control instructions can be sent to the controller via the MCP (Manual Control Pendant) or from a PC connected via the Ethernet link. The software development environment on the PC is called Staubli Robotics Studio (SRS) which contains all the tools available to develop and maintain robotics applications. There are three working modes for controlling the robot: manual mode, local mode and remote mode. In this application, local mode was selected and the robot is programmed using VAL3 language, in which the PC controller connects to the robot controller via Ethernet and after acquiring and processing the data from sensors, the PC transmits the real-time calculated coordinates to the robot controller. The VAL3

program then control the robot motion according to these coordinates information. More details of the CS8C controller are provided in Appendix A 1.3.

4.3.3 Ethernet communication

As mentioned above, a 100Mbit/s Ethernet communication via a twisted pair network cable was chosen for its higher bandwidth, since data transfer rate are critical for real time control of the application. Ethernet also makes the system more flexible for the further extension, since the controller can be connected to a local area network (LAN) via Ethernet. The content, reliability, size and speed of a network communication rely on a protocol. The term TCP/IP actually refers to the Internet Protocol Suite which is set of communications protocols used for the Internet and other similar networks. It is named using the Internet Protocol (IP) and Transmission Control Protocol (TCP), as they are two of the most important protocols in this standard.

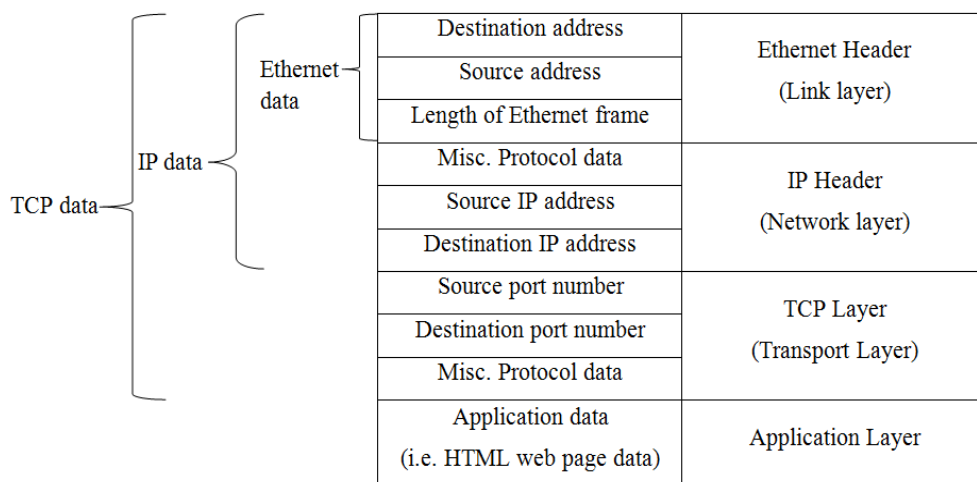
Table 4.2 Five-layer TCP/IP protocol

Layer 1	Physical	RS-232, 10BASE-TX,100BASE-TX
Layer 2	Link	Network interface and device driver
Layer 3	Network	IP,ICMP,IDMP
Layer 4	Transport	TCP,UDP
Layer 5	Application	Telnet,FTP,RCP,etc.

Table 4.2 demonstrates a five-layer TCP/IP protocol. The lowest layer is the physical connection with the network hardware like the network card, cable etc. The Link Layer contains communication technologies, it is responsible for decoding the signals from the network wires, passing this data to the network layer, and encoding data from the above layer. This is where the device driver resides. The Network Layer provides communication methods between multiple links of a computer and facilitates the interconnection of networks. It primarily contains the Internet Protocol,

which defines the fundamental addressing namespaces used to identify and locate hosts on the network, and decides where to send the data. This layer makes no guarantee the data will get there or not, only where to send it. The transport layer provides transmit of data flows for the network using different protocols (TCP, User Datagram Protocol (UDP)); and this is where a guarantee of reliability may be made. The application layer is the highest layer, and is where the user typically interacts with the network. This is where the web browser, telnet client, IRC, FTP, or other software resides.

Table 4.3 A sample Ethernet frame



The unit of data on a network is called a ‘packet’, which contains both the headers and the data. Table 4.3 illustrates a sample Ethernet frame. This is a packet that has gone from the link layer, all the way to the application layer. It can be seen that each layer treats almost all of the information from the previous layer as data, just adding its own headers which contain a combination of checksums, destination and origin address, protocol identifier and state information. Figure 4.18 illustrates a FTP connection via Ethernet, which is a typical example of TCP/IP application. Once the client computer has finished the connection to an FTP server via an Ethernet LAN, the client can specify information in the application layer. This information is then passed down to the transport layer, where a header is added. After this, it is passed to the network layer and encapsulated in an IP frame. Finally, it reaches the Link layer that transmits it across the physical layer. On the other end of the cable, the server acquires the information through the Ethernet card and driver, and the data is passed

up to IP, then TCP and finally to the application layer, the FTP program in the server. At each layer a new communication may be generated, which can only be understood by the same layer in the other machine. For example, if the TCP layer detected an error in the data received, it will transmit a request back to the other machine to re-send the packet without passing the bad data to the FTP layer.

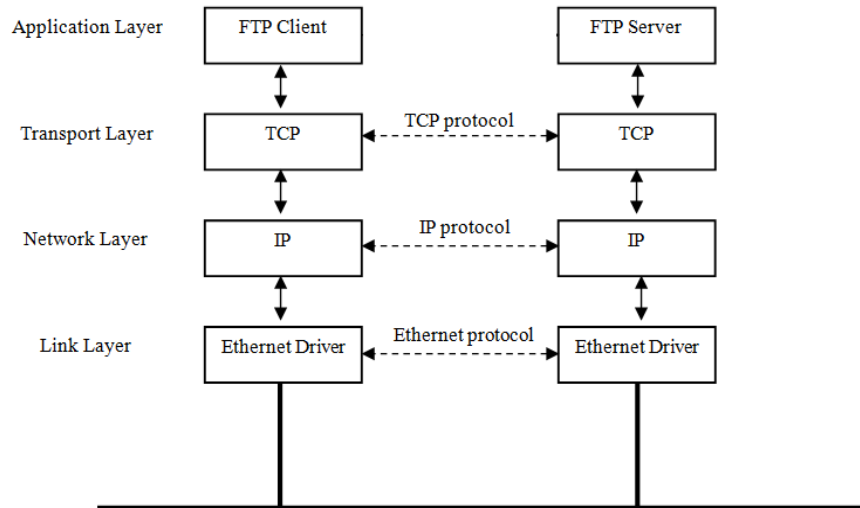


Figure 4.18 FTP ‘client-server’ connection via Ethernet.

This application uses the same method for communication with the Staubli CS8C controller, the only difference is in the application layer as illustrated in Figure 4.19. The Client is the QNX PC controller and the Server is the CS8C robot controller. The client and server programs are developed on these two controllers accordingly. The ‘binary data’ type is employed to encapsulate the command data transmitted from PC to CS8C, as this instruction type increases the performance of the controller by reducing encoding and decoding time. The IP address of the QNX is set to ‘192.160.0.10’ and the CS8C is ‘192.168.0.254’.

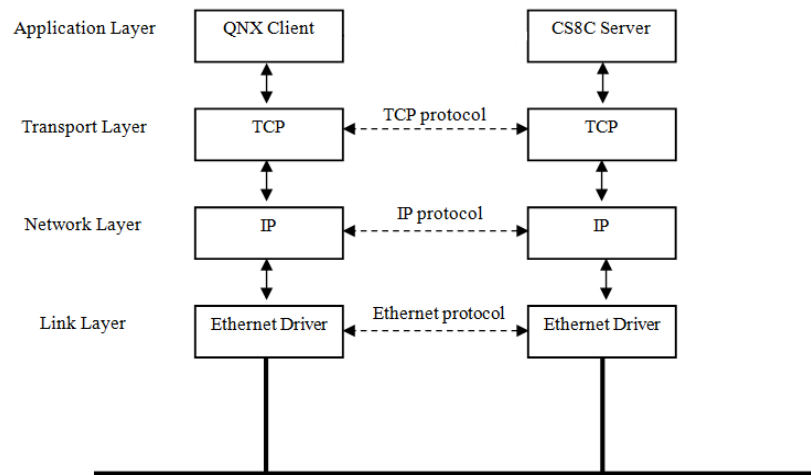


Figure 4.19 connection between QNX and CS8C controller via Ethernet.

Internet socket is used to establish the Ethernet communication between these two controllers. More detail on the socket connection is provided in Chapter 7. The remainder of this section concentrates on checking the data update rate between the two controllers once the connection has been established. The CS8C controller is multi-tasking in order to run the real-time path control instructions, so that the update rate is equivalent to the time period between these synchronous tasks. In order to obtain an optimal time period, a series of tests were conducted to check the timing of data transmission between the PC and the robot controller. Figure 4.20 outlines the test procedure. Although time could be saved if the robot controller does not reply to the PC during the data transmission, in this application, the robot controller is required to reply to the PC request once it completes receiving the data in each loop to guarantee the communication between the control PC and the robot controller is keep connected. The tests were carried out by simulating the data transmission in the real environment, which involved the following setup procedure:

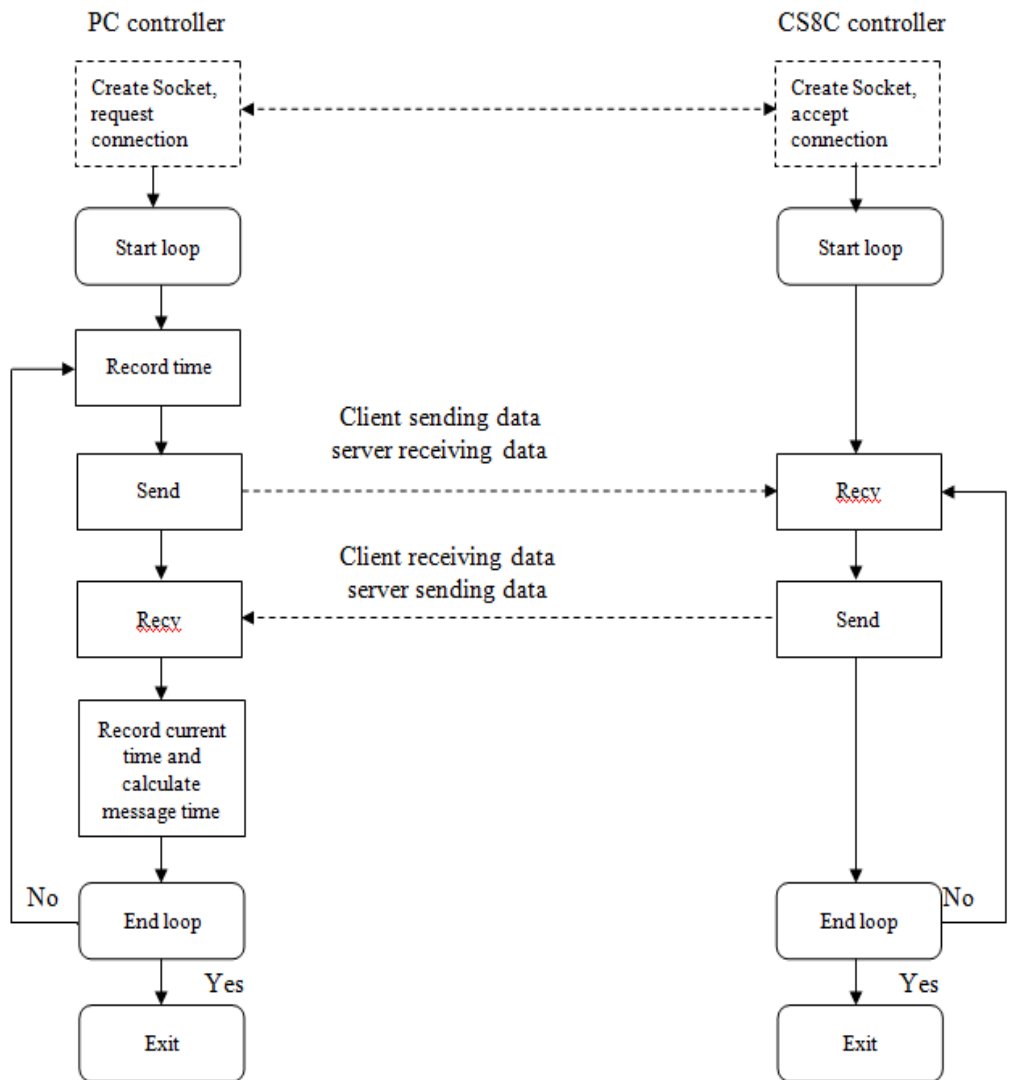


Figure 4.20 Timing test for data exchange.

- Establish the same data update rate on both the PC and the robot controller, for example, 4ms.
- Create an independent QNX thread which can record the actual time periods during the data transfer process to ensure there is no interference from the threads that carry out other system tasks. Multiple threads are employed in this application to run several different tasks in parallel, which more efficiently use the resources of the QNX system and improve the real-time performance. For example, during this test, both force and position sensor, time counting and data exchange are controlled using different threads: *forcethread()*, *positionthread()*, *timingthread()* and *datathread()*, so that they

run independently. The time period is defined as the time consumed for one cycle of the data exchange between the PC to robot controller.

- Run the system as if it were performing a real TMS experimental procedure, and record the time periods. All the control threads on the PC are running, to ensure the test carried out under real conditions, and the tasks on robot controller also executed. The passed message consisted of a data field of 50 bytes, as the maximum data sent from the PC will not exceed this.

The test was established to send data from the PC to the robot controller using a 4ms update rate. A number of time periods have been recorded (Figure 4.21). It can be seen that the actual time intervals only vary by $1\mu\text{s}$, thus making for a stable communication. This clearly shows that the actual time intervals are stable, and the 4ms update rate is achieved. Hence, the transmission rate of 4ms was adopted as it is the highest speed that can be achieved by the robot controller.

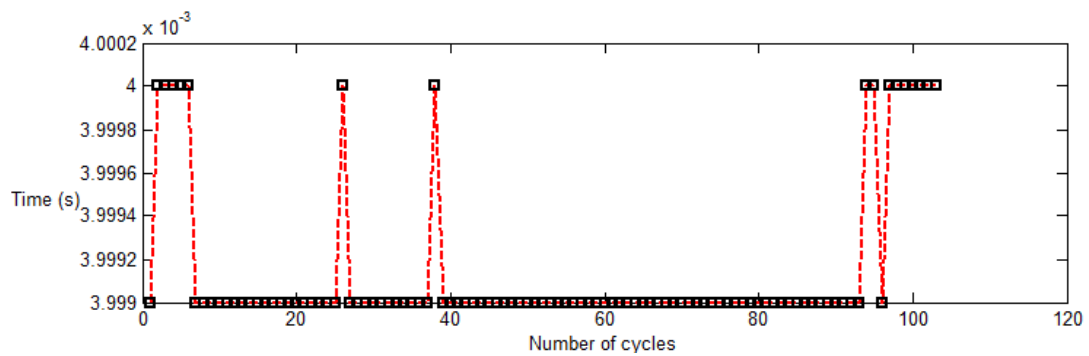


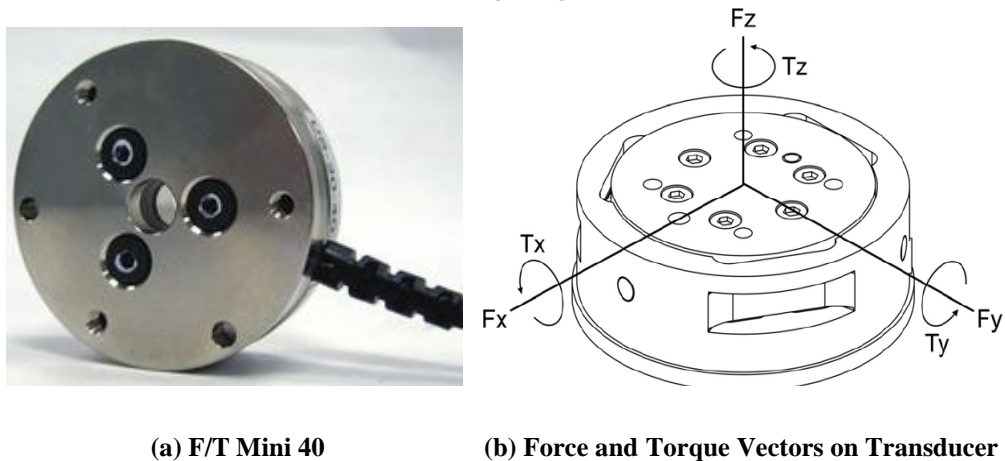
Figure 4.21 Time intervals with 4ms setup.

4.4 Force sensor systems

An ATI F/T force transducer and a pneumatic collision sensor are incorporated in the system; both are mounted between the robot tool flange and stimulation coil. The force sensor is used to monitor and regulate the contact force between the subject's head and coil in real-time. The pneumatic collision sensor is employed in the system to detect a mechanical overload if the robot encounters an obstruction, and can generate a signal to stop the robot and absorb the crash energy when an unpredictable error occurs with the force/torque transducer.

4.4.1 ATI force/torque sensor system

The force sensor system comprises an ATI Mini40 F/T transducer, an ATI DAQ interface power supply (DAQ-IFPS) (ATI, US) and a PowerDAQ PDL-MF (UEI, US) data acquisition system, as illustrated in Figure 4.23. The transducer is a six axis force and torque sensor that can measure the force and the torque about X, Y and Z direction (Figure 4.22).



(a) F/T Mini 40

(b) Force and Torque Vectors on Transducer

Figure 4.22 Applied Force and Torque Vectors on Transducer.

The transducer is a compact, rugged, monolithic structure that converts force and torque into analog strain-gauge signals. These analog signals are then amplified by the DAQ-IFPS and converted by the data acquisition system into 16-bits resolution at 500Hz. The PC then receives these transducer load information through the DAQ driver, and further converted to usable force and torque data using F/T software and transducer calibration data. A power supply unit is used to supply the power to the transducer, and raw power is converted to $\pm 5\text{VDC}$ by the DAQ-IFPS before it goes to the transducer.

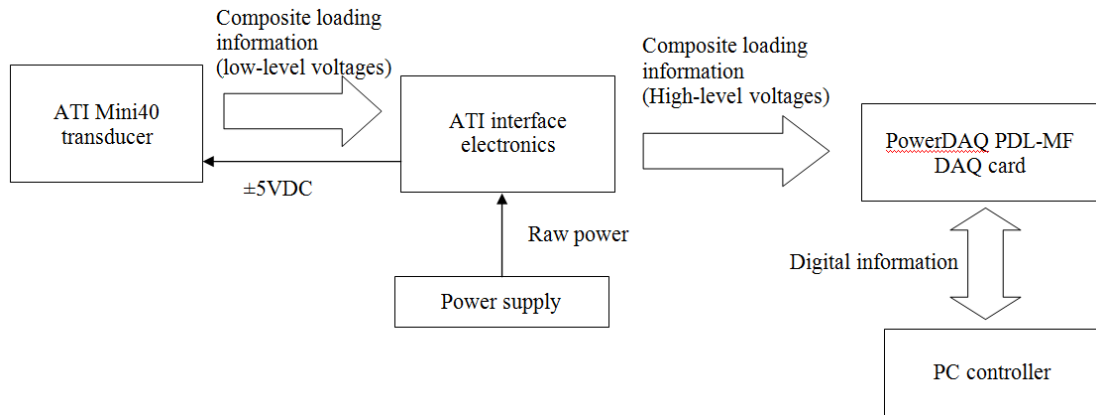


Figure 4.23 Outline of the force system[ATI, 2010].

The PDL-MF DAQ card is a 50k Samples/s, 16-bit PCI multifunction data-acquisition card. It provides 16 single-ended or 8 differential A/D channels, two 12-bit analog outputs, 48 digital I/O lines and three 24-bit counter/timer lines. It connects with the ATI interface circuit using differential signal connections (Figure 4.24), to provide common-mode rejection and can achieve better results than single ended connection.

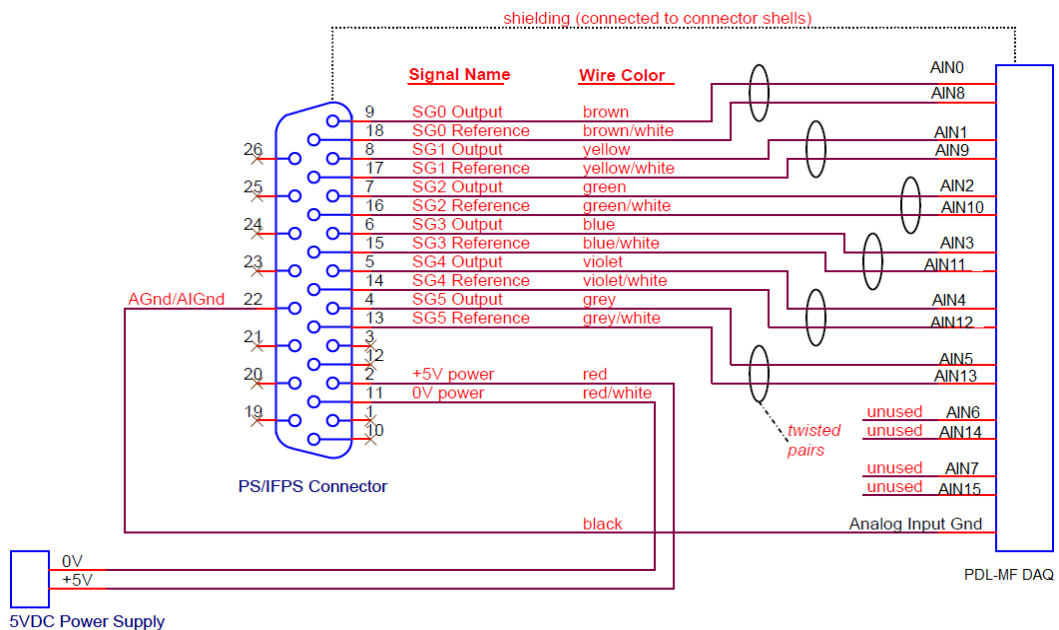


Figure 4.24 Differential Connections to a Data Acquisition System[ATI, 2010].

Figure 4. 25 shows the sequence to obtain the force and torque data. A calibration file which contains the specified transducer stiffness calibration matrix is loaded to the system, and it must match the serial number of the transducer. The DAQ system is then configured by setting up the sampling rate (500Hz), number of channels (6 channels). Once the configuration is complete, biasing is carried out if needed, which is used to store a voltage reading to be subtracted from subsequent readings and effectively "zeroing" the transducer output to remove tooling weight. Finally, the digital data from the data acquisition system is converted into forces and torques using ATIDAQ 'C' library.

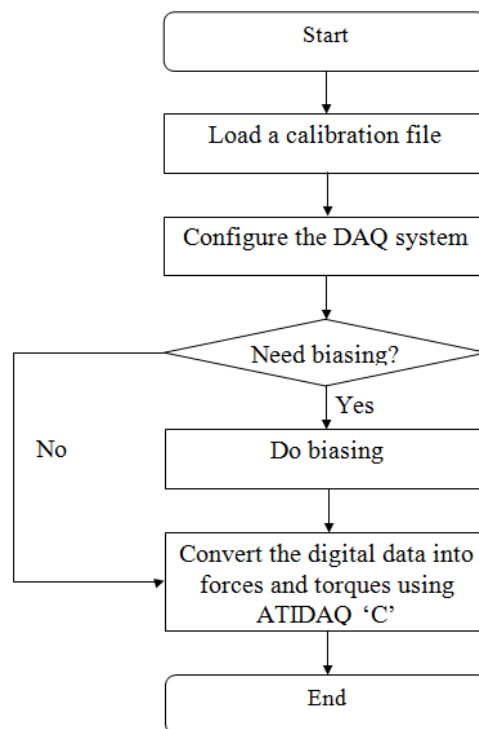


Figure 4. 25 Steps to obtain the force and torque.

Tests were carried out to check the output of the force sensor with the robot arm switched on and off respectively. Samples of biased force and torque data were recorded without any loading on the sensor. Figure 4.26(a) illustrates the sensor output with the robot arm switched off. The root mean square (RMS) of the force error in the X and Y axes is about $\pm 0.03\text{N}$, whereas in the Z axis is about $\pm 0.1\text{N}$. This is because the sensitivity of the force sensor in Z axis is triple of the other axes (Appendix A1.6.3). Slight effect was introduced by switching on the robot arm

(Figure 4.26(b)) especially in Z axis. The maximum RMS in Z axis is about $\pm 0.14\text{N}$. There is no clear difference found in the torques between these two conditions, as the sensitivity of the torque is much smaller than the force.

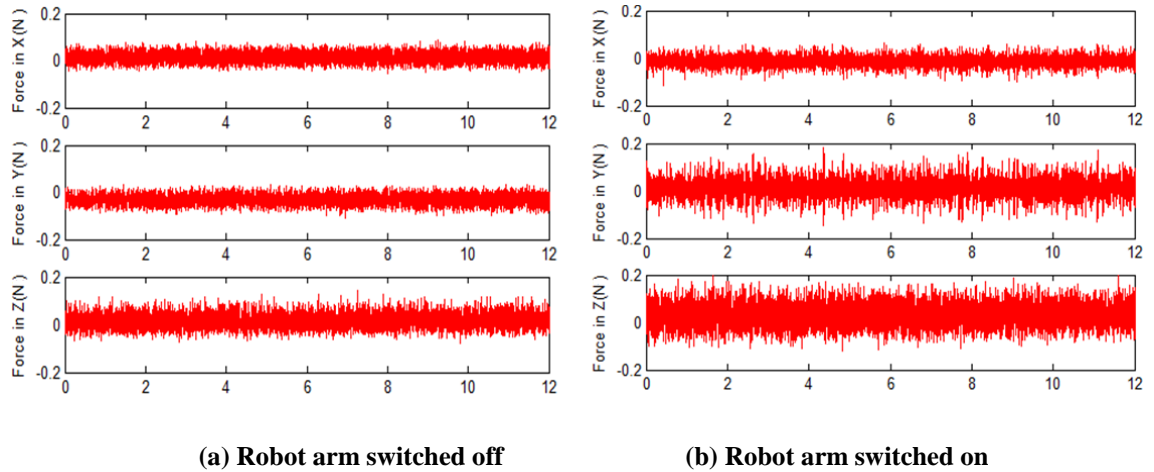


Figure 4.26 Force and torque data without loading.

4.4.2 Pneumatic collision sensor

A pneumatic collision sensor (RAD's Ultimatic™, Module U4618) is employed in the system for mechanically detecting if the robot has encountered an obstruction (Figure 4.27). It can generate a signal to stop the robot and absorb the impact energy if a fault occurs with the robot, or the subject's head collides with the coil.

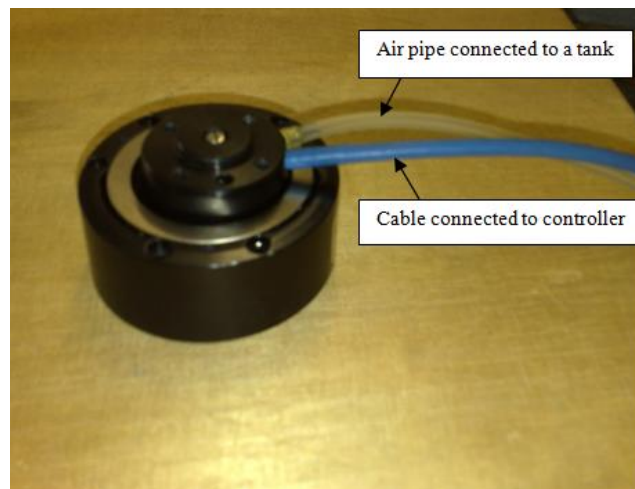


Figure 4.27 Pneumatic collision sensor (RAD's Ultimatic™, Module U4618).

There are several advantages in using pneumatic collision sensors:

- The force (and torque) resistance of the pneumatic collision sensor can be easily adjusted using an air pressure regulator to set different load levels as the different application.
- The switch sensitivity of collision sensor is also adjustable, with the amount of deflection determined by the user according to the system requirements. The switch sensitivity can be set from 0.64mm axial compliance to 2.5mm.
- It also can be used as compliance device rather than a collision sensor (Figure 4.28). A mechanical piston absorbs the crash energy without releasing air from the unit, which prevents tooling sag caused by loss of air pressure and also prevents compressed air from being released into the environment.

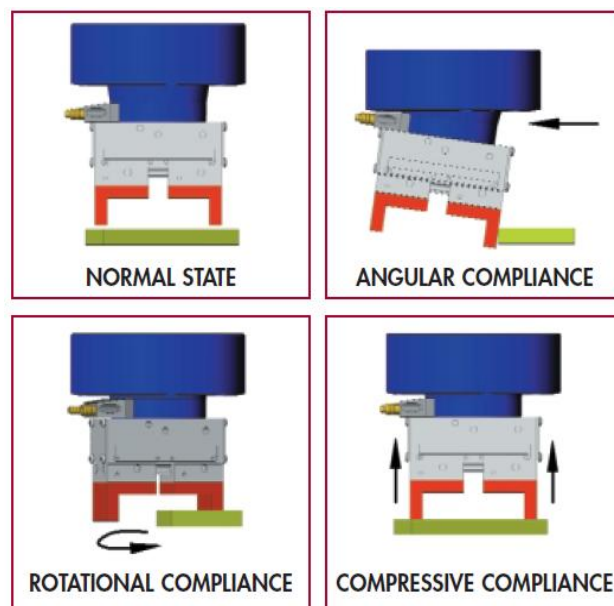


Figure 4.28 Compliance adjustment features [RAD, 2009].

The operating principle of the collision sensor is to establish the minimal points of contact required to restrain all degrees of freedom in normal operation. When a collision occurs, external forces upset this balance thereby allowing angular (rotational), or compressive compliance. This motion opens a normally closed switch and transmits a signal to the robot's E-stop or controller. A cross-section

through the sensor is illustrated in Figure 4.29. A piston provides resistance and crash energy absorption; if the force applied on the piston exceed the resistance, the normally closed switch inside the sensor opens and an E-stop is triggered within 2-6 ms.

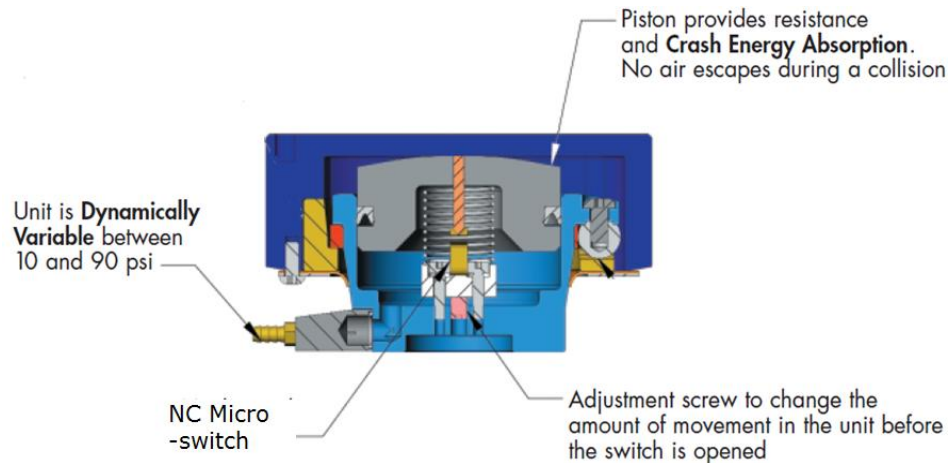
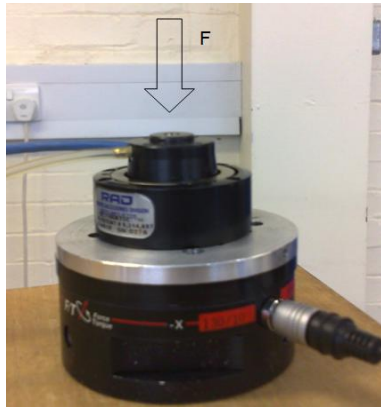
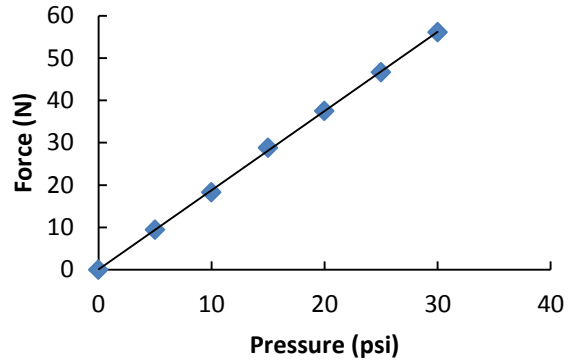


Figure 4.29 Cross section through the collision sensor [RAD, 2009].

In this application, the switch cable of the sensor is connected to an E-stop of the robot controller, and the air line (Figure 4.27) is connected to an air reservoir via a precision pressure regulator. The output of the regulator is adjustable between 0 to 30 psi. The collision sensor was calibrated and the relationship between the breakaway force and air pressure was established. The collision sensor is attached on the ATI Delta (F/T3305) force transducer with no relative movement between them. The force was applied along the Z axis of the force transducer, in the direction shown in Figure 4.30(a). When the pneumatic sensor reaches its limitation, the internal switch is opened and the force sensor stops the force recording. The results are shown in Figure 4.30 (b), it can be seen that the relationship between the compressive force and pressure is linear. A 10N limitation is set for the contact force between the coil and the subject's head; as the TMS coil is approximately 2kg (20N), the regulator is pre-set to 17psi to give a breakaway force of 30N.



(a)



(b)

Figure 4.30 Calibration of the pneumatic collision sensor. (a) Configuration for calibrating the regulator and pneumatic sensor. (b) The relationship between the pressure and the compressive force.

4.5 Definition of the tool centre point

The TCP (Tool Centre Point) is defined on the specific tool and the motion of the manipulator coordinated to move this point through space. Normally, a TCP is defined as the transformation from the tool flange frame to the tool centre frame, so the robot movement and positions are made relative to the tool centre frame.



Figure 4.31 Assembly of robot end-effector.

In this application, a F/T transducer, pneumatic collision sensor and the TMS coil (figure-of-8) were mounted on the robot as shown in Figure 4.31. Since precise placement the coil is essential for the success of the TMS stimulation, the accuracy of the calibration between the TCP and the tool flange need to be carefully addressed. The TCP can be mathematically expressed as a

transformation between a frame assigned at the coil centre and the tool flange. As shown in Figure 4.32, the origin is defined at the geometric centre of the figure-of-8 where the maximum magnetic field is induced. The coil is clamped to the F/T sensor with the coil frame aligned to the tool flange frame. Three TCP definition methods are introduced in the following section.

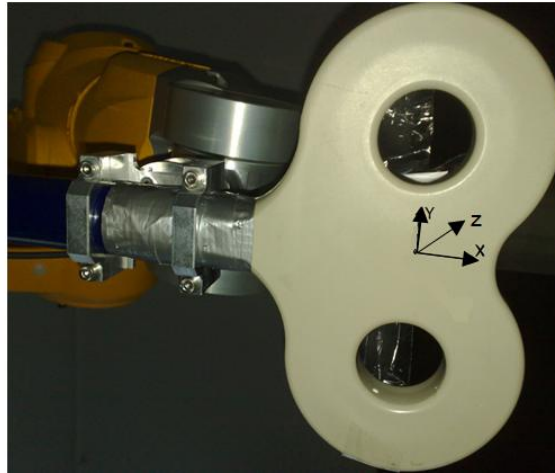


Figure 4.32 Left hand law is used to defined coordinate system (view from beneath of the coil). The X-axis of coil frame aligned with the handle of the coil.

4.5.1 TCP definition by teaching a reference point

The most common method used to define the TCP is by teaching a reference point using the following steps:

- A physical pointer whose length is accurately known is used to teach the reference point.
- A reference point should be defined in the workspace. This point needs to be carefully selected in order to easily identify the major axis (x-y-z) directions so that the resultant tool transform can be aligned properly.
- Mount the pointer to the tool flange and calculate the tool transformation H_p^T .
- Teach the reference point by applying the transform between the tool flange and pointer. The transform between robot base and the reference point H_p^B can be obtained through H_p^T and H_T^B (Figure 4.33).

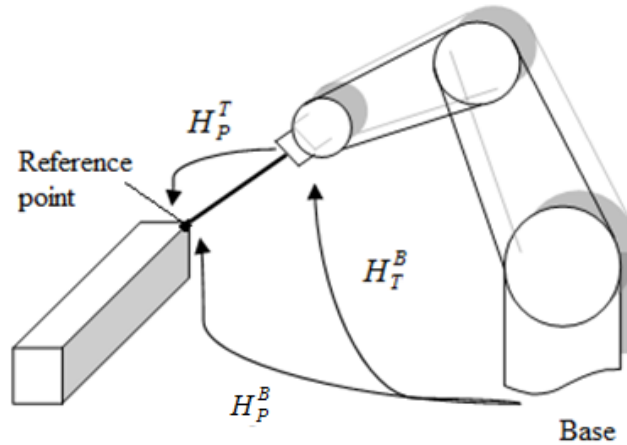


Figure 4.33 Transformations after teaching the reference point.

- Remove the pointer and mount the coil to the tool flange. Place the coil and ensure the TCP of the coil is at the reference point (Figure 4.34).

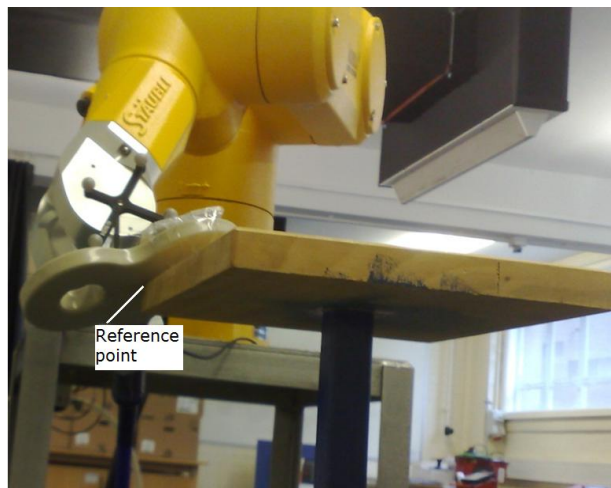


Figure 4.34 TCP of the coil make a contact with the reference point (the corner).

- Finally, the TCP transform can be calculated using Eq. (4.3).

$$H_P^T = (H_T^B)^{-1} * H_P^B \quad (4.3)$$

where H_P^B is obtained by the pointer tool and H_T^B can be obtained by the robot forward kinematics.

4.5.2 TCP definition using sphere equation

The distance along the tool X-Y-Z axes between the robot tool flange and the tool centre is constant, if the tool centre is considered as the origin of a virtual sphere. The robot flange is always a point on the surface of this sphere. In this way, the position offset of the robot flange and the tool centre point can be calculated by solving the following equation:

$$(x-a)^2 + (y-b)^2 + (z-c)^2 = r^2 \quad (4.4)$$

where a, b, c are the origin of the sphere (position of the tool centre) and r is the radius.

The equation can be easily solved by teaching at least 4 non-linear points. This is done by jogging the TCP contact against a fix point such as the tip of a nail with 4 different orientations. The movements should be made with respect to the tool frame, which ensures the movement is on the surface of the sphere. This method calibrates the tool centre without teaching any reference points and it is somewhat easier to carry out than the first method.

4.5.3 TCP definition using the tracking system

In the third method, the TCP is defined using the Polaris tracking system. A special ‘pointer’ tool (Figure 4.35) is used to define the TCP with respect to the position sensor frame. A coordinate system is assigned to the tip of pointer. In this way, the tip can be used to teach the coordinates of a point. To carry out the TCP definition using this method, the transformation from the position sensor to the robot base frame needs first to be established. More information on how to calibrate the position sensor with robot base coordinate system is given in the following chapter



Figure 4.35 Definition the TCP with a pointer.

There are several steps to describe the TCP definition:

- Carry out the coordinate system registration between the position sensor and robot base coordinate system H_B^T .
- Record the coordinates of the coil centre using the position sensor by pointing the tip of the pointer at the coil centre, from which the transformation H_B^T can be obtained.

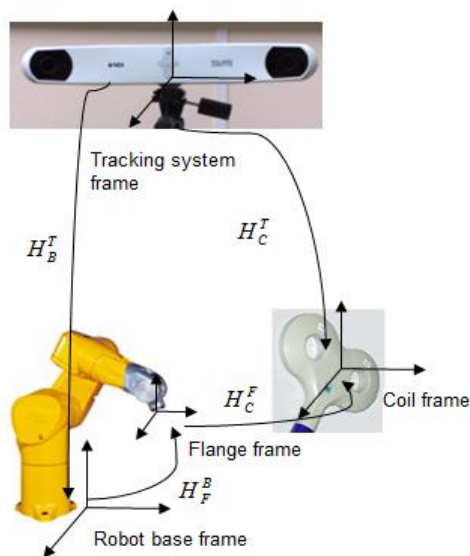


Figure 4.36 Transformations used for defining the TCP.

- Transformation from the flange frame to coil frame H_C^F is calculated using H_B^T , H_C^T and H_F^B (Figure 4.36).

The calibration was carried out using all three methods by repeating each test twenty times to obtain a mean and standard deviation, as illustrated in Figure 4.37.

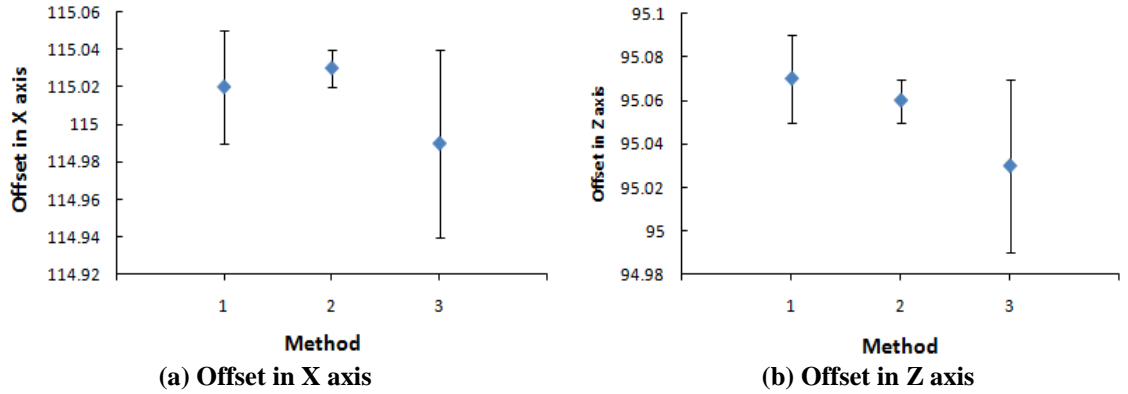


Figure 4.37 Calibration results using the three methods.

The recorded differences between the three methods were less than 0.05mm in both X and Z axes, but the technique using the sphere equation is the most straightforward way of defining the TCP, and for this reason it was employed in this application.

4.7 Summary

This chapter has presented the design of the robotic TMS system, and two major modes of control: Pre-operative acquisition and Tracking mode. A Staubli TX60 robot arm is adopted for manipulating the coil and an optical position tracking system (Polaris Spectra) is used to monitor the position and orientation. A 6 axis ATI F/T mini40 force system is employed for force sensing and a PC is used to collect and process the force and position data from the sensors, and transmit them to the Staubli robot controller via an Ethernet link, which has an update rate of 4ms. Passive reflective tools are designed and used along with the position sensor in the system, with an accuracy of 0.4mm. The TCP is defined using sphere equation as it provides the best accuracy.

CHAPTER 5

COORDINATE SYSTEM REGISTRATION

This chapter focuses on coordinate system registration. In order to track and compensate the movement of a subject's head, the position and orientation of the subject's head with respect to the robot base/tool coordinate system should be determined. A Polaris tracking system is used to establish the robot and subject's head coordinates system, and involves three key steps:

- Establish the relative position and orientation between the coordinate system of the passive tool which is mounted on the end-effector and the robot arm tool centre coordinate system.
- Use the passive tool to define the subject's head coordinate system so that the head movement can be tracked.
- Determine the relationship between the robot base/tool coordinate system and the subject's head.

5.1 Hand-eye calibration

The calibration methodology used in this application is derived from hand-eye calibration in visual servo systems, as illustrated in Figure 5.1. Hand-eye calibration is normally the calibration between the end-effector of a robot arm and a camera, or cameras mounted on it.

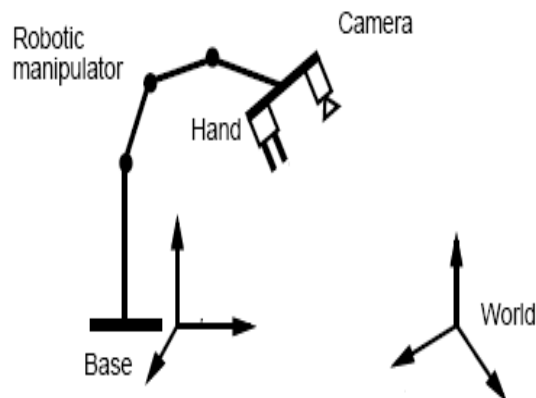


Figure 5.1 Hand-eye configuration of a visual servo system [Dornaika and Horaud, 1998].

Since it is not feasible to obtain an accurate relationship between the gripper and camera by direct measurement, a mathematical method yields a matrix equation of the form $AX = XB$ is proposed by Shiu and Ahmad [1987], where A describes the position and orientation of the coordinate system which is normally assigned to the robot wrist relative to itself after an arbitrary motion. B describes the position and orientation of the camera coordinate system after an arbitrary motion, and X is the transformation between the camera frame and the relative robot link frame (Figure 5.2). In this way, the relative position and orientation between the robot hand and the camera or between the robot base and the camera can be mathematically calculated by solving the matrix equation: $AX = XB$. Various generation, extension base on the equation were presented afterwards [Daniilidis, 1999, Dornaika and Horaud, 1998, Strobl and Hirzinger, 2006].

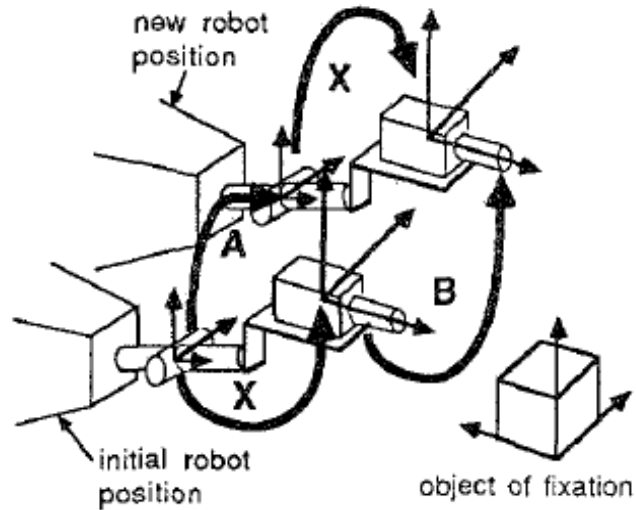


Figure 5.2 Finding the mounting position of a camera by solving a homogeneous transform equation of the form $AX = XB$ [Shiu and Ahmad, 1987].

5.2 Implementation of hand-eye calibration

The method proposed for coordinate systems registration in this study is inspired by the hand-eye calibration method used in visual servo systems, and as there is no camera system used in this application, the Polaris tracking system is an equivalent component used for guiding the motion of the robot and locating an object in the space. Because of its size, the position sensor cannot be mounted on the hand of the robot, in which case the passive tool is mounted on the coil which has a local coordinate system assigned to it. In this way, the tracking sensor is fixed in the robot work space, and the relationship between the robot base and position sensor frame can be determined after the calibration between the passive tool and coil frame.

Figure 5.3 demonstrates the coordinate frames used in the application. The following is a list of definitions for the homogenous transformation matrix:

H_C^B defines the coordinate transformation of the coil coordinate system relative to from the fixed robot base coordinate system.

H_p^C defines the coordinate transformation of the passive tool local coordinate system relative to the coil coordinate system.

H_T^P defines the coordinate transformation of the tracking system coordinate system relative to the passive tool local coordinate system.

H_B^T defines the coordinate transformation of the robot base coordinate system relative to the tracking system coordinate system.

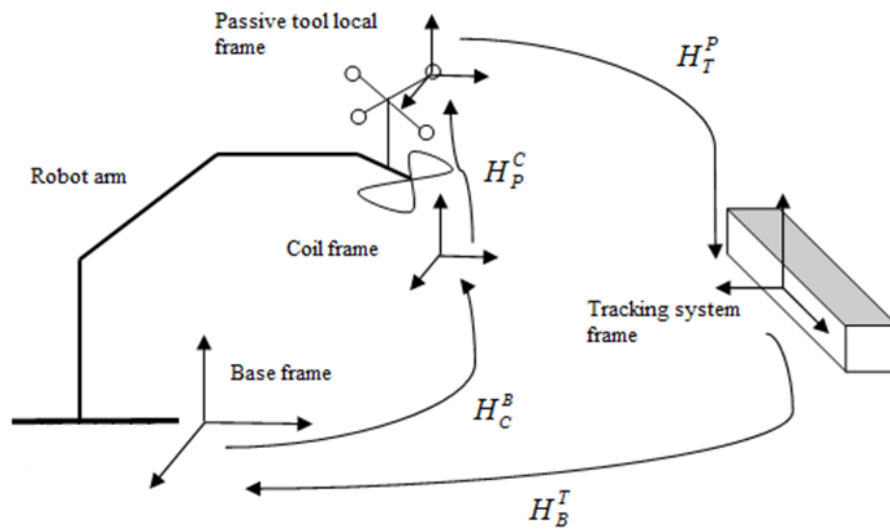


Figure 5.3 All coordinate systems and transformations used in the calibration process.

For these coordinate systems and the transformations illustrated in Figure 5.3, H_C^B can be obtained by computing the robot's forward kinematics, H_T^P can be measured by the tracking system, and H_P^C and H_B^T are the required transformations. According to the Shiu-Ahmad matrix equation $AX = XB$ described previously, in this application, a set of data for matrices A and B can be obtained by two distinct positions which result in an arbitrary robot movement (Figure 5.4). X is the required transformation: H_P^C as illustrated in Figure 5.3.

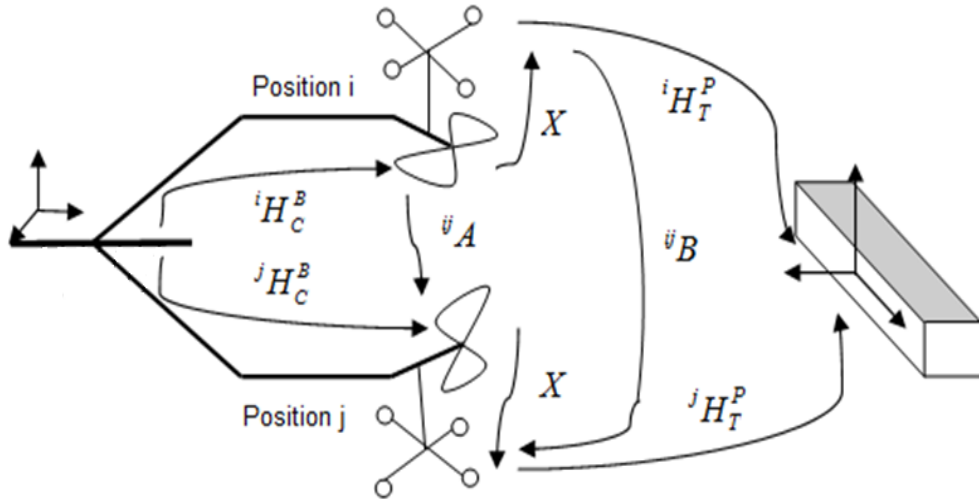


Figure 5.4 Robot moves from position i to position j , matrix X is the desired transformation.

In order to specifically analyse the transformation, a simplified form of Figure 5.4 is illustrated in Figure 5.5, and to locate the coil with respect to the tracking system frame, the transformation X has to be known at any instant. Base on the homogeneous transformations defined by Coil(i), Coil(j), Tool(i) and Tool(j) in the figure, X can be found by formulated the equation

$${}^{ij}AX = X{}^{ij}B \quad (5.1)$$

where ${}^{ij}A$ is the position and orientation of the coil relative to itself after the robot is moved from position i to position j . ${}^{ij}B$ is the position and orientation of the passive tool relative to itself after the robot is moved from position i to position j , and X is the transformation from the coil to the passive tool frame.

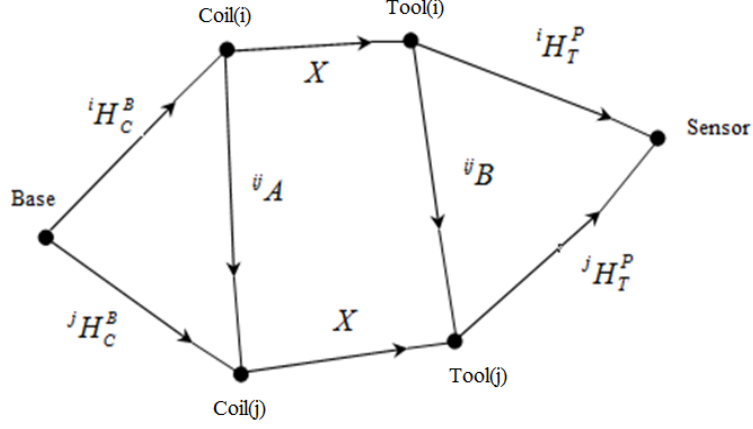


Figure 5.5 Transformations of the system when robot moves from position i to j .

One equation can be obtained from the homogeneous transformations defined by Base, Coil(i) and Coil(j):

$${}^j H_C^B = {}^i H_C^B {}^{ij} A \quad (5.2)$$

And another equation can be obtained from the homogeneous transformations defined by Sensor, Tool(i) and Tool(j):

$${}^i H_T^P = {}^{ij} B {}^j H_T^P \quad (5.3)$$

where ${}^i H_C^B$ is the transformation from the robot base frame to the coil frame when the robot is located at position i , and ${}^j H_C^B$ is the one at position j . ${}^i H_T^P$ is the transformation from the passive tool frame to the tracking system frame when the robot is located at position i , and ${}^j H_T^P$ is the one at position j .

The transformation matrices A and B , can be calculated as below:

$${}^{ij} A = ({}^i H_C^B)^{-1} {}^j H_C^B \quad (5.4)$$

$${}^{ij} B = {}^j H_T^P ({}^i H_T^P)^{-1} \quad (5.5)$$

where H_C^B can be obtained from the robot forward kinematics, and H_T^P is directly measured by the position sensor. X is the only unknown transformation for locating an object in the space, as shown in Figure 5.5.

5.3 Solutions for the Shiu-Ahmad homogenous matrix equation

$$AX = XB$$

Studies have been conducted by Shiu and Ahmad [1989] and Tsai et al.[1989] independently to solve the homogenous transform formulation $AX = XB$. Tsai's method has been shown to be more robust [Wang, 1992]. It has been proven in both papers that the solution to an equation of the form $AX = XB$ has one degree of rotational freedom and one degree of translation freedom. Hence, in order to obtain a unique solution for the equation, at least two sets of data for matrices A and B (three robot positions) need to be acquired initially. This can be achieved by two consecutive movements of the robot arm, and more robot positions can be used to obtain an increased set of data to improve the accuracy of the solution. Furthermore, the compulsory condition to ensure uniqueness is that the inter-position rotation axes are not co-linear for different pairs of stations. This means that if there are two sets of data A_1 and A_2 , the axes of rotation of A_1 and A_2 are neither parallel or anti-parallel to one another. Two different solutions for the calibration, inspired by Tsai-Lenz's method, have been presented in the following sections. The first method using the equation $AX = XB$ is referred to Tsai-Lenz method [Tsai et al., 1989], and the second method using the equation ${}^iHX = {}^jHYB$ is referred to DH method [Horaud and Dornaika, 1995].

5.3.1 Calibration using Tsai-Lenz method

The purpose for solving the equation is to decouple the rotation part from the translation part, and allow the translational part to be obtained, once the rotational part is initially calculated. The homogenous matrix equation $AX = XB$ can be expressed in terms of a 4×4 homogenous matrix below:

$$\begin{bmatrix} R_A & T_A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_X & T_X \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X & T_X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B & T_B \\ 0 & 1 \end{bmatrix} \quad (5.6)$$

where R is the 3×3 rotational matrix and T is the 3×1 translational vector. After multiplying, it becomes

$$\begin{bmatrix} R_A R_X & R_A T_X + T_A \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X R_B & R_X T_B + T_X \\ 0 & 1 \end{bmatrix} \quad (5.7)$$

Equating right and left sides, the rotational and the translational components are separated, and the problem reduces to the solution of two simpler equations as below:

$$R_A R_X = R_X R_B \quad (5.8)$$

$$R_A T_X + T_A = R_X T_B + T_X \quad (5.9)$$

Several steps have been introduced in this application in order to find the solution of Eqs.5.8 and 5.9 :

1. Build the matrices A and B . Consecutive robot movements should be carried out initially using Eqs.5.4 and 5.5 to calculate the value. At least two sets of data are required (two robot movements) to obtain a unique solution for the matrix equation.
2. Represent the rotation parts using the angle-axis representation from the Rodrigues formula in order to simplify the calculation. A Cartesian coordinate rotation can be considered as a rotation of a unit vector (n_1, n_2, n_3) by an angle θ [Rogers and Adams, 1989]. The 3×3 rotational matrix R is given by:

$$R = \begin{bmatrix} n_1^2 + (1 - n_1^2) \cos \theta & n_1 n_2 (1 - \cos \theta) - n_3 \sin \theta & n_1 n_3 (1 - \cos \theta) + n_2 \sin \theta \\ n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_2^2 + (1 - n_2^2) \cos \theta & n_2 n_3 (1 - \cos \theta) - n_1 \sin \theta \\ n_1 n_3 (1 - \cos \theta) + n_2 \sin \theta & n_2 n_3 (1 - \cos \theta) + n_1 \sin \theta & n_3^2 + (1 - n_3^2) \cos \theta \end{bmatrix} \quad (5.10)$$

One of the eigenvalues and eigenvectors of R must be 1 and the rotational axis respectively, since by definition, R is a rotation around the rotational vector $A \equiv (n_1, n_2, n_3)'$, and obviously :

$$A = RA \quad (5.11)$$

thus A is an eigenvector (or principal vector) and its corresponding eigenvalue is 1. From Eq.(5.10), the rotation matrix could be represent by a vector and an angle A and θ , and A can be defined using the Rodrigues formula:

$$A = 2 \sin \frac{\theta}{2} (n_1, n_2, n_3)', 0 \leq \theta \leq \pi \quad (5.12)$$

In this way, R is a function of A :

$$R = \left(1 - \frac{|A|^2}{2}\right)I + \frac{1}{2}(AA^T + \alpha \cdot Skew(A)) \quad (5.13)$$

where $Skew(A) = \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix}$ and $\alpha = \sqrt{4 - |A|^2}$.

3. Form a function of A'_X (Eq.5.14) for calculating the axis/angle of the rotational part:

$$Skew(A_A + A_B)A'_X = A_B - A_A \quad (5.14)$$

By using the axis/angle representation A_A , A_X and A_B and two significant lemmas (Eq.5.15 and Eq.5.16) proved by Tsai et al. [1989] below:

$$(A_A + A_B) \times A_X' = A_A - A_B \quad (5.15)$$

$$a \times b = \text{Skew}(a)b \quad (5.16)$$

where $A_X' = \frac{A_X}{\sqrt{4 - |A_X|^2}}$, and a and b are vectors.

4. Substituting A_A and A_B obtained from step one into Eq.5.14 and approximating A_X' using Singular Value Decomposition (SVD).
5. Compute R_X . A_X can be obtained by Eq.(5.17), and substituting A_X into Eq.(5.13), R_X is obtained.

$$A_X = \frac{2A_X'}{\sqrt{1 + |A_X'|^2}} \quad (5.17)$$

6. Finally, the translational part T_X can be computed by substituting R_X into Eq.(5.9).

5.3.2 Calibration using DH method

The homogenous matrix equation $AX = XB$ is widely used for hand-eye calibration. Another matrix equation can also be used to solve the calibration, and has the form ${}^iHX = {}^jHYB$ [Horaud and Dornaika, 1995]. It is generated by introducing the transformation Y in place of X used in the form of $AX = XB$. Because of the linear relationship between Y and X , the new equation has the same properties. Further details are introduced below.

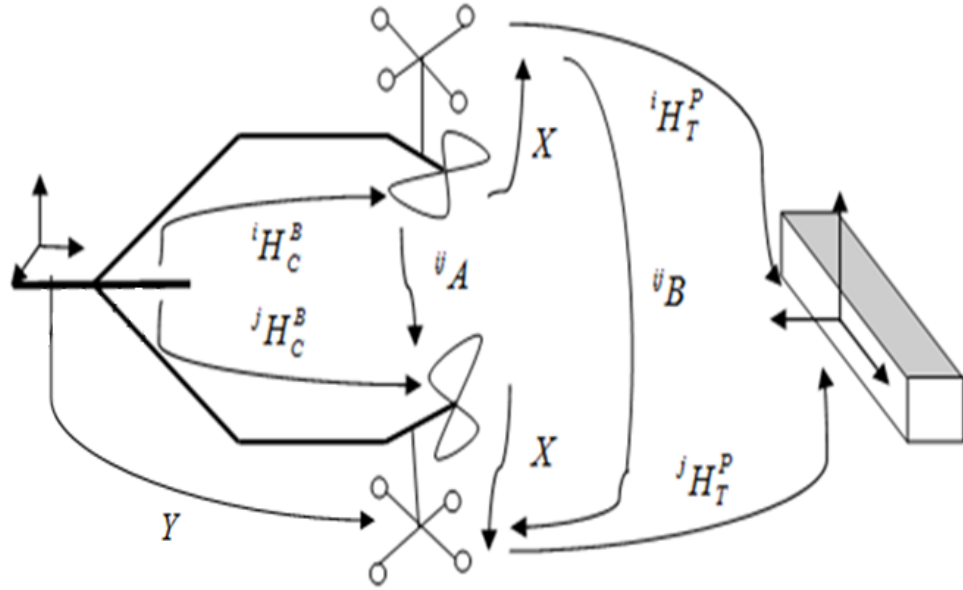


Figure 5.6 Two robot position i and j , matrix Y is the desired transformation.

From Figure 5.6, Y is defined as the transformation from robot base frame to the passive tool local frame in position j , which allows X to be written as a function of Y :

$$X = ({}^jH_C^B)^{-1}Y \quad (5.18)$$

substituting equations 5.4, 5.5 and 5.18 into 5.1 yields:

$$({}^iH_C^B)^{-1}Y = {}^jH_C^B Y {}^j\tilde{u}_B \quad (5.19)$$

this can be written in terms of 3×3 rotational matrix and 3×1 translational vector such that:

$$\begin{bmatrix} {}^iR_B^C & {}^iT_B^C \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_Y & T_Y \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^jR_C^B & {}^jT_C^B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_Y & T_Y \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B & T_B \\ 0 & 1 \end{bmatrix} \quad (5.20)$$

equating the right and left sides, the problem only requires the solution of two equations as follows:

$${}^i R_B^C R_Y = {}^j R_C^B R_Y R_B \quad (5.21)$$

$$({}^i R_B^C - {}^j R_C^B) T_Y = {}^j R_C^B R_Y T_B + {}^j T_C^B - {}^i T_B^C \quad (5.22)$$

since there is a linear relationship between X and Y , equations 5.21 and 5.22 can be expressed as a function of X :

$${}^i R_B^C R_X = {}^j R_C^B R_X R_B \quad (5.23)$$

$$({}^i R_B^C - {}^j R_C^B) T_X = {}^j R_C^B R_X T_B + {}^j T_C^B - {}^i T_B^C \quad (5.24)$$

hence, the problem reduces to find the solutions to Eq.5.23 and 5.24. Comparing Eq.5.23 with Eq.5.8, it can be noted that there is no difference within the rotational parts, however, there is a significant difference in the translational part. Using Eq.5.24, the translation is calculated from two independent robot positions but there is no need to address the relationship between two consecutive motions of the robot. The $AX = XB$ form used a combined rotation R_A and translation T_A , but the relationship of two consecutive robot movements affect the calibration accuracy significantly if a degenerate motion occurs. The major steps for solving the rotational part of the new form equation are similar to Tsai-Lenz method as follows:

1. Build the matrices A , B , ${}^j H_C^B$ and ${}^i H_B^C$. Consecutive robot movements should be carried out initially using Eq.(5.4) and (5.5). At least two robot movements are required to obtain a unique solution for the matrix equation.
2. Represent the rotation parts using quaternion. As illustrates in Appendix 1.5, rotational matrix R_A and R_B can be expressed using quaternion $Q_A = (a_0, a_1, a_2, a_3)^T = (a_0, a)^T$ and $Q_B = (b_0, b_1, b_2, b_3)^T = (b_0, b)^T$.
3. Transform Eq.(5.23) in terms of quaternion representation. Eq.(5.23) is equivalent to Eq.(5.8), and is written in the form:

$$Q_A * Q_X = Q_X * Q_B \quad (5.25)$$

Eq.4.25 can be further transformed to [Wehage, 1984]:

$$(Q_A^+ - Q_B^-) Q_X = 0 \quad (5.26)$$

where $Q_A^+ = \begin{bmatrix} a_0 & -a \\ a & a_0 I + \hat{a} \end{bmatrix}$, $Q_B^- = \begin{bmatrix} b_0 & -b \\ b & b_0 I + \hat{b} \end{bmatrix}$, $\hat{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ and

$$\hat{b} = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}.$$

4. Compute R_X . Q_X can be obtained by solving Eq.(5.26) using least square or SVD (singular value decomposition), and then transform it to matrix R_X .
5. Compute T_X . The translational part is computed by substituting R_X into Eq.(5.24).

5.4 Calibration experiment

In order to evaluate and select which of the previously described techniques is suited for this application, a calibration experiment was carried out to compute the homogenous transformation from the coil coordinate system to the passive tool local coordinate system H_p^C . Also to obtain the homogenous transformation from tracking system coordinate system robot base coordinate system H_B^T , and finally to compare the resulting errors.

5.4.1 Method of error checking

As illustrated in Figure 5.7, a relationship for the homogenous transformation matrix can be found using:

$$H_C^B * H_P^C * H_T^P * H_B^T = I \quad (5.27)$$

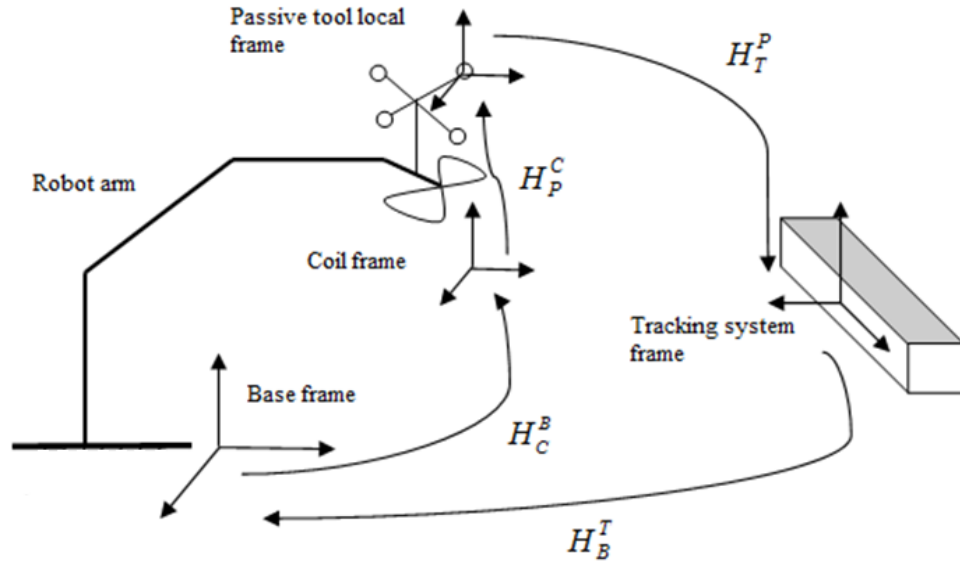


Figure 5.7 All coordinate systems and transformations in the calibration process.

Once H_P^C and H_B^T are obtained by the calibration, H_C^B can be obtained by:

$$H_C^B = (H_B^T)^{-1} * (H_T^P)^{-1} * (H_P^C)^{-1} \quad (5.28)$$

H_C^B can also be calculated by the robot forward kinematic equations, which is referred as ${}^2H_C^B$. Since there is no absolute H_P^C to compare, an error check can be made using H_C^B which is calculated from the function of H_P^C (Eq.5.28). The error in calibration between the robot and the tracking system can be obtained by comparing H_C^B and ${}^2H_C^B$. It can be observed from Eq.5.28 that the error obtained by this method includes the position tracking error (0.4mm) and the positioning error of the robot (0.02mm).

Setting ${}^2H_C^B$ as the reference, the root mean square (RMS) variation can be calculated using:

$$ROTATION_{RMS} = \frac{1}{n} \sqrt{\sum_1^n \frac{\|{}^2H_C^B(rotation) - H_C^B(rotation)\|^2}{M}} \quad (5.29)$$

$$TRANSLATION_{RMS} = \frac{1}{n} \sqrt{\sum_1^n \frac{\|{}^2H_C^B(translation) - H_C^B(translation)\|^2}{M}} \quad (5.30)$$

where the rotation and translation part of H_C^B and ${}^2H_C^B$ have been separated to calculate the separate rotation and translation RMS variations. n is the number of calibrations conducted for checking the error and M is the number of elements of the matrix or vector.

5.4.2 Experimental procedure of the calibration using Tsai-Lenz and DH method

The following steps were carried out for the experiment:

1. Fix the position sensor in the work space and mount the passive tool on the end-effector of the robot (Figure 5.8). The robot movement should be restricted to be within the measurement volume of the Polaris system.
2. Move the robot arm to $2n$ ($n \geq 2$) different positions, using n positions for calibration and the others for error checking. At each position, record the transformation ${}^2H_C^B$ and H_T^P .

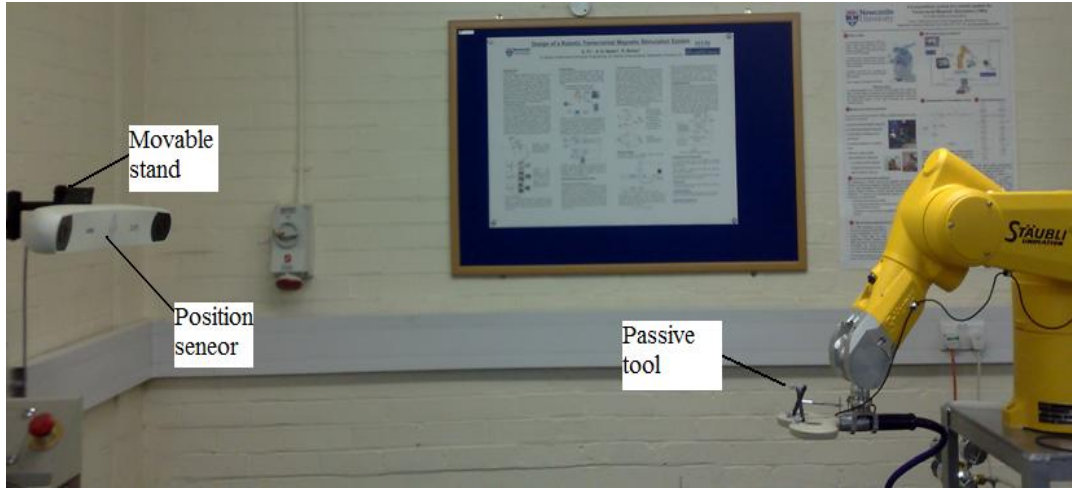


Figure 5.8 Experimental setup. The position sensor is mounted on a movable stand, and a passive tool is mounted on the robot end-effector.

3. Compute H_p^C using both Tsai-Lenz and DH methods with the data from position 1 to n collected in the previous step. Next, calculate H_B^T at each station (1 to n), and take an average of the values.
4. Calculate the transformation H_C^B using Eq.5.28, and compare the results with ${}^2H_C^B$ recorded in step 2 to evaluate the error.

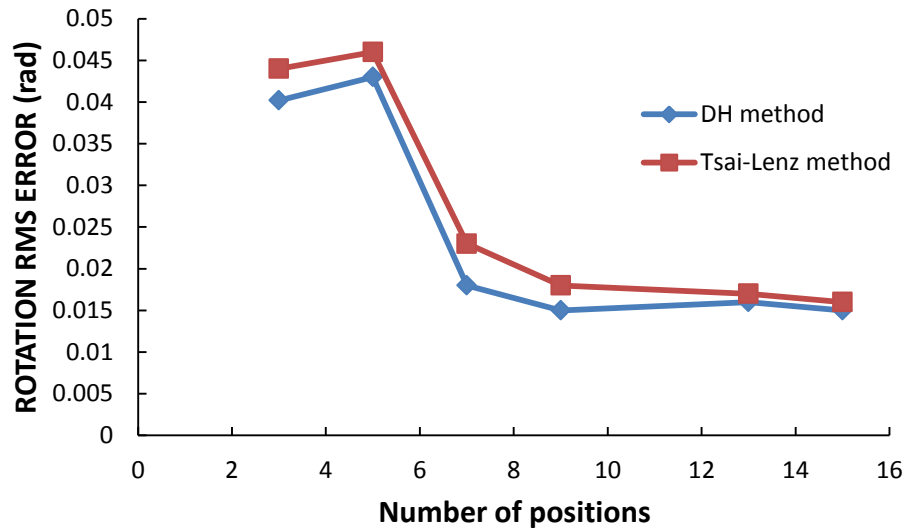
5.4.3 Results and conclusion

Table 5.1 presents the results of a calibration sequence carried out with different number of robot positions. As mentioned previously, the error of the calibration includes both the error of the tracking system and the positioning error of the robot. It can be noticed from the table that for 13 positions, the translational error is 0.55mm, but the error of the tracking system is claimed to be 0.4mm, which indicates that the calibration has an accuracy better than 0.2mm.

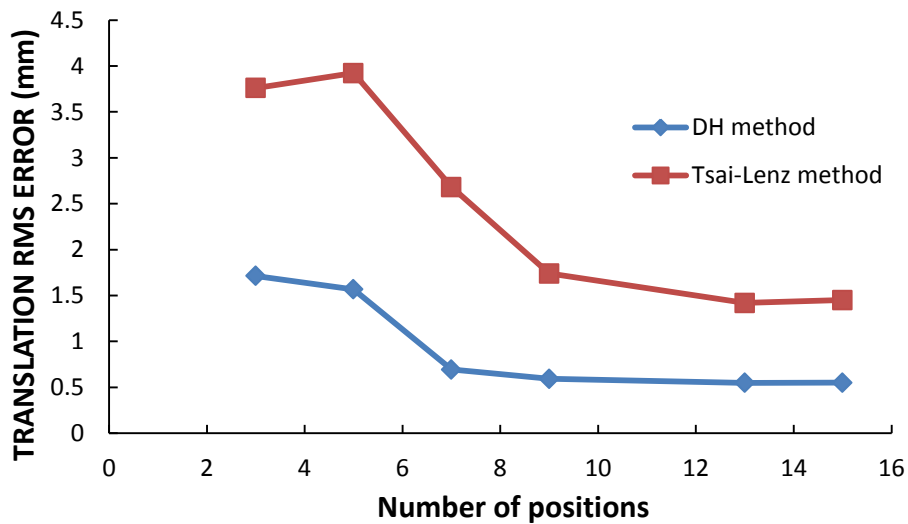
Table 5.1 Errors of calibration

Number of positions	DH method		Tsai-Lenz method	
	Error in rotation (rad)	Error in translation (mm)	Error in rotation (rad)	Error in translation (mm)
3	0.0402	1.7143	0.044	3.76
5	0.043	1.5665	0.046	3.9221
7	0.018	0.694	0.023	2.6804
9	0.015	0.5932	0.018	1.74
13	0.016	0.5493	0.015	1.42
15	0.015	0.5502	0.016	1.45

As illustrated in Figure 5.9, both the rotation and translation RMS values are decrease to a minimum after 10 positions. It can be observed that the rotational error with five positions is slightly increased in both methods because of degenerated motions. However, the translational error is different; the error of the Tsai-Lenz method is also increased, while the error of the DH method is still decreased. These results verify the better performance of Eq.5.24 which uses two independent robot position to calculate the translation but not a combined rotation R_A and T_A . The combined rotation and translation of A used for calculating the translational part can accumulate the errors to the translational part.



(a) Error in rotation



(b) Error in translation

Figure 5.9 RMS error in rotation and translation with two difference form equations respectively.

It can be seen that the accuracy of both methods are significantly improved compared with the direct measurement using a reference frame. And the experimenter can adjust the Polaris system to obtain a better view for the system, as the re-calibration is easy to carry out with the methods. The distribution of the error

obtained from DH method is studied in the x-y plane (Figure 5.10), and shows as random data points collected in the x-y plane (150*200mm), The results shows that the error is randomly distributed in the space.

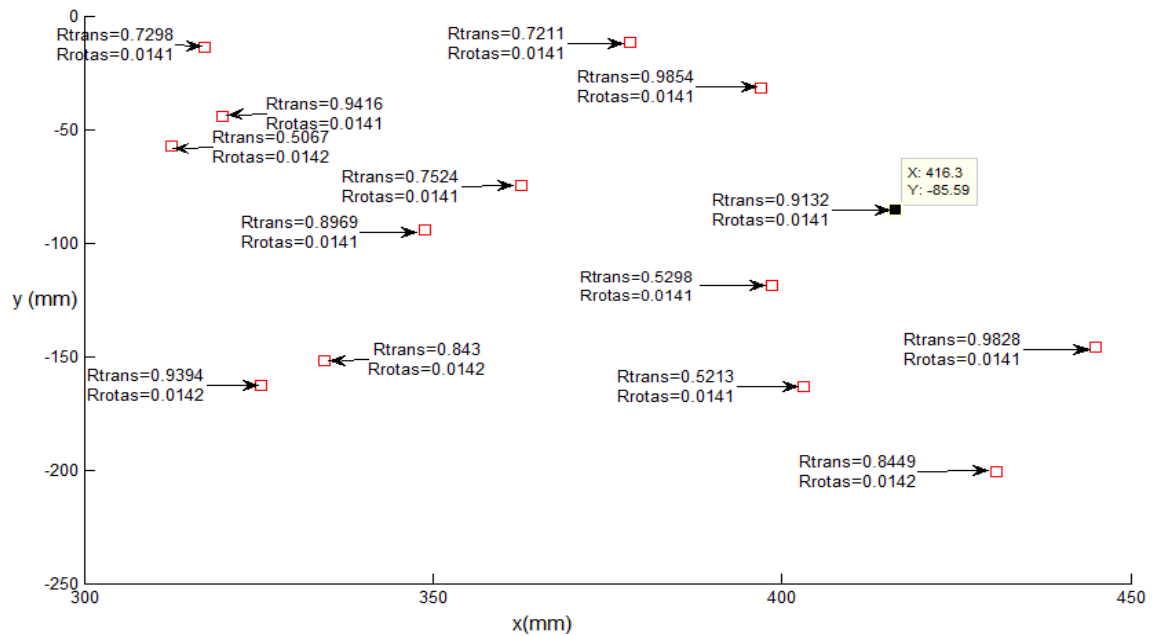


Figure 5.10 Distribution of the error in translation **Rtrans** (mm) and rotation **Rrotas** (radians).

5.4.4 Practical consideration

The number of robot positions used in the calibration significantly affects the accuracy in this system configuration. There are two other factors that can significantly affect the accuracy of the calibration, regardless which algorithm is used. The first is the inter-position rotation angle, i.e. the rotation angle generated by the rotational part of transformation A . Figure 5.11 presents the rotational and translational errors of a 7 position calibration using different inter-position rotation angles. It can be seen that the magnitudes of both the rotational and translational errors are proportional to this angle. Hence, in order to obtain better accuracy, the inter-position rotation angle should be made as large as possible when moving the robot between positions.

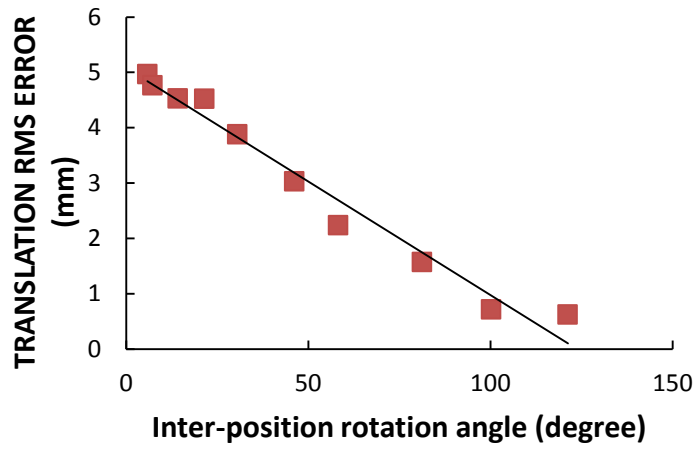
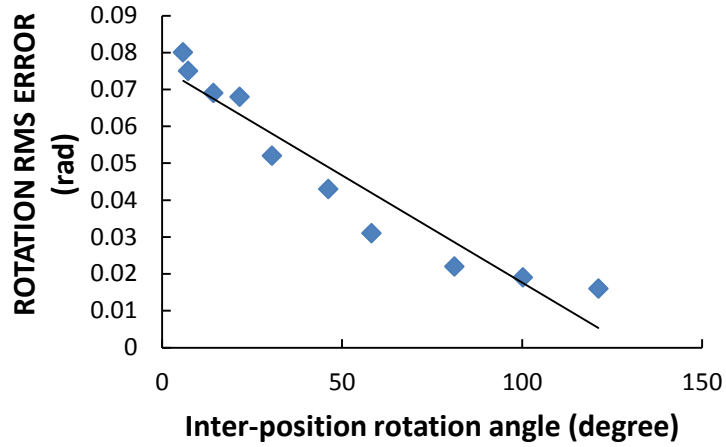


Figure 5.11 Rotational and translational errors with different inter-position rotation angle.

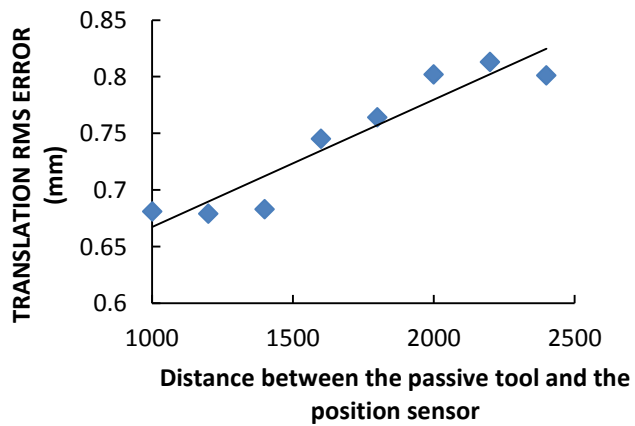


Figure 5.12 RMS error in translation and rotation with same robot movement but different distance between the tool and position sensor.

Another factor that affects the translational error is the centre distance between the passive tool local coordinate system and the centre of the Polaris Spectra tracking system. As illustrated in Figure 5.12, if the robot movements are obtained at a shorter distance, accuracy is improved. Note that this distance does not affect rotational error.

5.5 Coordinate systems registration in TMS system

The transformations between the coordinate systems of the robot arm and the tracking system have been addressed in previous sections. This section provides a further description of the relationship of all the coordinate systems with the subject's head coordinate system introduced.

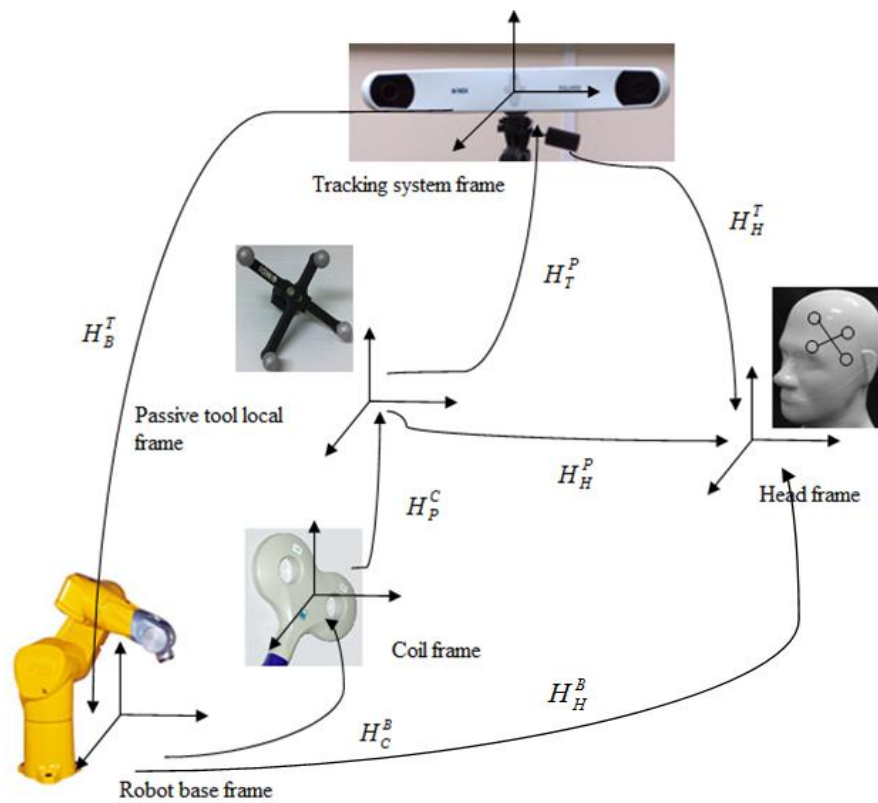


Figure 5.13 Coordinate systems and transformations in the robotic TMS system.

Figure 5.13 shows all the coordinate systems and key transformations in the robotic TMS system. Most of the definitions of the transformation matrices have been presented in previous section with the exception of:

H_H^P defines the coordinate transformation of the head coordinate system relative to the passive tool local coordinate system.

H_H^T defines the coordinate transformation of the head coordinate system relative to the tracking system coordinate system.

H_H^B defines the coordinate transformation of the head coordinate system relative to the robot base coordinate system.

5.5.1 Definition of the subject's head coordinate system

To monitor and track the movement of the subject's head, a coordinate system must be assigned, using a passive tool fixed on the subject's head. In this way, the local coordinate system of the passive tool can be used to define the head coordinate system. The Polaris system is used to track the relationship between the robot arm and head coordinate systems.

5.5.2 Transformation between passive tool and head coordinate systems

The transformation from the passive tool and head coordinate system is obtained in order to establish the relationship between the robot arm base/tool and the head coordinate system (Figure 5.14).

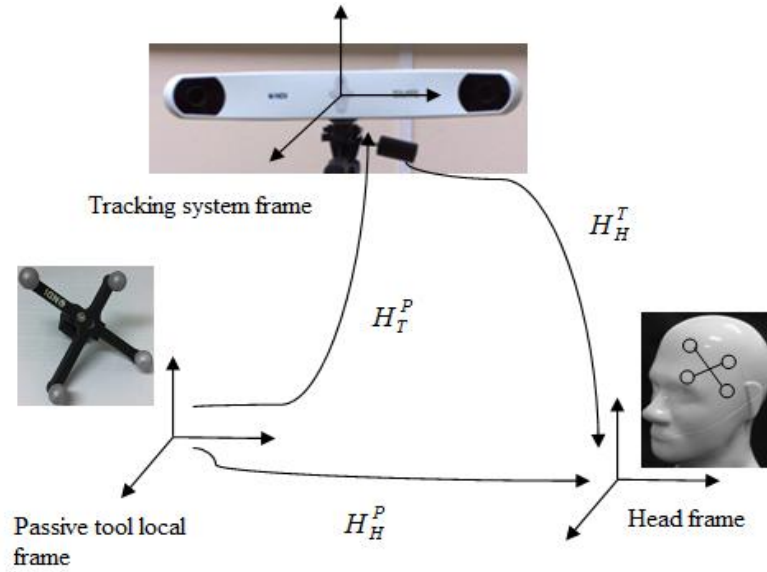


Figure 5.14 Transformations between passive tool and head coordinate system.

Using the tracking system, the transformation can be easily obtained via the following equation:

$$H_H^P = H_T^P * H_H^T \quad (5.31)$$

furthermore, the transformation from robot base coordinate system to head coordinate system H_H^B can be calculated by:

$$H_H^B = H_C^B * H_P^C * H_H^P \quad (5.32)$$

5.6 Summary

This chapter has presented and evaluated two techniques for coordinate system registration based on the methodology of Tsai-Lenz and DH. The accuracy of calibration of the DH method using the transformation equation ${}^iHX = {}^jHYB$ was found to give improved accuracy over the Tsai-Lenz method, and was subsequently adopted for use in this application.

CHAPTER 6

DESIGN OF A COMBINED POSITION AND FORCE CONTROLLER

This chapter presents the position and force control methods used in the real-time tracking mode of the system. Position control algorithms used to track a moving object with an external position sensor are proposed. Furthermore, a combined ‘hybrid’ position and force controller is developed. Most conventional hybrid position and force controllers were developed for moving the manipulator on a fixed but constrained surface, which requires hand mounted sensors to obtain the force feedback and encoders attached to the driving motors of the manipulator for position feedback [Raibert and Craig, 1981]. However, moving objects are tracked by the robot arm in this application, and an external position sensor is required to track this movement. This requires the position and orientation feedback to be transformed from the position sensor frame to the robot frame. In addition, the force control not only deals with the small force perturbations, but also compensates for the subject’s movement in its controlled direction. The position control strategy is addressed as follows.

6.1 Position tracking control strategy

The control strategy for position tracking is composed of two interlinked processes. The first one tracks the movement of the subject's head to obtain and update the continuous estimation of the new position and provide this information to the robot controller. The other is to control the robot motion according to the tracking process. Following the pre-operative acquisition, the system will perform the tracking by running these two processes simultaneously.

As discussed in Chapter 4, the tracking mode is activated after determining the stimulation site. This ensures that the coil centre point and the stimulation site on the subject's head coincide at the moment when tracking starts. The position of the stimulation point with respect to the head coordinate system can be represented by the relative transformation H_H^P between the passive tool local frame and the head frame at this moment (Figure 6.1). In this way, the relative transformation H_H^P should be kept constant to ensure the coil is maintained at the location of the stimulation site even when the subject's head moves. As shown in Figure 6.1, if the coil is in position and reaches the desired site when the subject's head is at position 1, the transformation H_H^P at this point is obtained using Eq.(6.1).

$$H_H^P = H_T^P * {}^1H_H^T \quad (6.1)$$

Assuming the subject's head moves to position 2, \hat{H}_H^P should equal H_H^P in order to ensure the coil is still maintained in the same relative position and orientation. Since the transformation from the tracking system to the head coordinate system ${}^2H_H^T$ is changed, the transformation from the passive tool to the tracking system frame H_T^P should also make the corresponding change in order to ensure H_H^P remains unchanged.

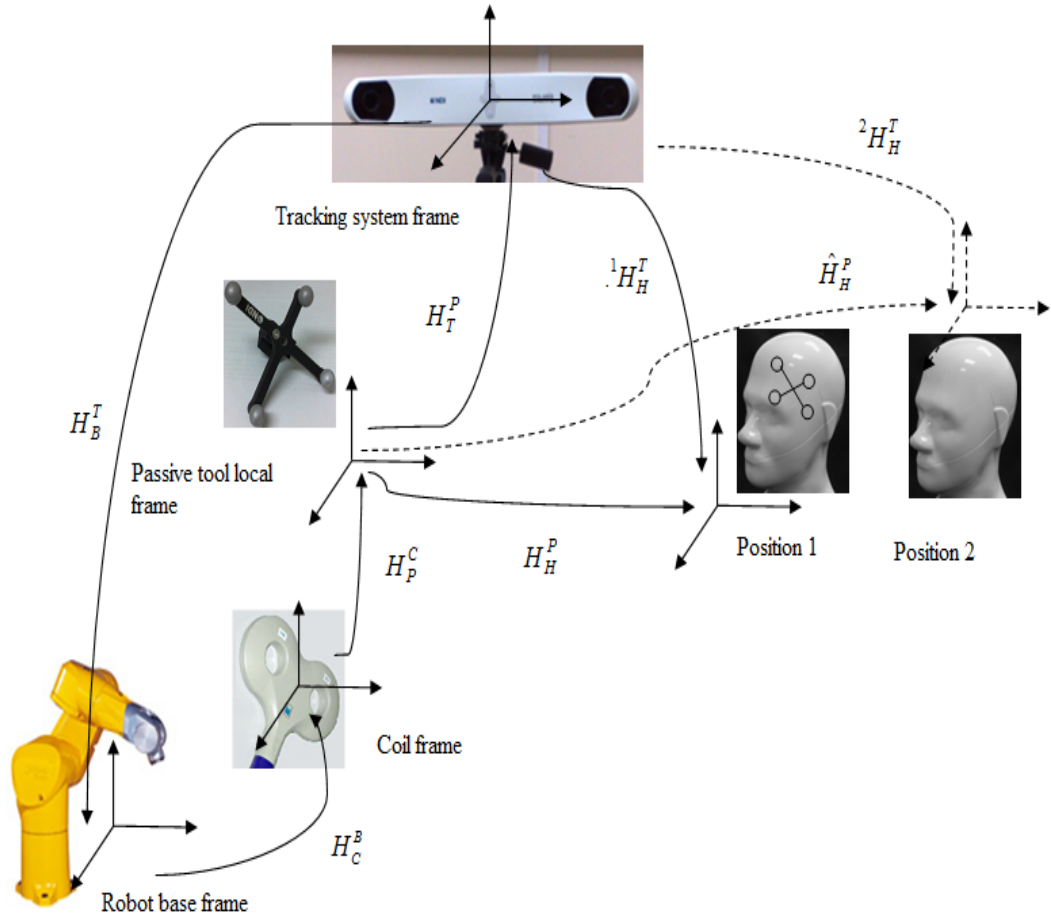


Figure 6.1 Changes in transformations when subject moves from position 1 to position 2.

As described above, the tracking function can be represented as H_T^P , such that:

$$F(H_T^P; {}^iH_H^T, H_H^P) = H_T^P - H_H^P * {}^iH_H^T \quad (6.2)$$

where $\{F\}$ is defined as the function, the transformation H_T^P is the variable to be controlled, whilst the variable ${}^iH_H^T$ is obtained in real-time using the tracking system and H_H^P is a constant transformation.

Furthermore, the transformation H_T^P can be converted into a function with respect to the robot kinematic transformation, then F becomes a function of H_C^B :

$$F(H_C^B; {}^iH_H^T, H_H^P) = (H_P^C)^{-1} * (H_C^B)^{-1} * (H_B^T)^{-1} - H_H^P * ({}^iH_H^T)^{-1} \quad (6.3)$$

where H_C^B is the transformation of robot forward kinematics, and H_P^C and H_B^T are the matrices obtained from the coordinate systems registration.

A control method (i.e. PID control or fuzzy logic control) can also be introduced, where, if α denotes the control method, the equation becomes:

$$F(H_C^B; {}^iH_H^T, H_H^P) = \alpha((H_P^C)^{-1} * (H_C^B)^{-1} * (H_B^T)^{-1} - H_H^P * ({}^iH_H^T)^{-1}) \quad (6.4)$$

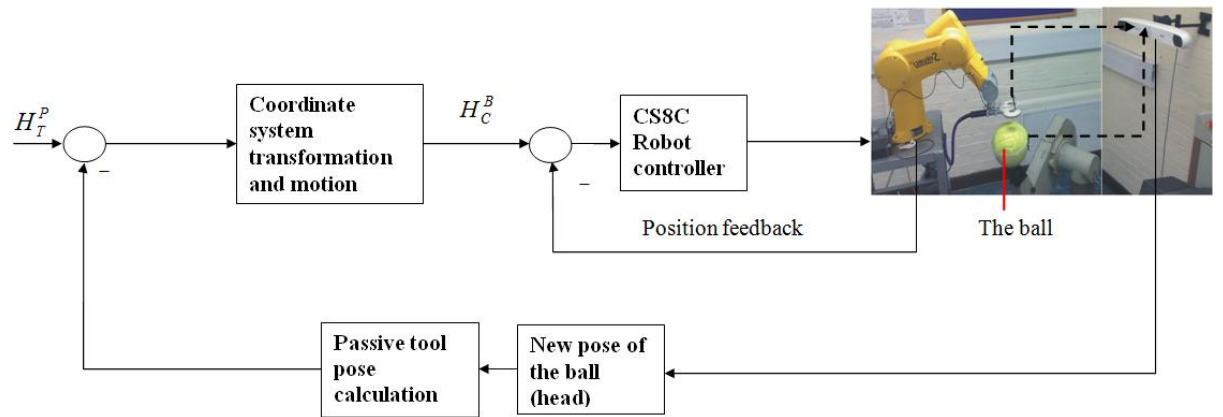


Figure 6.2 Control structure of the tracking system.

The control structure is shown in Figure 6.2. The tracking system updates the current position of the subject's head (the ball in the figure) and the system uses the positional (and orientation) changes to calculate the new robot position.

6.2 The combined force and position controller

6.2.1 The conventional hybrid controller

The conventional hybrid force and position controller [Raibert and Craig, 1981] has two complementary sets of feedback loops for separated control of position and force (Figure 6.3), each with its own control law and sensor feedback which must be transformed from the sensor's coordinate system to the robot's coordinate system before the position or force errors are calculated.

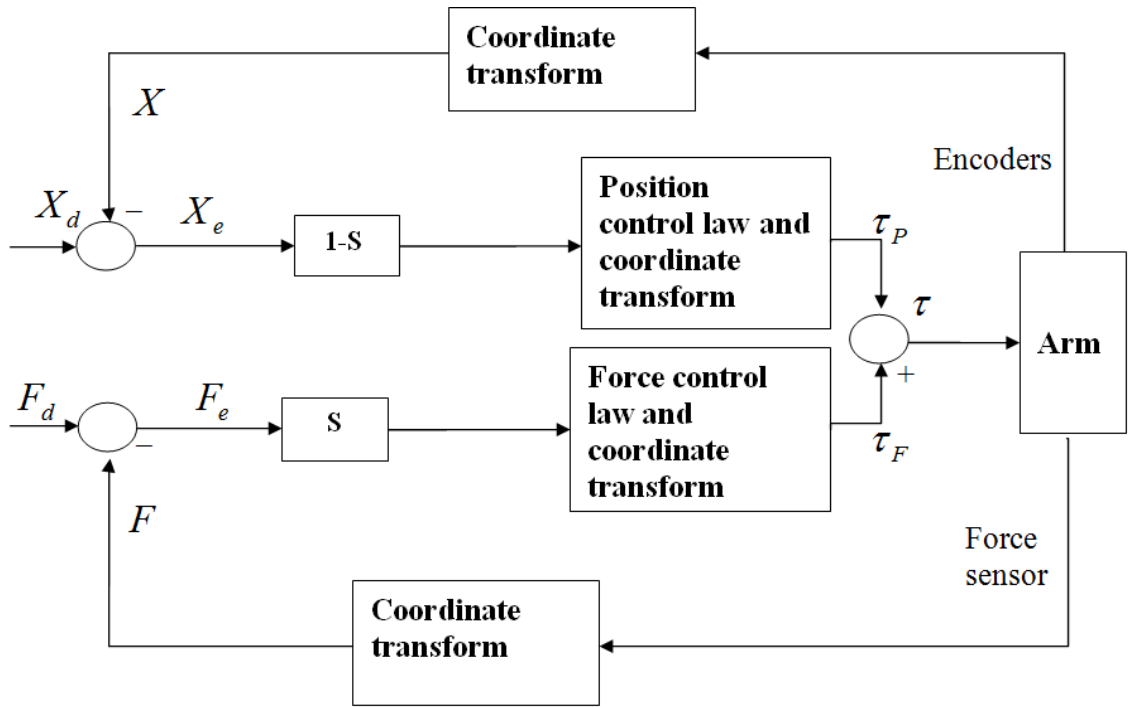


Figure 6.3 Conceptual organisation of conventional hybrid controller [Raibert and Craig, 1981].

The actuator control signal can be expressed as Eq.6.5:

$$\tau = A(S \times F_e) + B((1 - S) \times X_e) \quad (6.5)$$

where:

τ is torque applied on the actuators;

F_e and X_e are force error and position error respectively;

A and B are force and position compensation functions respectively;

S is the compliance selection matrix.

A compliance selection matrix is used to specify which axes are under force control (indicated by $S_i = 1$) and which are under position control (indicated by $S_i = 0$). For

example, if there are three degrees of freedom and $S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, then the second

axis is controlled by position while the others are controlled by force.

6.2.2 The combined force and position controller

The hybrid force and position control method is normally employed on manipulators with joint-mounted position sensors and wrist-mounted force sensors. In this system, an additional external position tracking sensor is used to track a moving object in all six axes rather than following a trajectory on a fixed object.

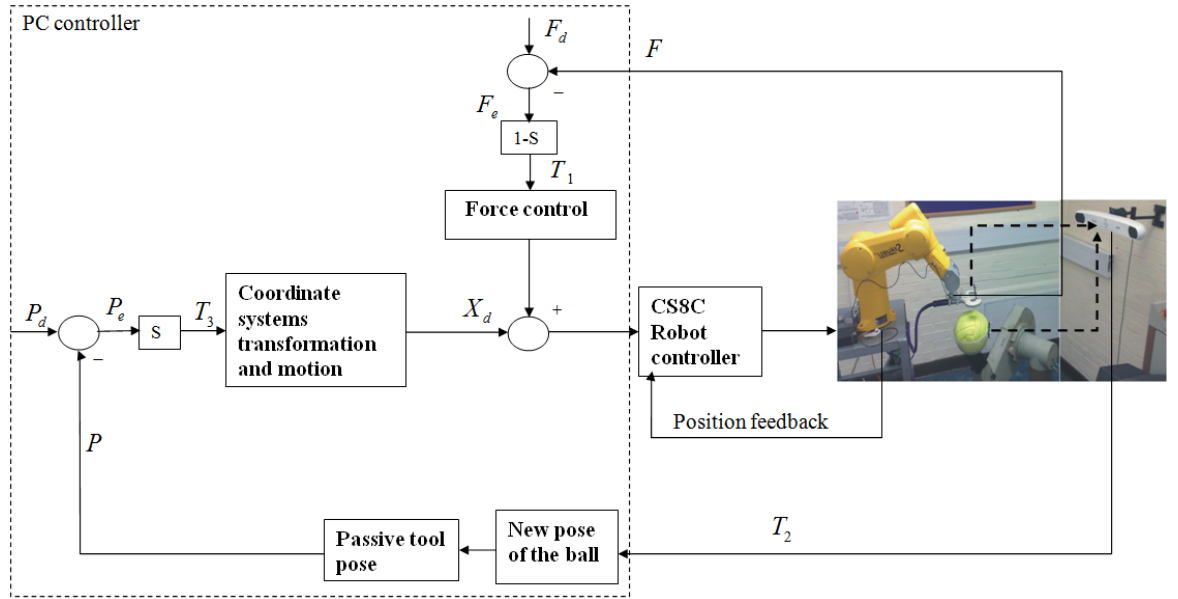


Figure 6.4 Combined position and force controller.

The combined control system is shown in Figure 6.4. Position control is applied along the unconstrained axes of the robot tool coordinate system, while force control is utilised along the constrained axes (Z axis with respect to tool frame). There are two main external loops within the control system: the position sensor and force sensor feedback loops, from which the desired position of the robot is computed, and then transmitted to the robot controller. The control equation for the combined position and force controller is given by as Eq.6.6:

$$X_d = \alpha((1-S) \times F_e) + \beta(S \times P_e) \quad (6.6)$$

where X_d is the desired position, F_e is the force error and P_e is the position error. S is the compliance selection matrix which is used to select which axes are position or

force controlled. T_1, T_2, T_3 in Figure 6.4 are transformation matrices used to ensure the coordinates are registered in the same coordinate system. α and β are force and position control laws respectively. A PI control law is adopted in this application and the functions of position and force control are written as:

$$p(t) = (1 - S)(K_{PP}P_e(t) + K_{PI} \int P_e(t)dt) \quad (6.7)$$

$$f(t) = S(K_{FP}F_e(t) + K_{FI} \int F_e(t)dt) \quad (6.8)$$

where:

K_{FP}, K_{FI} are proportional and integral force feedback gains;

K_{PP}, K_{PI} are proportional and integral position feedback gains;

the force and position errors are computed using:

$$F_e = F_d - F \quad (6.9)$$

$$P_e = P_d - P \quad (6.10)$$

F is the force measured using the force sensor, P is position and orientation measured by the position sensor, F_d is desired force, P_d is desired pose. In this application, the motion of the Staubli robot is controlled with respect to its tool coordinate system, so that the calculated coordinates have to be converted to with respect to the tool coordinate system.

6.3 Introduction of experimental plan

Experiments were carried out to examine the proposed control method. The relative position changes of both passive tools were used to evaluate the performance of the tracking control, as they were both collected by the position sensor and this can ensure the data was recorded in the same conditions. The major experiments were designed as follows: first of all, preliminary tests on position control were firstly conducted to verify the proposed position tracking algorithms. Tests were designed

to track the trajectories of a XY table, and optimal gains of position control were also obtained, which can be further used in the combined position and force control. The second experiment was to establish the ‘optimally’ gains for force control using both constant gains and gain scheduling method to demonstrate the advantages of the gain scheduling method. Finally, the combined position and force controller was tested in terms of disturbance occurs in XY plane, X,Y and Z axes, and orientation axes to demonstrate the behaviour of the proposed controller and the interaction between the axes controlled either by force and position controller.

$$ITAE = \int_0^{\tau=\infty} t|e(t)|dt \quad (6.11)$$

PI controller was applied on both position and force controlled axes, as the derivative term of a PID controller proved too sensitive to noise. Full factorial tests were carried out to find the regions of the gains, and the responses of the system with these gains were also evaluated and verified using the integral of time multiplied by absolute error (ITAE). This criterion is frequently used for gains tuning and evaluating the response of a controller [Seborg et al., 2006], with the best response corresponding to the minimum ITAE (Eq.6.11). Since it is not practicable to use $\tau = \infty$, the convention is to choose a value of τ sufficiently large so that $e(t)$ for $t > \tau$ is negligible. $\tau = 5$ seconds was used in the tests, which is enough to cover the period of system response. More details of these experiments are in the following section.

6.4 Preliminary testing of the position tracking method

Experiments were designed to test position tracing algorithms using a proportional-integral (PI) controller; the derivative term of a PID controller proved too sensitive to noise from the position sensor. The controller output is given by:

$$p(t) = (K_{pp}P_e(t)) + K_{pi} \int P_e(t)dt \quad (6.12)$$

where: K_{pp}, K_{pi} are the proportional and integral feedback gains; P_e is position error.

Although the lack of derivative action makes the system slower to reach the set point and to respond to perturbations than a well-tuned PID system, it makes the system more stable, which is more important in this application. An existing X-Y table (top left of Figure 6.5) was used to move the tracked objects and provide disturbance motions to the robot arm during the tests. The table had a travel range of 0 to 300 mm in both axes, and incorporates two single-axis PDX15 stepper drives (Parker Hannifin Ltd.). The PDX15 drive incorporates an RS232C-programmed indexer, which features a complete range of motion control commands using Digiplan's X-Code command language. Through the user interface, the speed and trajectory of the stepper motor can be controlled, to provide different conditions for testing the behaviour of the proposed algorithms.

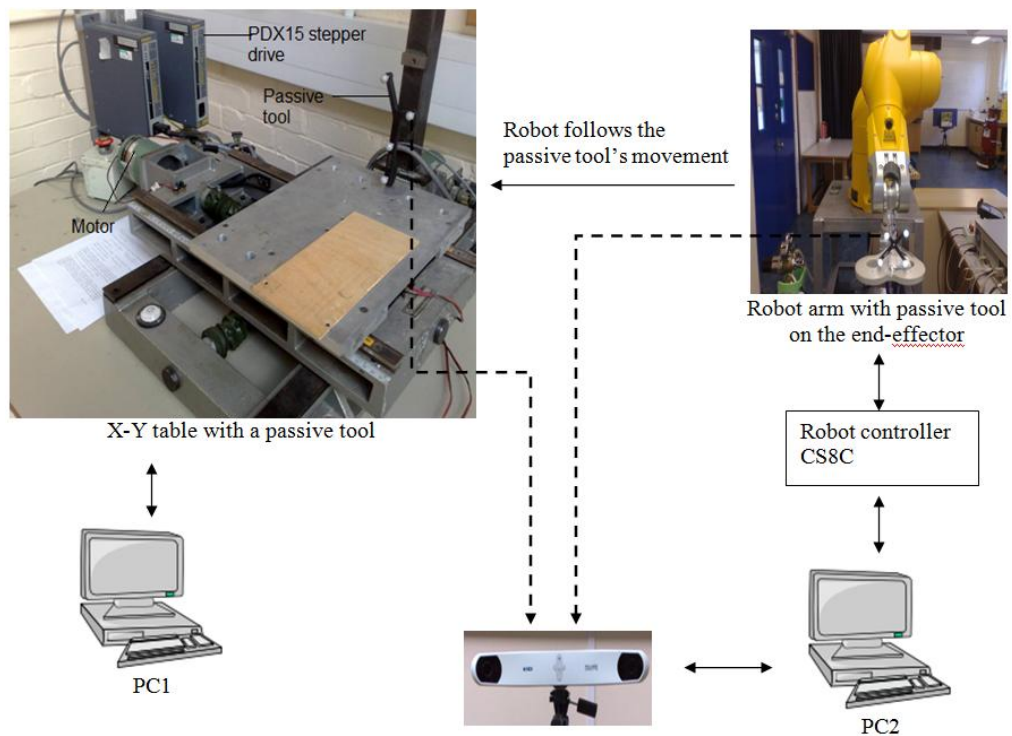


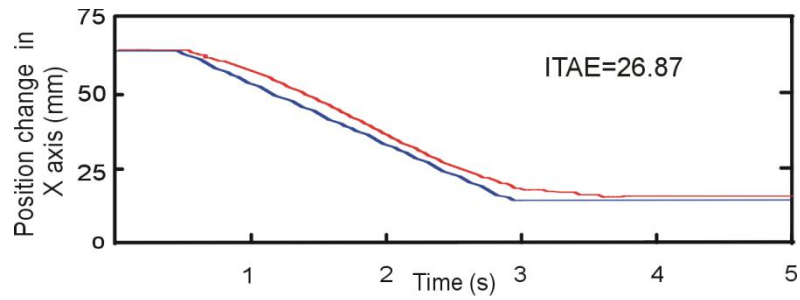
Figure 6.5 Structure of the test rig.

The test rig is illustrated in Figure 6.5. One passive tool was mounted on the X-Y table, and another was mounted on the robot end-effector to establish the relationship between the table and the robot end-effector. PC1 was used to independently control the movement of the X-Y table. The Polaris position sensor tracks the movement of the X-Y table by monitoring the position of the passive tool. PC2 was used to

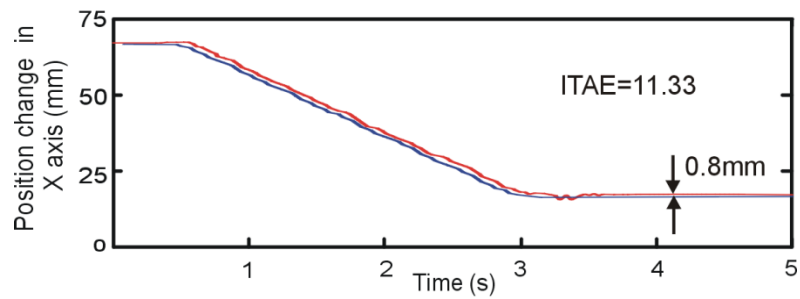
generate the updated robot position using the tracking algorithms to maintain the relative position between the robot end-effector and the X-Y table. During the experiment, the relative position changes of both tools were used to evaluate the performance of the tracking control methodology. Tests were initially conducted to establish the relative position and orientation between both tools after the coil was placed at the start position. The system then initiates the tracking by moving the X-Y table.

1. Tracking performances with different gains

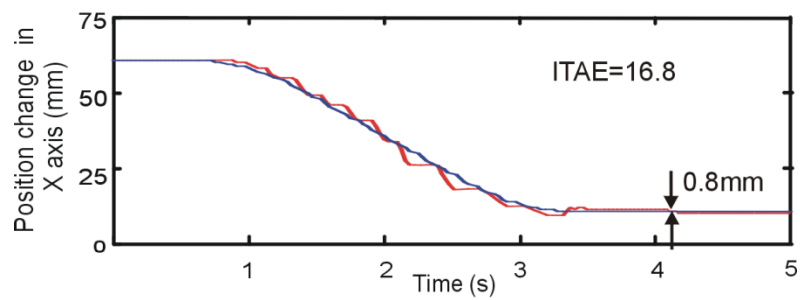
The tests were carried out to establish the optimal proportional and integral gains to minimise tracking error. Full factorial tests were carried out to find the regions of both gains, selected results were demonstrated here to show the system performance. The robot arm was first moved to the start position and then the relative position and orientation of both tools was recorded. The X-Y table was commanded to move along the X axis for 50mm at a speed of 20mm/s. The effect of different proportional gain is illustrated in Figure 6.6. The system response is improved by increasing the proportional gain (Figure 6.6 (a) and (b)), and the ITAE is reduced to $11.33\text{mm}\cdot\text{s}^2$ with $K_{pp} = 0.021, K_{pi} = 0.015$. With a further increase in $K_{pp} = 0.03$, the lag is minimised but there is a significant oscillation of the robot (Figure 6.6(c)). In addition, the ITAE rises to $16.8\text{mm}\cdot\text{s}^2$, which indicates that the system is becoming less stable at these gains. The position error of the system in steady state is consistently about 0.8mm once it reaches the steady state.



(a) System behaviour with $K_{pp} = 0.012, K_{pl} = 0.015$



(b) System behaviour with $K_{pp} = 0.021, K_{pl} = 0.015$

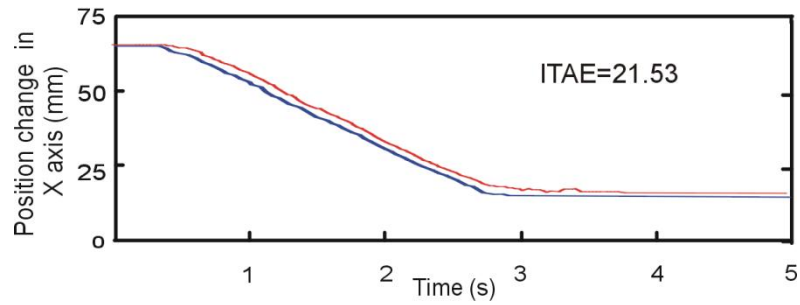


(c) System behaviour with $K_{pp} = 0.026, K_{pl} = 0.015$

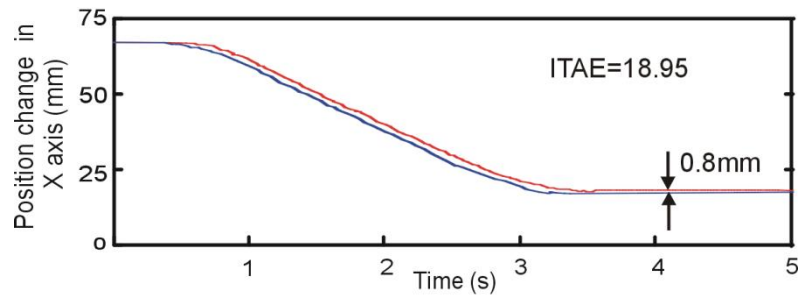
Figure 6.6 Changes of trajectories of both passive tools with different proportional gains. Blue curves are the trajectory of the tool mounted on the X-Y table and red curves are that of the tool on the robot arm.

Figure 6.7 (a)-(c) demonstrate the system's behaviour at different integral gains. Figure 6.7 (a) shows the response to the disturbance motion where $K_{pp} = 0.021, K_{pl} = 0.004$; it can be seen that with the smaller gains, the responsiveness is slow (ITAE of $21.53\text{mm}\cdot\text{s}^2$). When a larger integral gain is employed (Figure 6.7(b)), ITAE is reduced to $18.95\text{mm}\cdot\text{s}^2$. Some perturbations occur with further increase in

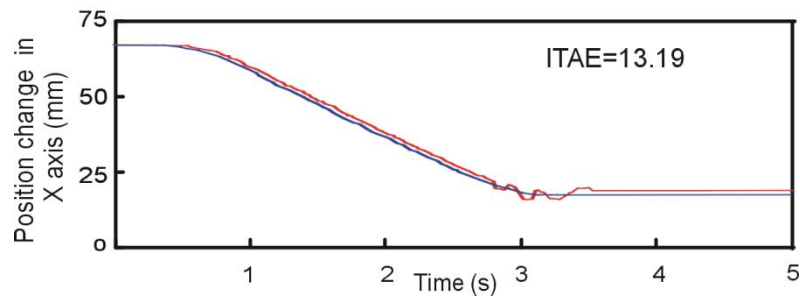
K_{PI} from 0.015 to 0.02 and the ITAE becomes to $13.19mm \cdot s^2$, which is increased compared with $K_{PP} = 0.021, K_{PI} = 0.015$ (Figure 6.7 (c) and Figure 6.6(b)).



(a) System behaviour with $K_{PP} = 0.021, K_{PI} = 0.004$



(b) System behaviour with $K_{PP} = 0.021, K_{PI} = 0.012$



(c) System behaviour with $K_{PP} = 0.021, K_{PI} = 0.02$

Figure 6.7 Changes of trajectories of both passive tools with different integral gains. Blue curves are the trajectory of the tool mounted on the X-Y table and red curves are that of the tool on the robot arm.

2. Performance at different speeds

Using the best combination of gains ($K_{PP} = 0.021, K_{PI} = 0.015$) from previous tests, it was necessary to establish how the speed of the X-Y table affects the system

response. The experimental procedure was the same as previous, but the X-Y table was moved along the X axis over a range of 10-80mm/s for 100mm. The performances of the method at different speeds are illustrated in Figure 6.8. It can be seen that the robot has a better response when the table is moving at speeds less than 30mm/s (ITAE is round $11mm \cdot s^2$), with a progressive deterioration in ITAE with increasing speed (ITAE increases to $29.8mm \cdot s^2$ at 80mm/s).

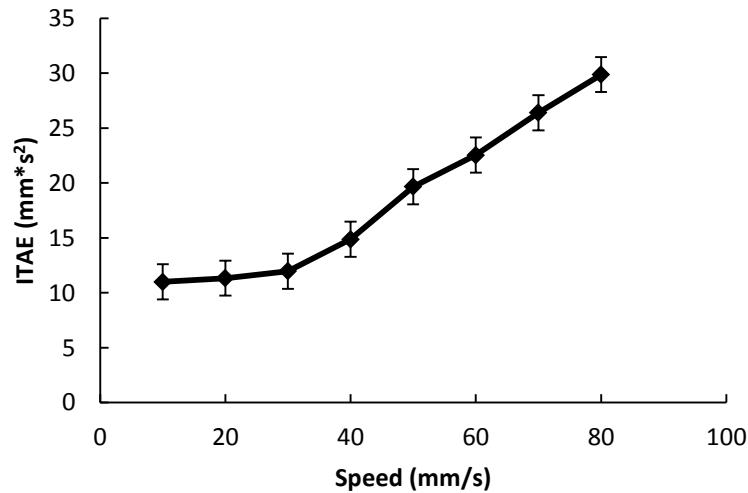


Figure 6.8 System performance at different speeds. The error bars show the standard derivation.

It can be seen from the test results that if the gains are too low, the control action of the system is poor when responding to system disturbances, which results in a less responsive controller and reduced tracking performance (Figure 6.6(a)). In contrast, a higher gain results in a more responsive controller, however, the system can become less stable if the gains are too high (Figure 6.6(c)). The response of the system is affected by the speed of the disturbances. The system can track the X-Y table's movement with an ITAE of around $12mm \cdot s^2$ when the speed is less than 40mm/s (Figure 6.8), however, the ITAE increases significantly when the speed was more than 40mm/s. The speeds recorded for head movements are generally less than 20mm/s (Chapter 3), and thus the gains $K_{pp} = 0.021, K_{pl} = 0.015$ are suitable for these tasks, and they are selected for the control. The steady state error of the system is about 0.8mm; this error includes the error of the coordinate system registration

which is about 0.6mm. Therefore, the error of the control method should be better than 0.3mm.

6.5 Testing of the combined position and force controller

Tests were undertaken in order to establish the performance and behaviour of the combined force and position controller, to check the feasibility of the control method with regards to stability, accuracy and any interaction between the position and force control loops. In these tests, the force controller is applied on the Z axis of the tool coordinate system and the position controller is applied on the other five axes.

6.5.1 Head motion simulator

A six degrees of freedom (DOF) PUMA 560 robot arm was used to manipulate an indoor football and provide disturbance motions or reaction forces during the tests. The football has a thick, soft fibrous skin and was mounted on the wrist of the PUMA to simulate the subject's head (Figure 6.9).

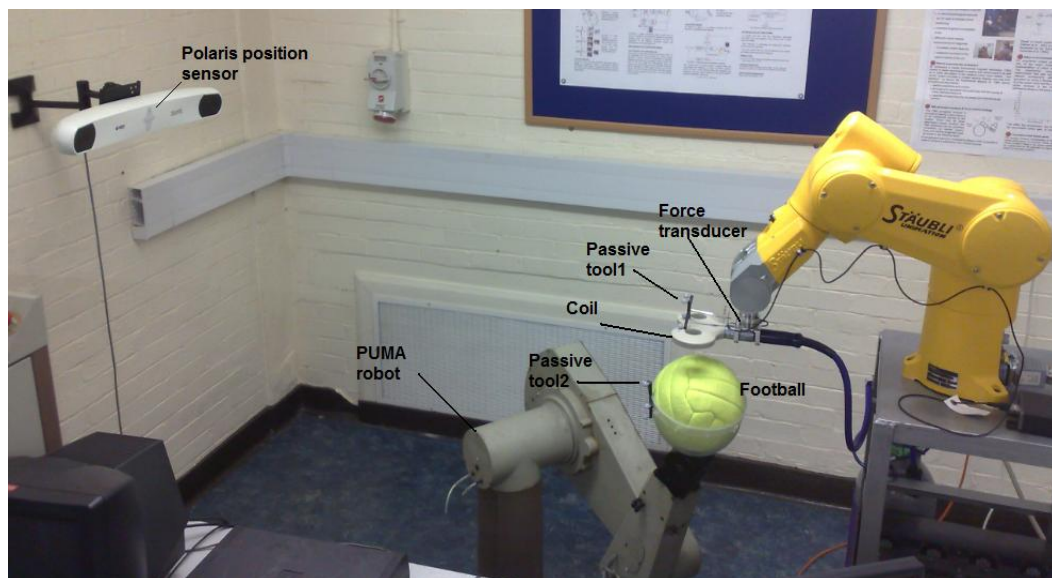


Figure 6.9 Test rig for the combined position and force controller.

The base of the PUMA robot arm was mounted on the floor at an appropriate distance from the Staubli robot in order to provide a suitable range of movement for both robots. The TMS figure-of-eight shaped coil is mounted on the Staubli robot and the F/T force sensor is placed between the coil and robot tool. Passive *tool2* is

attached to the ball to define its coordinate system, and *tool1* is mounted on the coil to establish the relationship between the ball and coil. The Polaris position sensor is fixed on the wall of the working cell to track the motion of the two passive tools. An independent PC is connected to the PUMA robot controller via an RS232 serial connection over which commands can be transmitted.

Since the application involves force control, the overall contact ‘environment’ stiffness was measured. This was established by moving the end-effector of the Staubli robot in the Z axis and making contact with the ball on the PUMA robot without force feedback. The PUMA robot was maintained stationary throughout this procedure, and contact force and position coordinates were recorded during the movement.

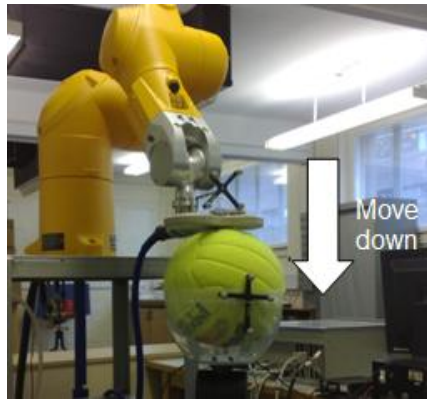


Figure 6.10 Staubli robot moves down and makes contact with the ball.

The Staubli robot was moved 1mm in 10 increments (0.1mm per step), and 10 force data were recorded at each step. The relationship between displacement and force is illustrated in Figure 6.11. The mean force at each position is plotted and the error bar shows the standard deviation. The calculated contact stiffness K_{stf} is 10.2N/mm. Note that this is the effective stiffness of the overall system and includes both arms, the ball and coil.

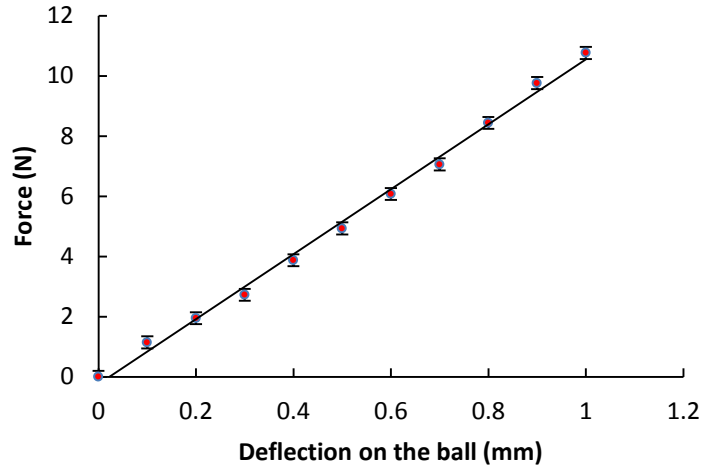


Figure 6.11 Relationship between the force and deflection in the Z axis. Error bars show the standard deviation.

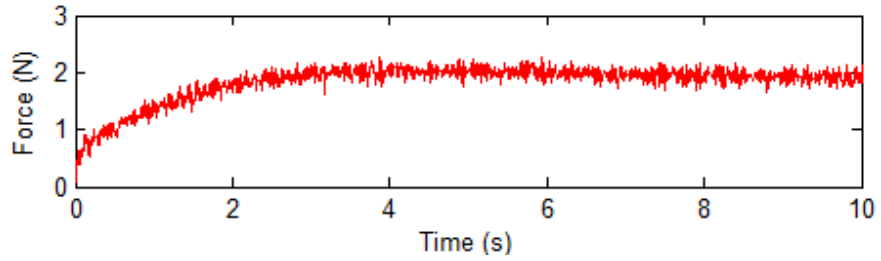
6.5.2 Experimental design and results

Experiments were carried out to examine the proposed position-force control method. The relative position changes of both passive tools were used to evaluate the performance of the tracking control, as they were both collected by the position sensor and this can ensure the data was recorded in the same conditions. To simulate a disturbance motion, a speed of up to 20mm/s is programmed in the PUMA robot, which is faster than the general speed of movements of a human (Chapter 3). As preliminary test on position control has been conducted previously, four major tests were designed in this section: the first test is step response test on Z axis to establish the ‘optimally’ gains for force control. Test 2 shows the limitation of force control using constant gains. Gain scheduling method is proposed and tested in Test 3 to establish a set of optimal gains. Finally, the combined position and force controller was tested in Test 4 in terms of disturbance occurs in XY plane, X,Y and Z axes, and orientation axes.

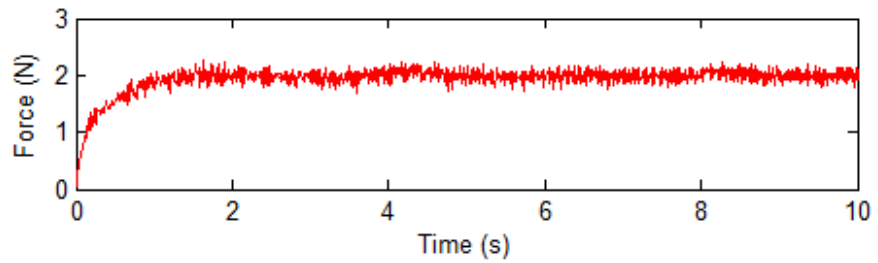
Test 1: Step response test

The purpose of this test was to ‘optimally’ tune the proportional and integral gains for the force control when the ball was stationary; the force was applied along the Z axis of the coil frame. A force set point of 2N was selected for the force control. Full

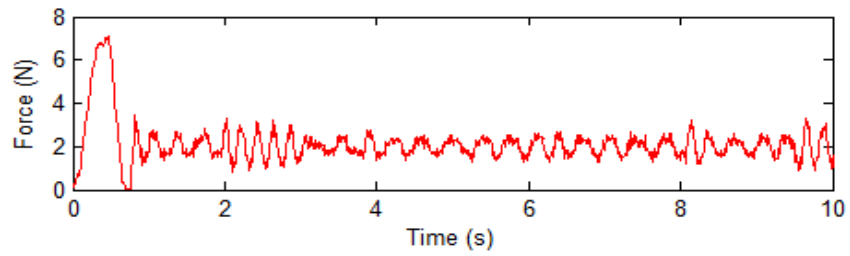
factorial tests were carried out to find the regions of both gains, selected results were demonstrated here to show the system performance.



(a) $K_{FP} = 0.005, K_{FI} = 0.002$



(b) $K_{FP} = 0.012, K_{FI} = 0.002$



(c) $K_{FP} = 0.02, K_{FI} = 0.002$

Figure 6.12 Effect of proportional gain on force step response.

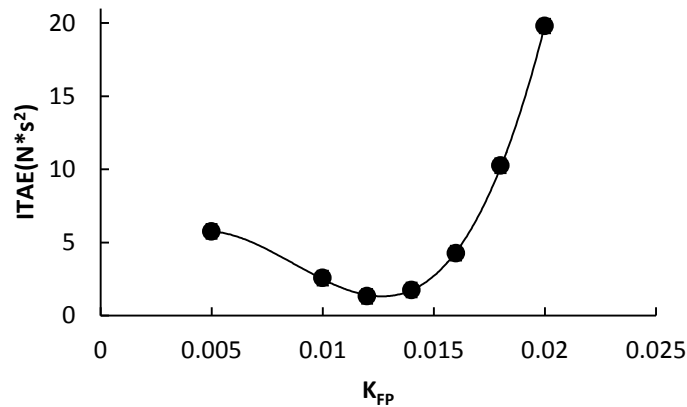


Figure 6.13 ITAE of different K_{FP} . Error bar is the deviation.

Figure 6.12 illustrates the step response for varying proportional gains K_{FP} , with a fixed integral gain $K_{FI} = 0.002$. It can be observed that the rise time improves with increasing gain. With $K_{FP} = 0.012$, inasmuch that rise time is of the order of 1.2 seconds, with a steady state error of 0.4N. When $K_{FP} = 0.02$, the response is oscillatory (Figure 6.12(c)), with a large initial overshoot, enters a limit cycle, and is deemed close to being unstable. The ITAE of more different proportional gains are illustrated in Figure 6.13, and it can be seen that the response is ‘optimised’ when $K_{FP} = 0.012$, and the value increases after this as the system response becomes less stable.

The step responses for different integral gains K_{FI} are illustrated in Figure 6.14. With $K_{FI} = 0.001$ (Figure 6.14(a)), the rise time is about 2 seconds without overshoot; the mean force error in the steady state is less than 0.4N. However, when K_{FI} is increased to 0.004 there is a significant overshoot with no further improvement in rise time or steady state error (Figure 6.14 (b)). The minimum ITAE value is obtained when $K_{FI} = 0.002$, and it increases gradually with the increment of K_{FI} (Figure 6. 15).

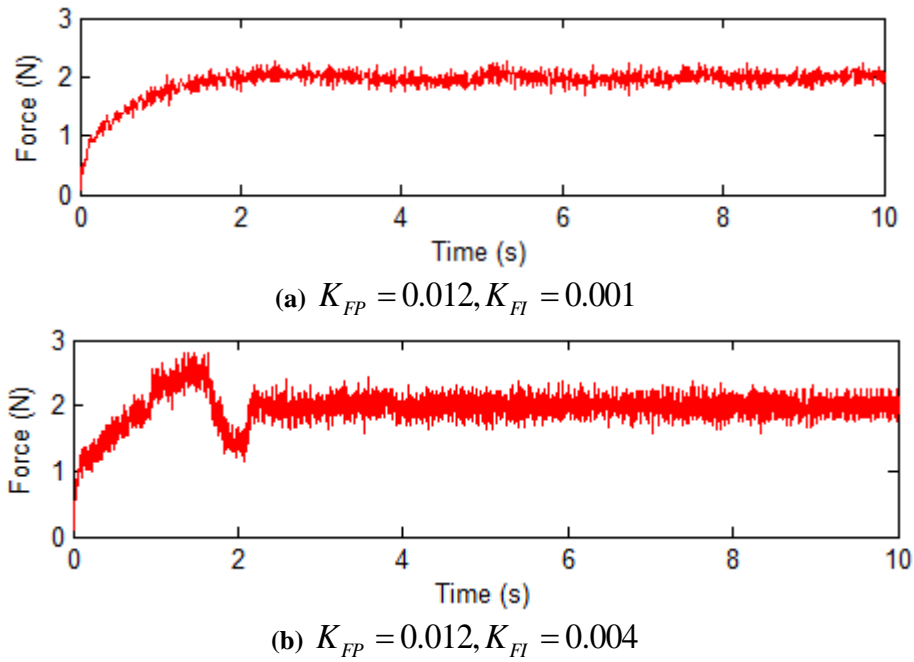


Figure 6.14 Effect of integral gain on force step response.

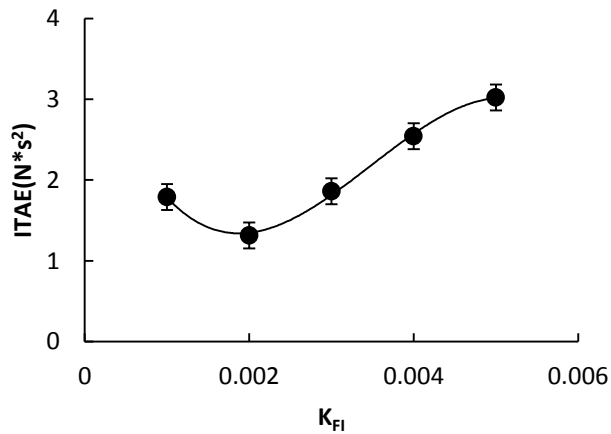


Figure 6.15 ITAE of different K_{FI} . Error bar is the deviation.

Test 2: Force control with disturbance input using constant gains

This test was carried out to establish the tracking performance of the force controller, when the ball is moved along the Z axis a prescribed distance at varying velocity. The force control set point was again fixed at 2N, and the ‘optimised’ gains of $K_{FP} = 0.012$ and $K_{FI} = 0.002$ were used. The sequence was as follows: first, the coil was moved to a position close to the ball (but not in contact), and then the force control established. Once the contact force had reached the steady state, the PUMA

robot moved the ball 10mm (the distances recorded for head movements are generally around 10mm (Chapter 3)) towards the Staubli robot in the Z direction (as the red arrow shows in Figure 6.16(b)) at a speed of 20mm/s and then was moved back to the same start position at the speed.

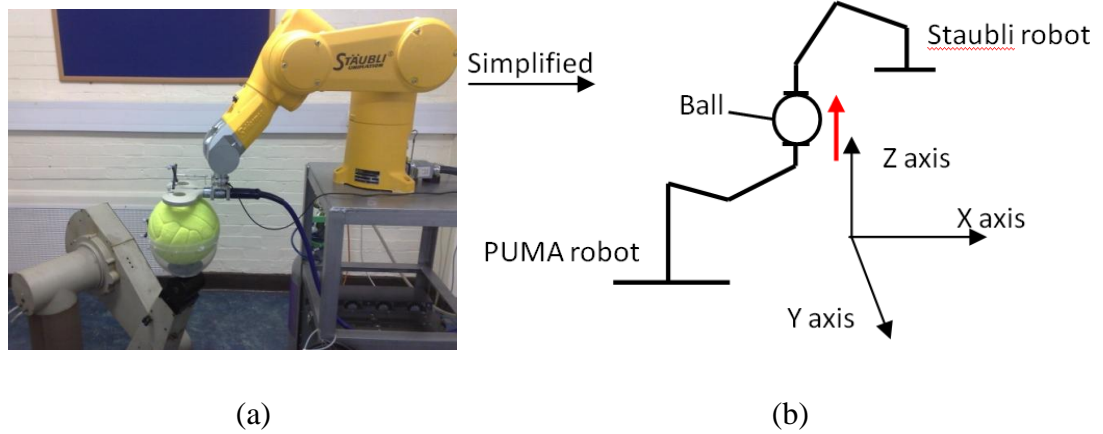


Figure 6.16 PUMA moves the ball in Z axis. (b) is a simplified picture from (a).

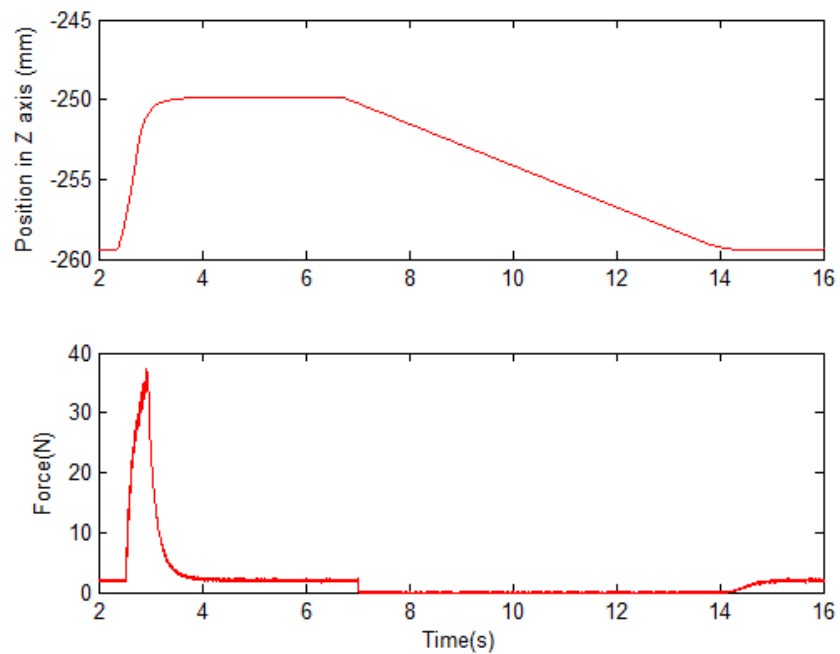


Figure 6.17 Force control with constant gains $K_{FP} = 0.0012$ $K_{FI} = 0.002$.

Figure 6.17 shows the trajectory and contact force of the Staubli robot. It can be seen that the force increases to $>30\text{N}$ when the ball is moved towards the Staubli robot, and contact is totally lost when the ball is moved away from the coil. From this it can

be deduced that the response of the force controller is totally inadequate, as the maximum velocity of the Staubli robot appears to be restricted to approximately 1.5mm/s, as evidenced by its finite velocity over the time period 7-14 seconds.

In order to improve the responsiveness of the system, an increased proportional gain $K_{FP} = 0.025$ was set for the same test. Figure 6.18 illustrates the force control responses to disturbance input with constant gains $K_{FP} = 0.025$, $K_{FI} = 0.002$, in which a faster responsive time was achieved. When the PUMA was moved towards the coil, the contact force increased to 5N (an error of 3N) and the time taken to accomplish the movement was reduced to 0.5s. However, when the PUMA moved away from the coil the contact force was reduce to around 0.5N (contact lost in some occasion). Note that even in the steady state an oscillatory perturbation in the contact force is present as a consequence of using an increased gain.

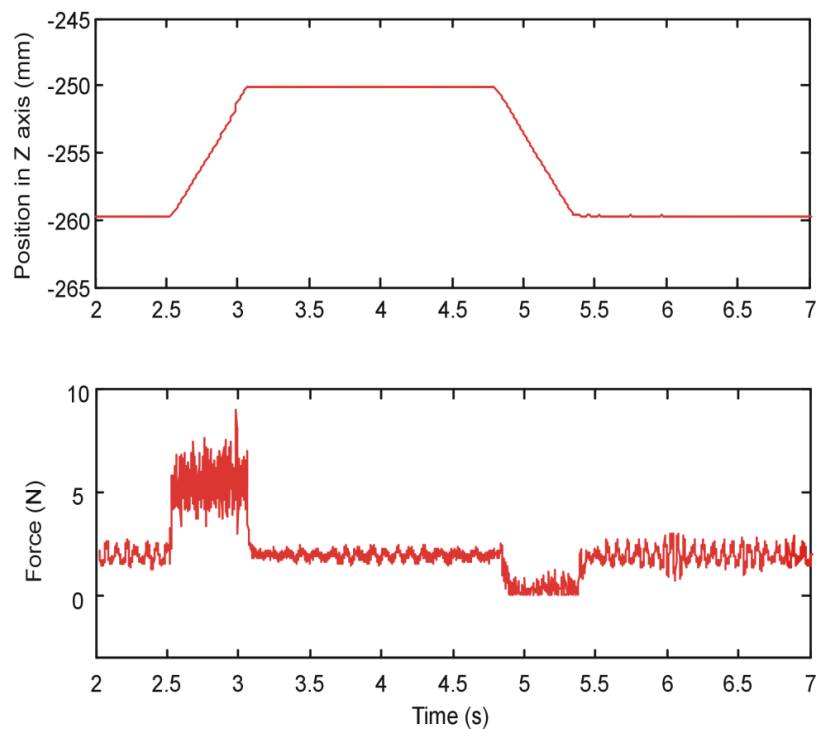


Figure 6.18 Force control with larger constant gains.

These results indicate that force control using constant gains lead either to a less responsive or a less stable force response, thus this is inadequate for the force control in the application. In order to improve the performance of the force control, gain scheduling method is proposed as follows.

Test 3: Force control using gain scheduling

Gain scheduling is an effective way of controlling systems whose dynamics change with the operating conditions. There is conventional gain scheduling (CGS) [Shaohua et al., 1997], fuzzy gain scheduling (FGS) [Zhao et al., 1993], neural network gain scheduling (NNGS) and neural fuzzy gain scheduling method (NFGS) [Shaohua et al., 1997]. NNGS and NFGS utilize the learning capabilities of the neural network or fuzzy neural network so that the controller can schedule the control parameters without prior knowledge of the changing dynamic conditions. These are sophisticated methods and may involve extensive computation to estimate the control parameters, which may result in a slow system response. Zakaria is working on the force control using the NFGS [Zakaria, expecting submit in 2012]. This application focuses on the design of CGS force control; a schematic diagram of the CGS control system is illustrated in Figure 6.19. The main advantage is that it is easier to implement and the parameters can be adjusted easily in response to the changing dynamics, as the parameters are pre-defined and no parameter estimation is required during the control process. One major disadvantage of CGS is that the parameter change may be rather abrupt across the region boundaries, which may result in unstable performance across the transition regions.

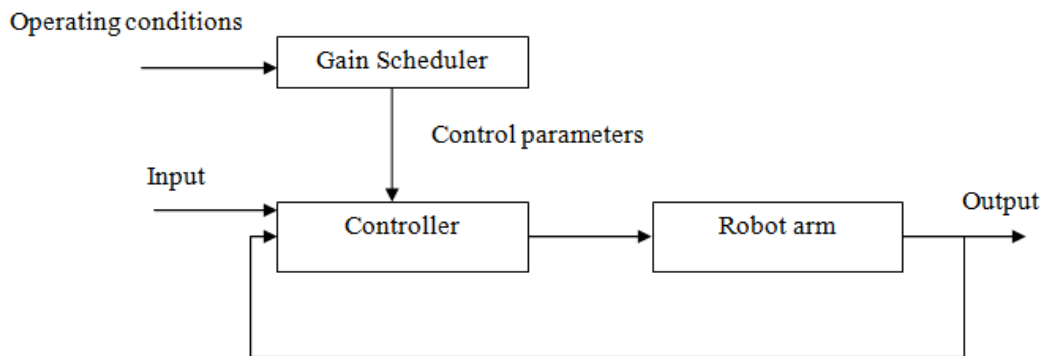


Figure 6.19 Conventional gain scheduled system.

The proposed CGS method involves changing K_{FP} as it has a major influence on response to a disturbance input, while K_{FI} has limited effect (described in following section). The equation for the gain scheduling method is expressed as:

$$K_{FP} \propto f(K_{stf}, T, E(T)) \quad (6.13)$$

where K_{FP} is the proportional gain, K_{stf} is the stiffness of the system, T is the period of one loop and $E(T)$ is the force error. K_{stf} and T can be considered fixed with respect to the system. The force error $E(T)$ is divided into several intervals, and K_{FP} is determined accordingly. In order to simplify the process of tuning the gain K_{FP} , Eq.6.14 is used to establish the relationship between all the gains, once the force error intervals are confirmed.

$$K_{FP}(n) = K_{FP}(n-1) + K_c * K_{FP}(n-1) \quad n = 1, 2, 3, \dots \quad (6.14)$$

where $K_{FP}(n)$ is the gain used for the n^{th} interval, n is the number of intervals. K_c is the gain used to calculate the $K_{FP}(n)$. $K_{FP}(0)$ is the gain obtained when the system is in the steady state. Table 6.1 presents an example of the gains with $K_c = 0.1$, with an interval chosen as 1N. This is estimated by multiplying the resolution of PUMA, to the stiffness: $0.1\text{mm} * 10.2 \text{ N/mm} = 1.02\text{N}$, and is the smallest force change when the PUMA robot moves one step.

Table 6.1: An example of gain scheduling for force control

F_e	K_{FP}
$0.5 \leq F_e < 1.5$	0.0132
$1.5 \leq F_e < 2.5$	0.01452
$2.5 \leq F_e < 3.5$	0.015972
$3.5 \leq F_e < 4.5$	0.017569
$4.5 \leq F_e < 5.5$	0.019326
$5.5 \leq F_e $	0.021259
....

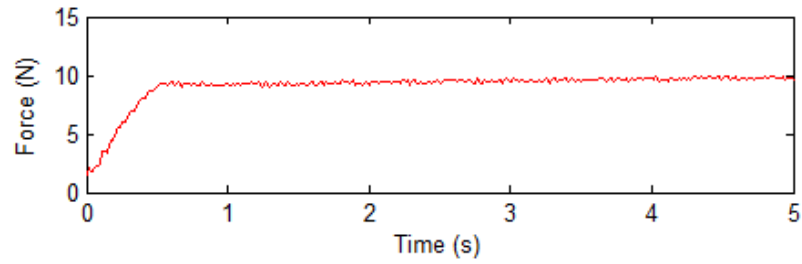
The gain scheduling method is first tested on the stationary ball. The sequence is: the Staubli robot was initially driven by the force set point of 2N, after attaining the steady state, the force set point is increased to 10N. The force response of the system

was evaluated using both the integral of time multiplied by absolute error (ITAE) and integral absolute error (IAE). Different values of K_c with $K_{FP}(0) = 0.012$ and $K_{FI} = 0.002$ were used in the test, and the gains are illustrated in Tables 6.2.

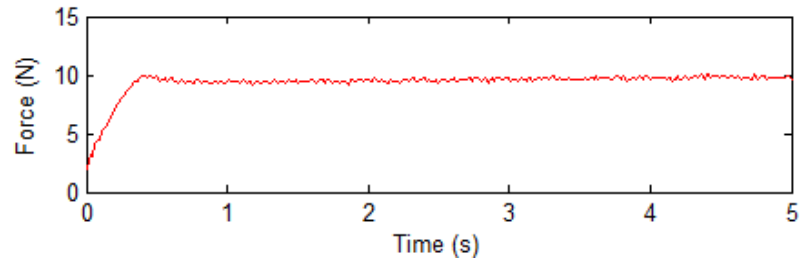
Table 6. 2: Gain scheduling for force control

F_e	K_{FP} when $K_c = 0.2$	K_{FP} when $K_c = 0.25$	K_{FP} when $K_c = 0.3$	K_{FP} when $K_c = 0.35$	K_{FP} when $K_c = 0.4$
$0.5 \leq F_e < 1.5$	0.0144	0.015	0.0156	0.0162	0.0168
$1.5 \leq F_e < 2.5$	0.01728	0.01875	0.02028	0.02187	0.02352
$2.5 \leq F_e < 3.5$	0.020736	0.023438	0.026364	0.029525	0.032928
$3.5 \leq F_e < 4.5$	0.024883	0.029297	0.034273	0.039858	0.046099
$4.5 \leq F_e < 5.5$	0.02986	0.036621	0.044555	0.053808	0.064539
$5.5 \leq F_e $	0.035832	0.045776	0.057922	0.072641	0.090354

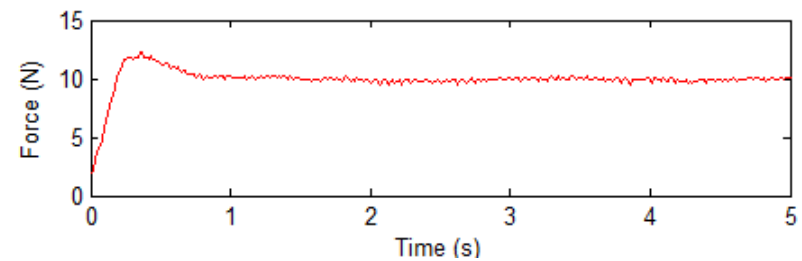
The system response is shown in Figure 6.20(a)-(e) respectively. Figure 6.20(a) illustrates the system response when $K_c = 0.2$, the system rise time is about 0.5 second which is significantly faster than that with the constant gain method. In addition, no overshoot occurs after it reached the steady state. As the gain K_c increases to 0.25, a faster response (0.2 seconds) is obtained and slight overshoots appear when $K_c = 0.3$ and $K_c = 0.35$, but the system returns to a steady state quickly after this (Figure 6.20 (c) and (d)). Any further gain increases ($K_c = 0.4$ in Figure 6.20(e)) causes even more overshoot.



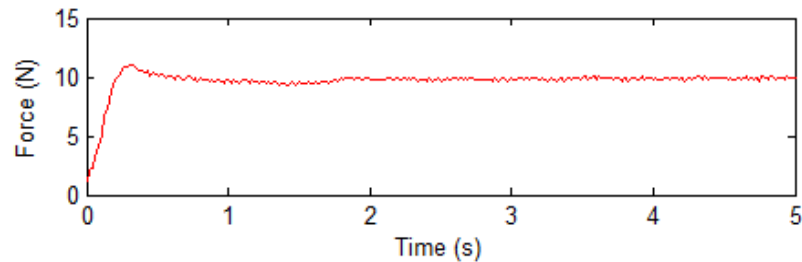
(a) System force response with $K_c = 0.2$



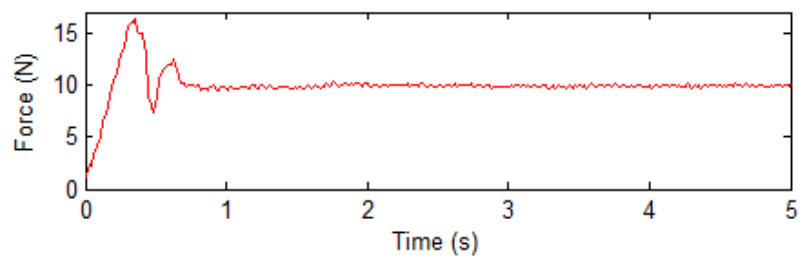
(b) System force response with $K_c = 0.25$



(c) System force response with $K_c = 0.3$



(d) System force response with $K_c = 0.35$



(e) System force response with $K_c = 0.4$

Figure 6.20 Force response of the system with different K_c .

ITAE was calculated for the time interval $[0,1]$, as force error changed significantly with the increment of time during this period; IAE was calculated for time interval $[1,5]$ where the system is in steady state (Table 6.3 and Figure 6. 21). It can be seen

that the value of ITAE is $0.7118 N \cdot s^2$ when $K_c = 0.2$, and the value is decreased to about $0.4 N \cdot s^2$ where $K_c = 0.25$ and $K_c = 0.3$. The smallest value of ITAE is obtained when $K_c = 0.35$, and it increases to $0.7034 N \cdot s^2$ when $K_c = 0.4$, as the overshoot contributes to the ITAE. The IAE has a similar behaviour with these different gains. A bigger value ($1.9919 N \cdot s$) is obtained with $K_c = 0.2$, and the smallest value of IAE is attained when $K_c = 0.3$ (Figure 6. 21).

Table 6.3 ITAE and IAE of the system force response with different K_c when the ball is stationary

K_c	0.2	0.25	0.3	0.35	0.4
ITAE($N \cdot s^2$) (from 0-1s)	0.7118	0.3958	0.4056	0.2181	0.7034
IAE($N \cdot s$) (from 1-5s)	1.9919	1.6152	0.5937	0.6102	0.575

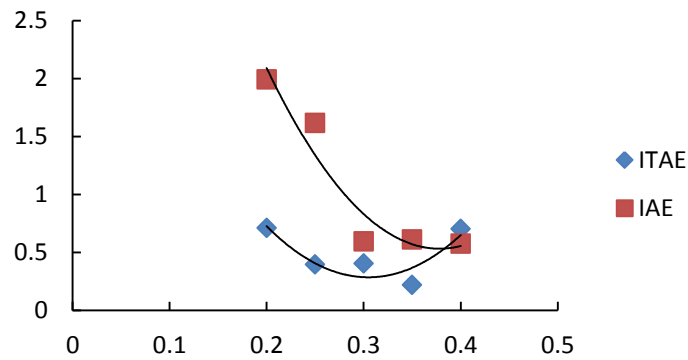


Figure 6. 21 ITAE and IAE using different K_c .

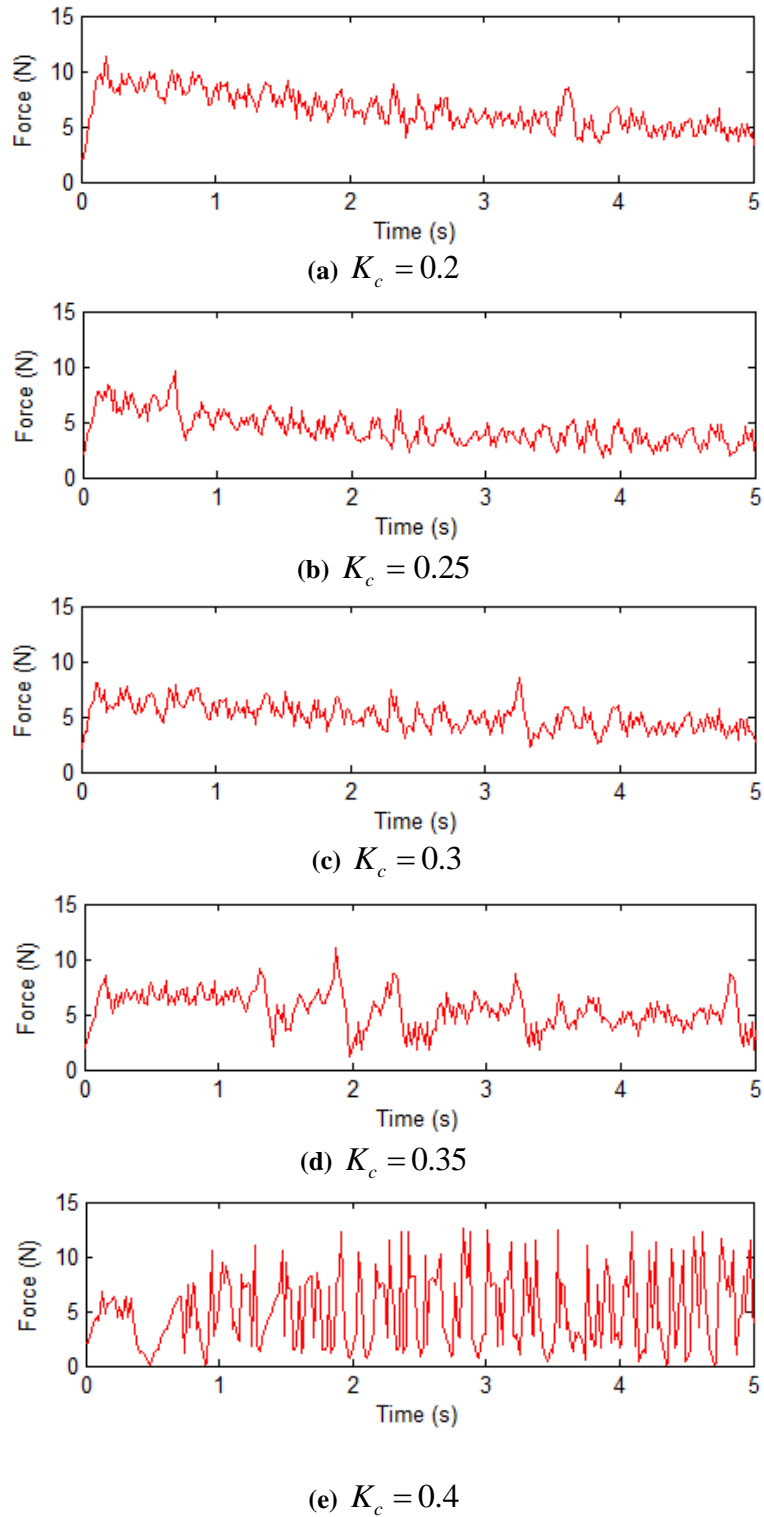


Figure 6.22 System response on the moving ball with different gains.

The previous test suggests that both $K_c = 0.3$ and $K_c = 0.35$ could be the optimal gain for control. In order to confirm it, the system was then tested using the gain scheduling force control at a set point of 2N on the moving ball, with proportional

gains from Tables 6.2 and constant integral gain of $K_{FI} = 0.002$. The Staubli robot was moved into contact with the stationary ball, and a contact force of 2N established. The PUMA robot was then moved towards the coil at a speed of 20mm/s. The actual contact forces were recorded during the movement for different K_c gains. Figure 6.22 illustrates the system responses to the moving ball. As illustrated in Figure 6.22(a), for a low value of gain $K_c = 0.2$ the system response is relatively slow. As motion begins, the contact force increases to 12N, and then gradually reduces and is maintained at 5N. When the gain K_c is increased to 0.25 (Figure 6.22(b)), a faster response is obtained and the force attempts to stay constant at 4N after a quick rise at the beginning. When K_c is set to 0.3 as in Figure 6.22(c), the system response is similar to $K_c = 0.25$. Further gain increases ($K_c = 0.35$ in Figure 6.22(d)) cause greater oscillation during the movement, and the system becomes less stable and large amplitude vibrations occur when $K_c = 0.4$ (Figure 6.22(e)). The ITAE against K_c is shown in Table 6.4 and Figure 6. 23. Initially the value is decreased with the increment of K_c , and a minimum value of ITAE is $39.14 N \cdot s^2$ when $K_c = 0.3$. After this point, the ITAE value starts to increase again and it reaches $50.54 N \cdot s^2$ when $K_c = 0.4$. The trend of the ITAE values is consistent with the results found previously.

Table 6.4 System responses with different gains

Kc	0.2	0.25	0.3	0.35	0.4
ITAE($N \cdot s^2$)	54.51	41.9	39.14	44.26	50.54

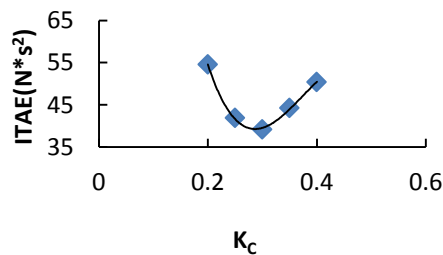


Figure 6. 23 ITAE using different K_c .

Same test as test 2 was carried out using gain scheduling method. The result is illustrated in Figure 6.24. Compared with the results shown in Figure 6.18, the maximum force error reduces to 5N when the ball moves towards the coil, and the system returns steady state after the ball stops. This indicates the improvement on force control using the gain scheduling method.

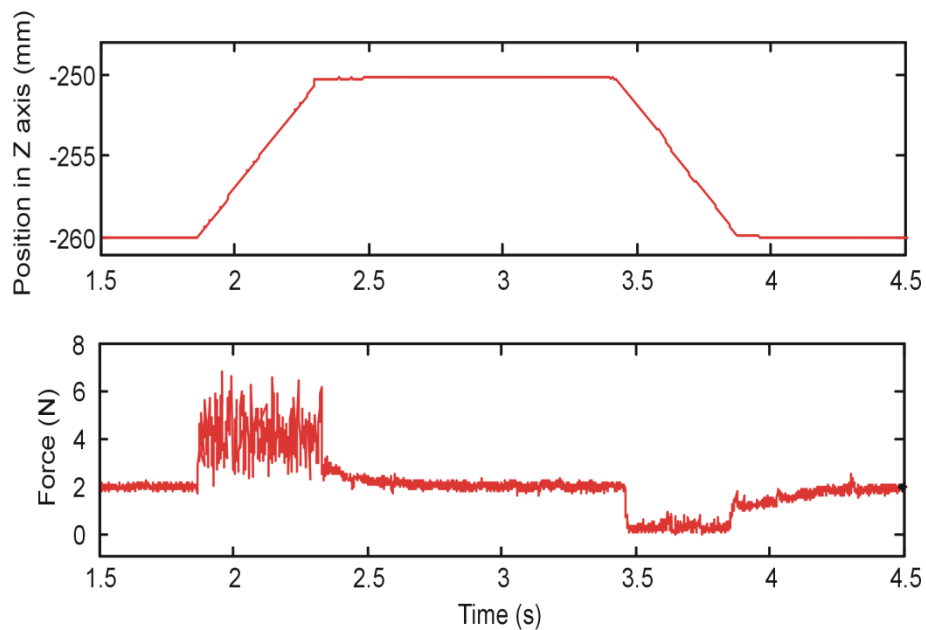


Figure 6.24 Force control in Z axis using gain scheduling method.

Test 4: Combined position and force control

This test was designed to demonstrate the behaviour of the proposed controller and the interaction between the axes controlled either by force and position controller. Force control is applied on the Z axis using the gain scheduling method with $K_c = 0.3$, whereas the remaining five axes are position controlled with $K_{pp} = 0.021, K_{pl} = 0.015$.

1. Disturbance occurs in XY plane

In this test, a disturbance was applied in the position controlled directions. The control force was set to $F_d = 2\text{N}$. The PUMA robot moves the ball in the XY plane

(red arrow in Figure 6.25) a distance of 10mm at a speed of 20mm/s. Longer distance can be achieved and found the performance is consistent, results of longer distance is in Appendix 2. 10mm is used here because the distances recorded for head movements are generally around 10mm (Chapter 3).

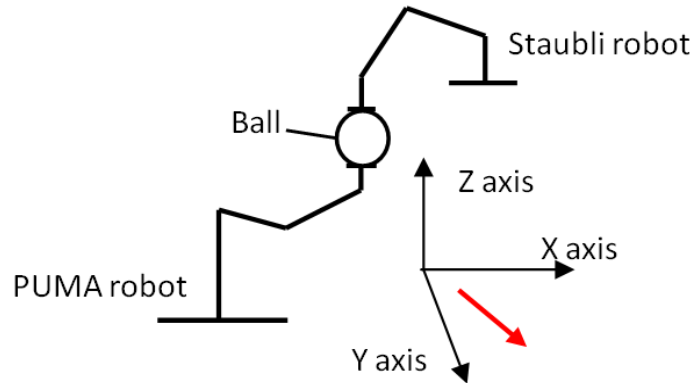


Figure 6.25 PUMA moves the ball in XY plane.

The trajectory of the Staubli robot and the resulting force response is shown in Figure 6.26. The position of the Staubli robot changes along the X-Y axis as it tracks the movement of the ball, however, there is also a slight perturbation in the Z axis (less than 0.5mm) with the force control attempting to maintain the 2N contact force. Regulation of the contact force is seen to be good during the steady state, however, there is a noticeable disturbance whilst the ball is moving. There is no change in orientations during the movement.

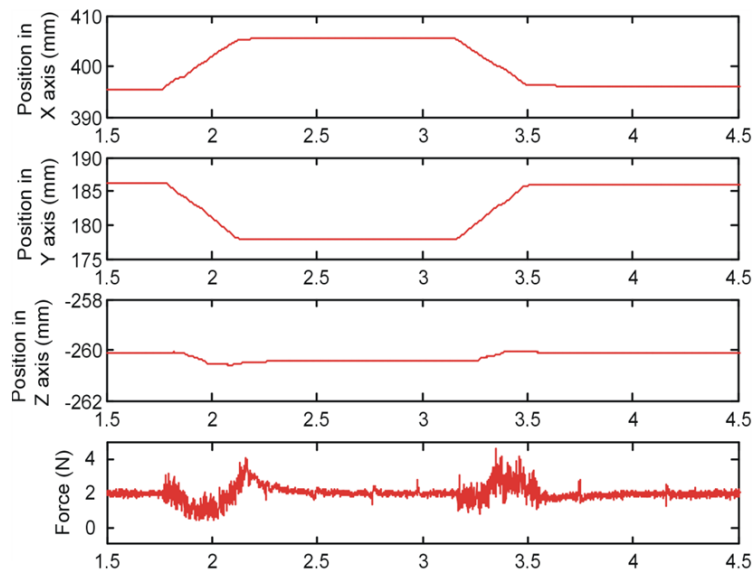


Figure 6.26 Behaviour of position and force controller with the position disturbance input.

A comparison of the trajectories of the ball and coil is shown in Figure 6.27. Whilst this demonstrates that both profiles match, there is a noticeable lag of 150-200ms between them. This is better than the 300ms reported in Richter's study [Richter L. et al., 2010]. The average root mean square (RMS) error during the movement is around 2.4mm, and the average RMS error in steady state is around 0.8mm.

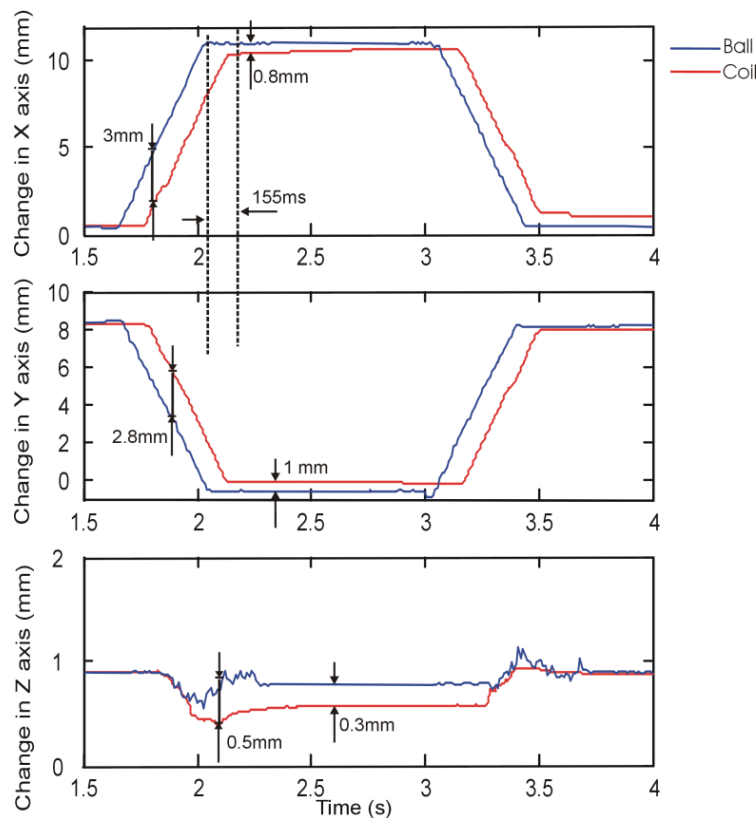


Figure 6.27 Comparison of changes of trajectories between the Staubli robot and ball. The error display on the picture shows the mean error during the movement and in steady state respectively.

In order to further evaluate the tracking capability and the interaction between the force and position controlled axes, the ball was moved around a circular path of 10mm diameter in XY plane, whilst attempting to maintain a contact force of 2N. The trajectories of both the ball and the coil are shown in Figure 6.28; the blue line is for the ball and the red line for the coil (Staubli robot). The Staubli robot (coil) tracks the circular movement of the PUMA (ball), with a marginal movement in the Z

direction. The average RMS error is 2.68mm during the movement which is slightly larger than the previous test. The steady state error is around 0.8mm.

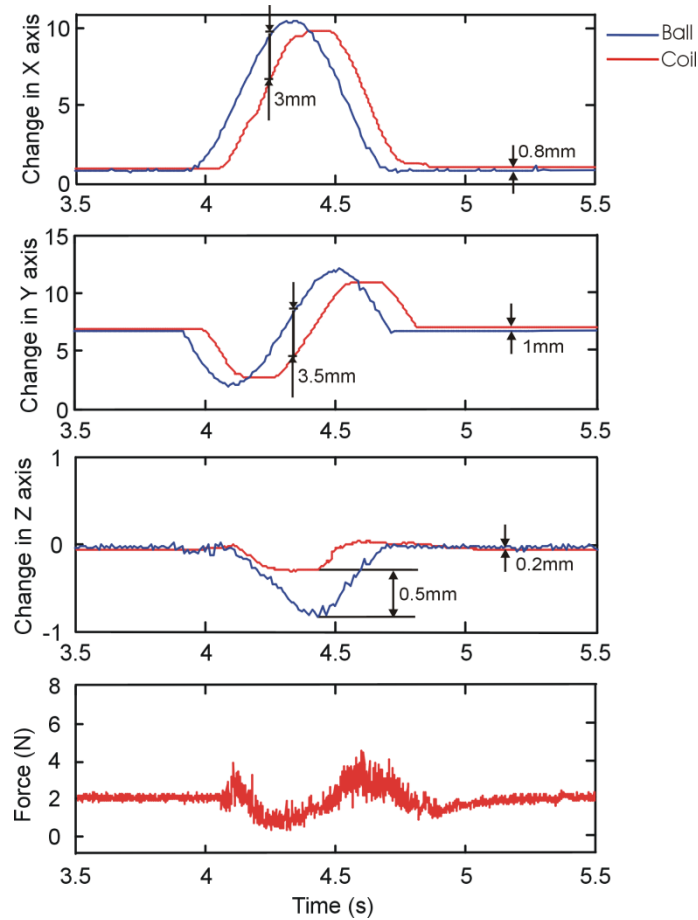


Figure 6.28 Profiles of trajectories when the disturbance is a circle. The error display on the picture shows the mean error during the movement and in steady state respectively.

The performance of the combined position and force controller on a long displacement is shown in Figure 6.29. A disturbance was applied in the position controlled directions. The control force was set to $F_d = 2\text{N}$. The PUMA robot moves the ball in the XY plane (red arrow in Figure 6.25) a distance of 100mm at a speed of 20mm/s. The system behaviour and performance are similar to the previous test using 10mm. The position of the Staubli robot changes along the X-Y axis as it tracks the movement of the ball, however, there is also a slight perturbation in the Z axis (less than 0.5mm) with the force control attempting to maintain the 2N contact force. The averaged RMS error during the movement is around 2.4mm and reduced to 0.8mm in steady state, which is consistent to previous tests.

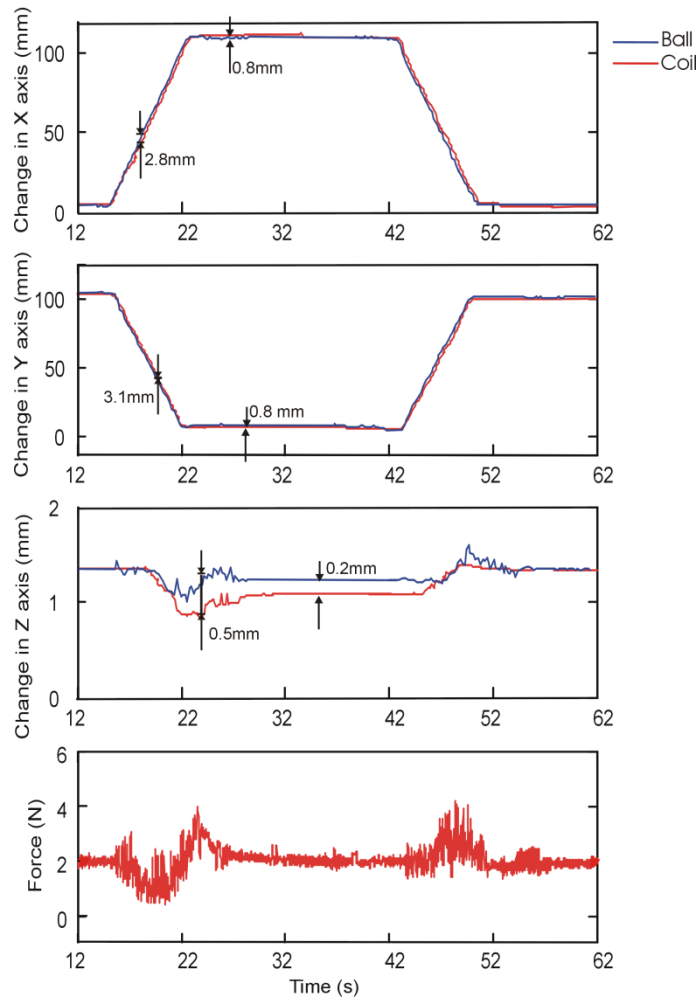


Figure 6.29 Trajectories when the system deals with 100mm displacement in X and Y axes. The error display on the picture shows the mean error during the movement and in steady state respectively.

2. Disturbance occurs in X,Y and Z axes

Previous tests have tested the response to the disturbances in the XY plane, this test was designed to evaluate the tracking performance when moving in all three directions, again with a set force of 2N in the Z axis. The experimental procedure was similar to the previous tests, but with the ball being moved a distance of 10mm in the X, Y and Z axes simultaneously at a speed of 20mm/s (Figure 6.30).

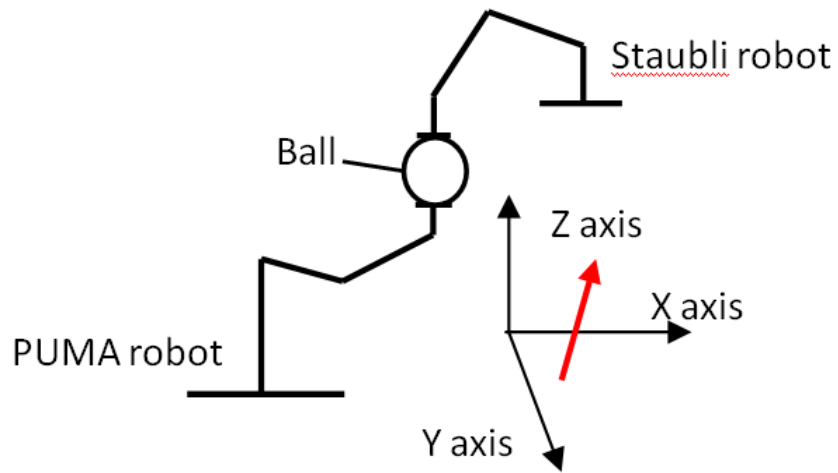


Figure 6.30 PUMA moves the ball in 3 direction simultaneously.

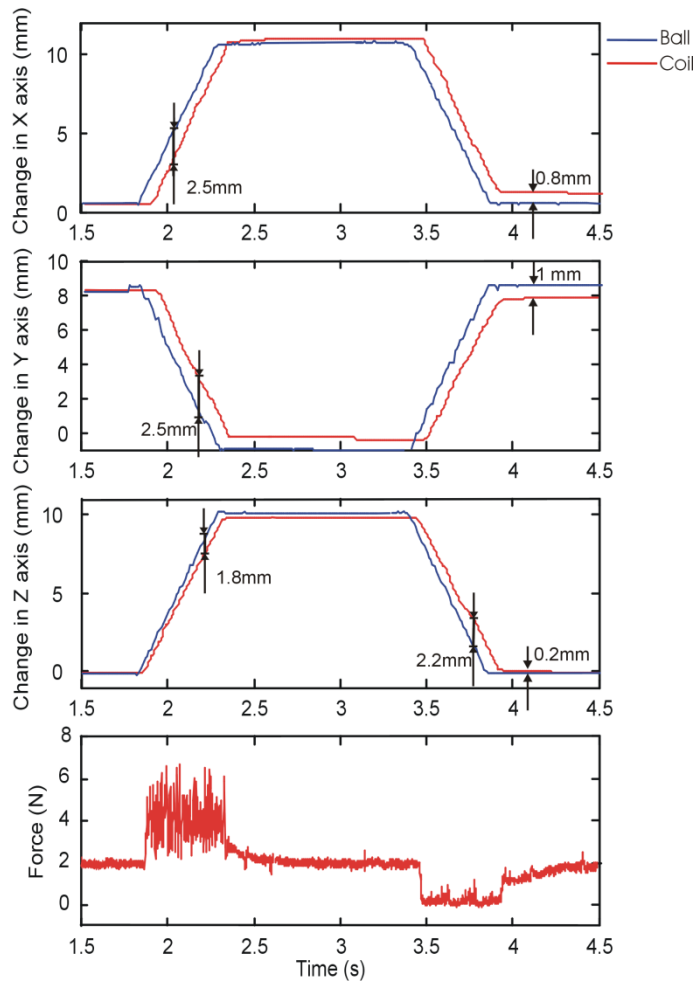


Figure 6.31 Combined position and force control with a desired force of 2N. The error display on the picture shows the mean error during the movement and in steady state respectively.

Figure 6.31 illustrates the tracking performance for both position and force, when the ball is moved along a straight line in Cartesian space. The trapezoidal trajectories in X and Y (top two figure) clearly illustrate the tracking performance, with an averaged position error of 2.5mm during the movement. This compares with the Z axis where it reduces to 1.8mm, and which is influenced by the combined position and force control. The desired 2N contact force is only maintained during the stationary dwell period. When the ball is moved towards the coil the contact force has a maximum force error of 4N with significant perturbations. In contrast, when the ball is moving away from the coil the contact forces appears to drop to zero which implies contact is lost and becomes intermittent.

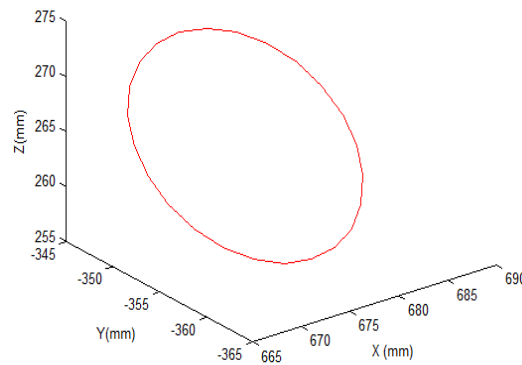


Figure 6.32 Circular path in 3D space.

Another test using a circular path of 20mm diameter (around 10mm in each axes, which is consistent to previous test) in Cartesian space (Figure 6.32) for the ball to test the tracking capability and the interaction between the force and position control of the system. The desired force was again set to 2N. The top three profiles show that the Staubli robot is able to track the movement along a curve in Cartesian space. The force increases to 6N at the beginning of the movement and is maintained around 4N until a change in direction occurs in the Z axis when it drops close to 0N. Stable force control is only achieved once motion has ceased.

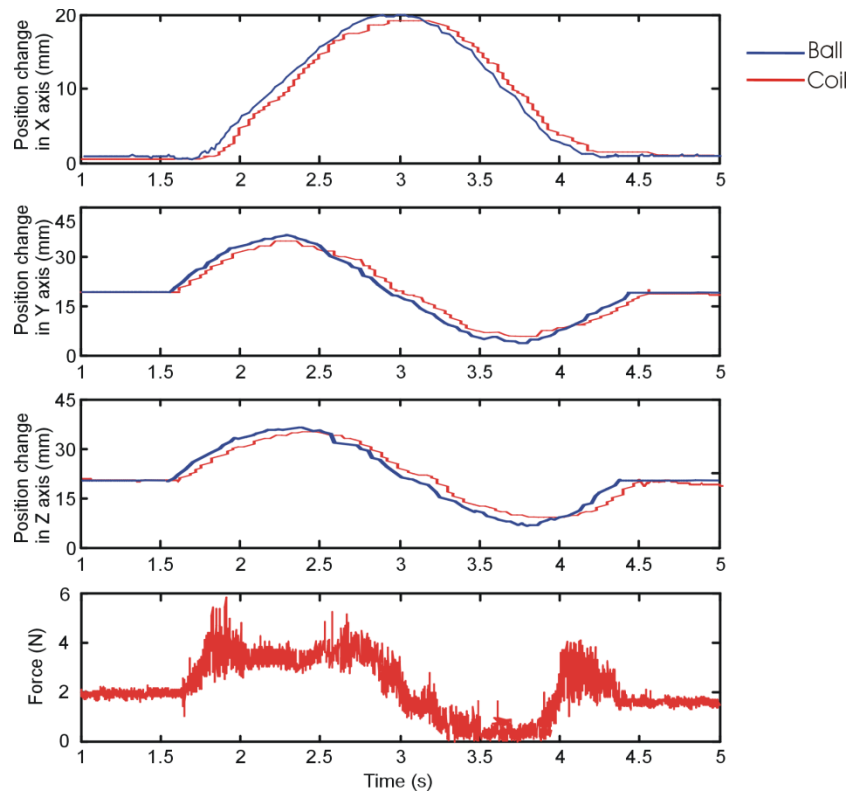


Figure 6.33 Position and force response to the 3D circle disturbance.

Another test was designed to evaluate the tracking performance when moving in all three directions, again with a set force of 2N in the Z axis and the ball moves a distance of 10mm in the X, Y and Z axes simultaneously at a speed of 20mm/s. The system behaviour again is consistent (Figure 6.34), with a similar profile and RMS errors (2.5mm during the movement, 0.8 in steady state). These results indicate the system performance is not affected by the movement of the distance.

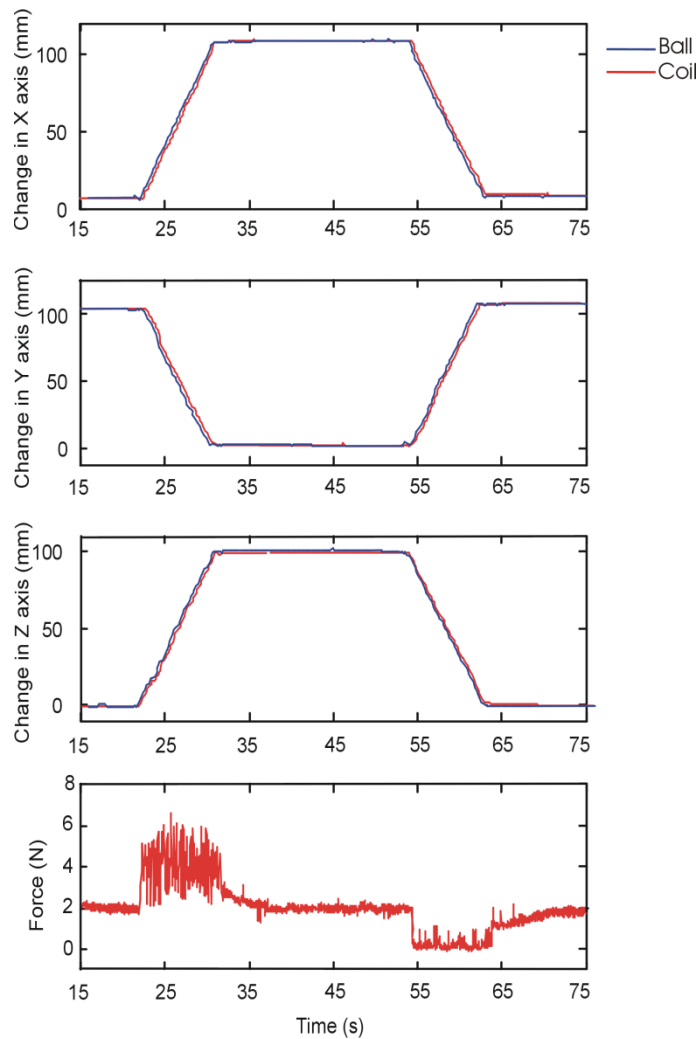


Figure 6.34 Trajectories when the system deals with 100mm displacement in X,Y and Z axes. The error display on the picture shows the mean error during the movement and in steady state respectively.

3. Disturbance in rotational axes

The previous tests were performed with a displacement disturbance but no change in orientation. This test was established to demonstrate the system behaviour when there was an orientation change. An orientation disturbance of the ball induces both changes in position and orientation to the coil, as the contact point to be maintained is not the tool centre point (TCP) of the PUMA robot (Figure 6.35) and the coordinate systems of both robots are not aligned. The TCP of PUMA was defined as the centre point of the tool flange which is about 40mm below the bottom of the ball.

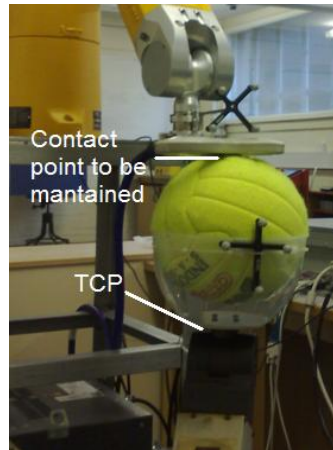


Figure 6.35 TCP of the PUMA robot.

The desired force was set to 2N, the PUMA robot was rotated through an angle of 10 degrees about its Y axis of the base frame, at a speed of 1rad/s. As illustrated in Figure 6.36, the top three profiles show the change in the X, Y and Z axes, the middle figure is the force response and the rest of them show the orientation about the X, Y and Z axes. Since the relationship is not linear between the movements of the ball and coil, the Staubli robot makes its movement in five directions in order to track the movement of the ball, and the maximum force error is about 2N.

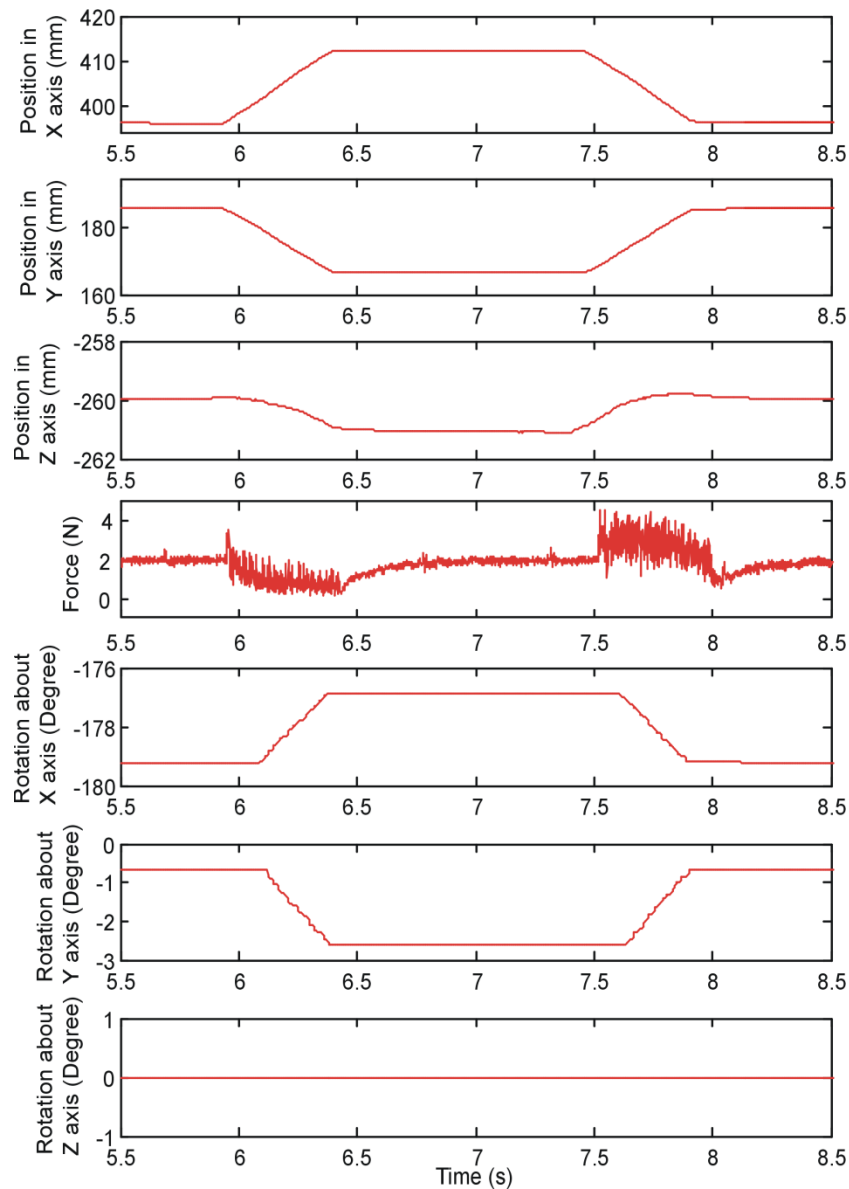


Figure 6.36 System behaviour when there is an orientation disturbance input. . All rotation axes are with respect to coil frame.

Since the relationship is not linear between the movements of the ball and coil, the changes in positions recorded by the position sensor for the ball and coil are not linear. This makes the error check for orientation more difficult to be performed than that of position. In order to conduct the error check, a test was carried out by aligning the Z axis of the PUMA robot with that of the Staubli robot, and a rotation of 10 degree was made about the Z axis. The trajectory of the coil is shown in Figure 6.37, the coil rotates about the Z axis and there is no noticeable effect on the other directions. The error of the rotation is about 0.5 degree in steady state.

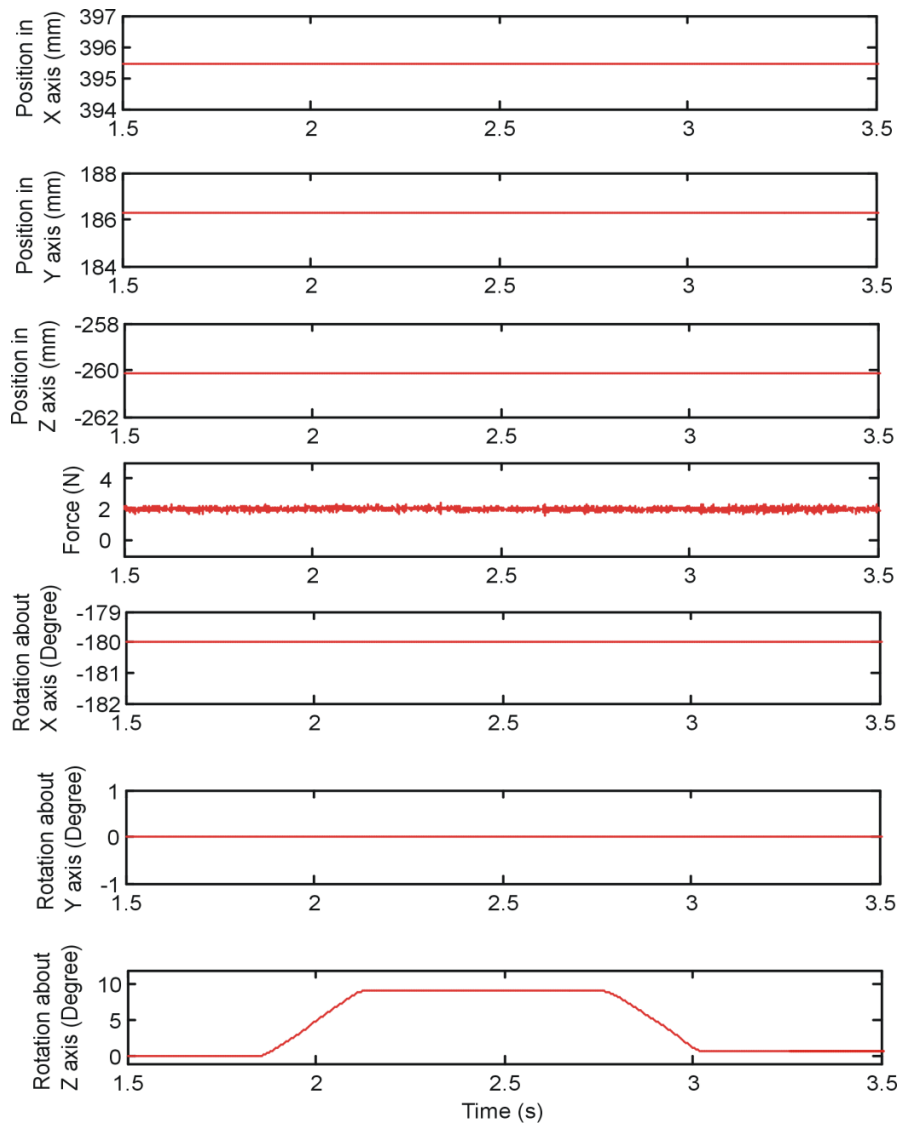


Figure 6.37 Trajectories of the Staubli robot followed by a rotation input about the Z axis. All rotation axes are with respect to coil frame.

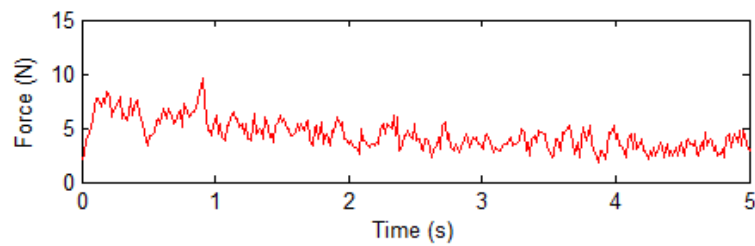
6.5.3 Practical considerations

The following practical considerations were encountered in the course of the above experiments:

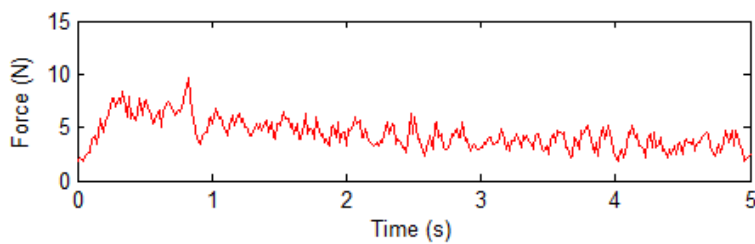
1. Effect of K_{FI} on force behaviour when using gain scheduling method

In this application, the gain scheduling method only adjusts the proportional gain K_{FP} . Although the integral term K_{FI} has an effect on accelerating the system to the steady state and reducing the steady state error, limited effect has been found when it is applied to the moving object. A similar test was carried out using procedure of test

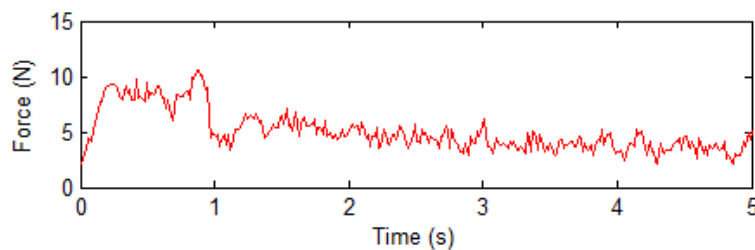
3, with $K_c = 0.35$ and $K_{FI} = 0.002, K_{FI} = 0.005, K_{FI} = 0.01$ respectively. The force responses are illustrated in Figure 6.38. A noticeable difference is that the contact force increases to 8N with a larger K_{FI} as motion begins (Figure 6.38(c)). The rest behaviour of these three responses are similar, which gradually reduces and is maintained at 5N. The ITAE values confirm that they have similar responses, which is $39.14 \text{ N} \cdot \text{s}^2$ ($K_{FI} = 0.002$), $39.19 \text{ N} \cdot \text{s}^2$ ($K_{FI} = 0.005$), $39.26 \text{ N} \cdot \text{s}^2$ ($K_{FI} = 0.01$).



(a) $K_{FI} = 0.002$



(b) $K_{FI} = 0.005$



(c) $K_{FI} = 0.01$

Figure 6.38 System force response using different K_{FI} .

2. Effects of K_{PP}, K_{PI} on the interaction between position and force control

Test 4 has shown that the position control has a slight effect on the force controlled axis. This section discusses how the gains used for position control affect the

interaction. Figure 6.39 illustrates the behaviour of the system when the PUMA robot moves the ball in X direction at a speed of 20mm/s and the Staubli robot tracks the movement with greater gains of $K_{pp} = 0.028, K_{pl} = 0.02$.

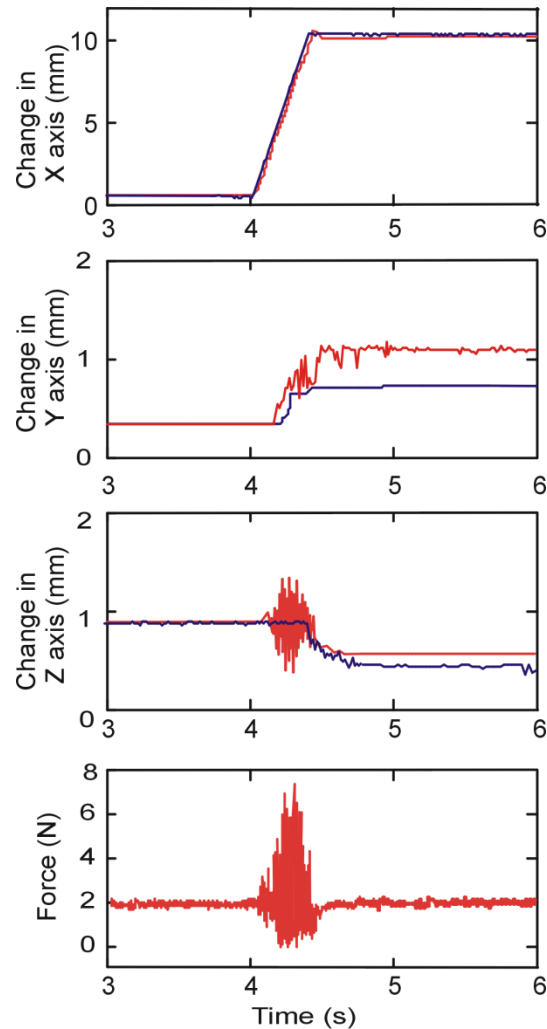


Figure 6.39 System behaviour when the Staubli robot tracks the movement of the ball with a fast response time.

It can be seen that the lag between the trajectory of the ball (blue line) and coil (red line) was reduced compared with the response illustrated in Figure 6.26 ($K_{pp} = 0.021, K_{pl} = 0.015$ were used). However, there is a significant vibration in the Z direction caused by the movement in the X direction and the maximum force error reaches 7N with significant perturbations. Slight oscillation also occurs in the Y axis during the movement by about 0.3mm. This result indicates that the greater

gains used in position control can reduce the lag between the movements of the ball and coil, however, the robot movement becomes unstable in Z axis.

3. Force response with different speeds

Another test was conducted using the method in test 3 with $K_c = 0.35, K_{PI} = 0.0015$, but with different speeds of 10mm/s, 20mm/s, 30mm/s, 40mm/s, 50mm/s and 60mm/s. The ITAE of the force responses with these speeds are illustrated in Figure 6.40. It can be seen that the robot has a better response when the ball is moving at speeds less than 40mm/s (ITAE is around $40\text{mm}\cdot\text{s}^2$), with a progressive deterioration in ITAE with increasing speed. The reason is the limitation of the gain scheduler is reached when the ball moves at a speed more than 40mm/s.

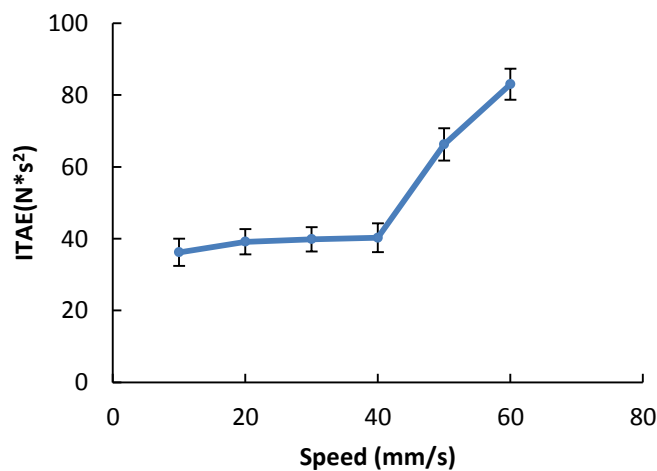
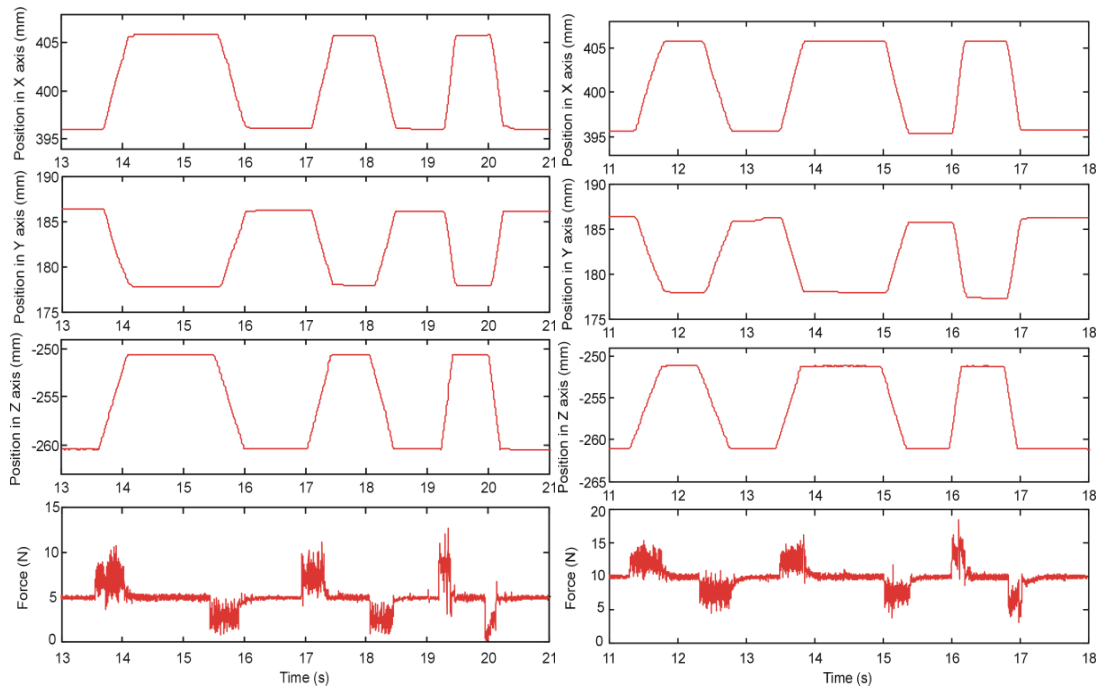


Figure 6.40 ITAE of force response with different speeds. Error bars show the standard deviation.

4. System behaviour with different desired force

Previous tests were carried out using a desired force of 2N. This section examines the system behaviour with different forces. The experimental procedure was similar to test 4, but the desired force was set to 5N and 10N respectively, at speeds of 20mm/s, 25mm/s and 50mm/s. The results are illustrated in Figure 6.41.



(a) $F_d = 5N$

(b) $F_d = 10N$

Figure 6.41 System behaviour with different desired force.

The top three curves of Figure 6.41(a) demonstrate the trajectory of the coil, and the bottom one is the force response of $F_d = 5N$. Compared with the coil's trajectory in Figure 6.31, there is no significant difference found in the changes of positions with these two different force set points. In addition, the average RMS error of the position during the movement is around 2.5mm (Figure 6.42), which is consistent with the error found in test 4 when $F_d = 2N$. The force behaviours are also similar, whereby the force increases when the ball moves towards the coil, and is decreased when the ball moves away. However, with the larger force, the contact can be maintained even at the speed of 50mm/s. This indicates the force response is faster with a larger force set points. Figure 6.41(b) is the corresponding system response of where $F_d = 10N$, and the force response is more stable compared with the smaller force, where the contact between the ball and coil is never lost with this force.

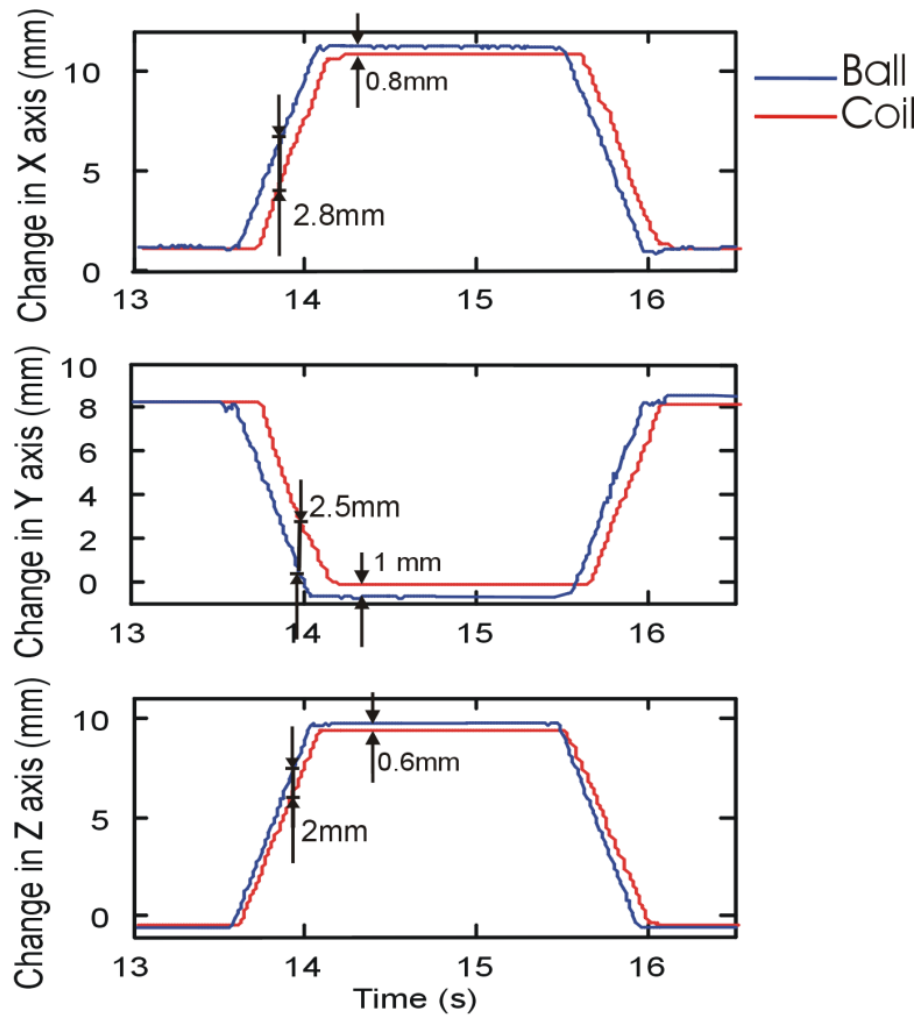


Figure 6.42 Position errors with 5N contact force. The error display on the picture shows the mean error during the movement and in steady state respectively.

The ITAE values of these three contact forces are illustrated in Figure 6.43, where the ITAE decreases with the increment of the force. This indicates that the force error of larger force set point is better than the smaller set point during the movement and the system is more stable and faster when a larger force set point is used for control.

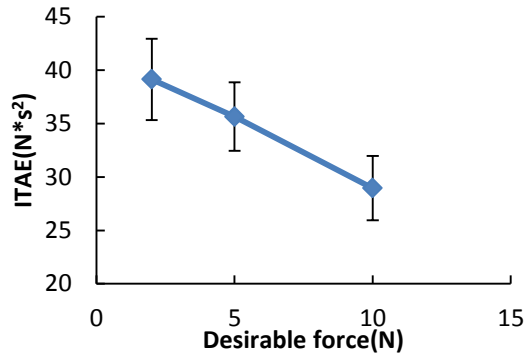


Figure 6.43 ITAE value of different desired forces. Error bars show the standard deviation.

5. System behaviour with different update rates

The ITAE of the force responses with different update rates were calculated using $K_c = 0.3, K_{fl} = 0.0015$. The ITAE value of three update rates of 4ms, 8ms and 16ms are illustrated in Figure 6.44. It can be seen that lowering the update rate of the control system leads to reduced system performance. The ITAE is $40 N \cdot s^2$ with the fast rate of 4ms, and this value is significantly increased to more than $90 N \cdot s^2$ when the update rate is set to 16ms.

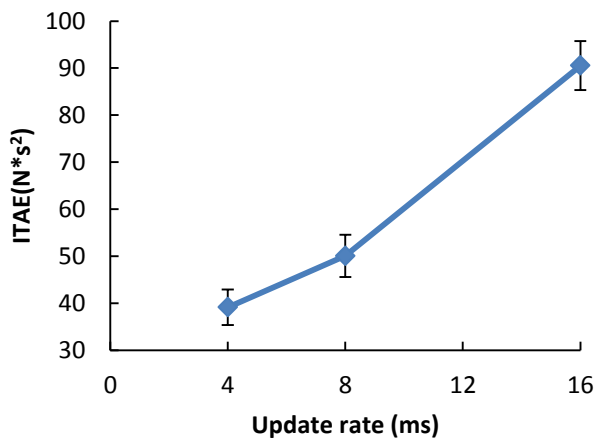


Figure 6.44 ITAE value of different update rate.

6. System behaviour of tracking natural movements

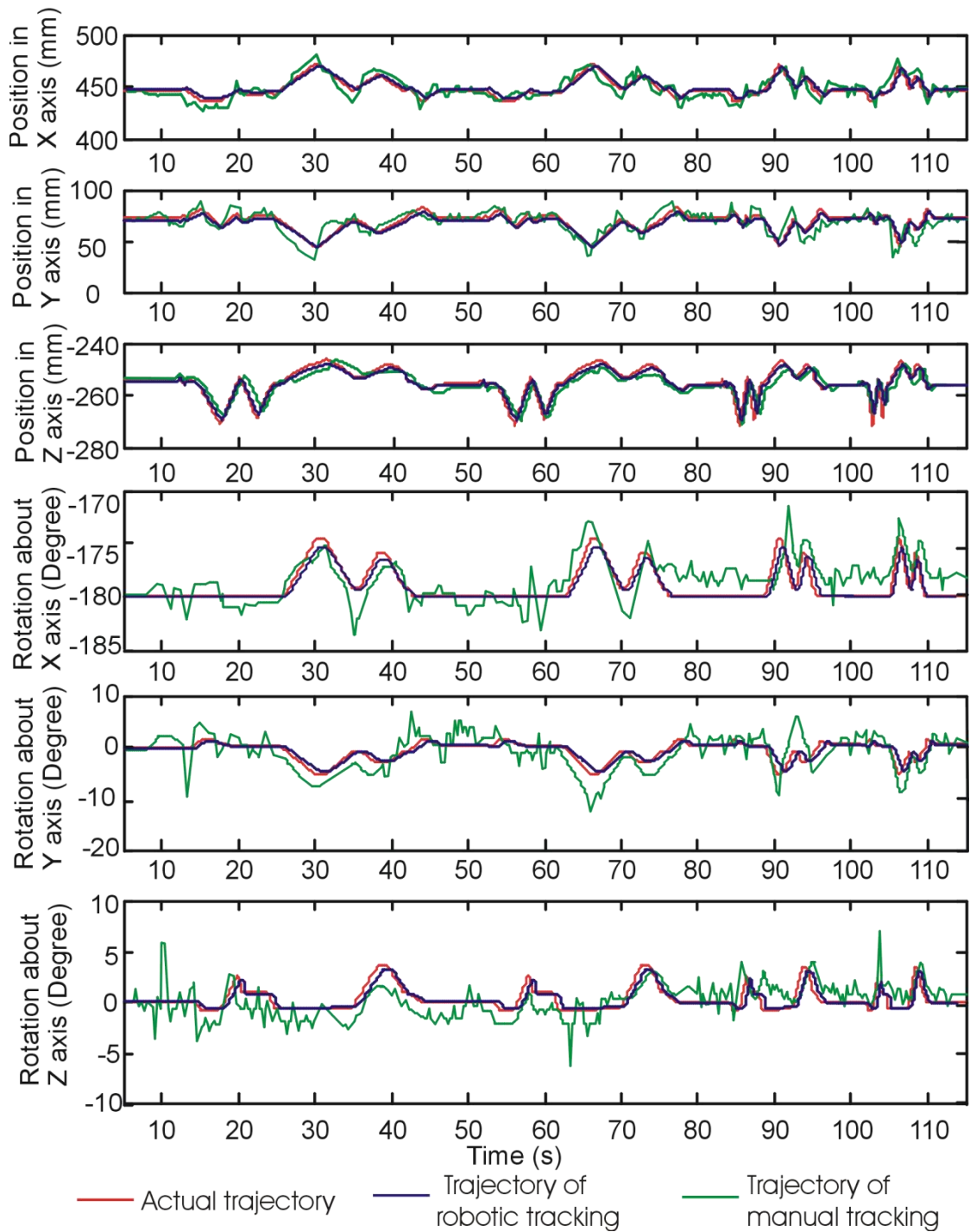


Figure 6.45 System behaviour with natural data.

The disturbance motions used in previous tests are all predefined movements. This section tracks the natural movements of a subject using the combine position and

force controller. Twenty positions and orientations were randomly selected from the subject's movement data captured in Chapter 3, and they were input into the PUMA robot to simulate the natural movement. The PUMA robot moved the ball randomly between these predefined points. System behaviours are illustrated in Figure 6.45; there are both position and orientation changes during the movement, the trajectories of the robotic tracking are a combination of the behaviours addressed in previous tests. The trajectories of the robotic tracking follow the predefined trajectories with a maximum RMS error of 2.6mm in position and 1.3 degree in orientation, which is consistent to previous tests. In addition, the trajectories of manual tracking are also illustrated in the figure, it is obtained by manually following the natural movement. It can be seen that it is difficult for the experimenter to manually maintain and refine the position and orientation when the PUMA robot moves. The error in position and orientation are significantly worse than the robotic TMS system, and different trajectory of the manual tracking is obtained with different experimenter, and it also varies from time to time with the same experimenter.

6.6 Summary

A combined position and force controller is proposed to track movements of moving objects. In this application, an external position sensor and wrist-mounted force sensor are used to control a Staubli manipulator using external real-time path control. The controller is proposed to control all six axes of the manipulator, where the Z axis of the coil frame is force controlled and the remaining is position controlled. Experiments were carried out and the results show the controller has the capability to track the position and orientation disturbance in any axis. In this application, the force response cannot be satisfied using the force PI controller with the constant gains as it introduces either slow force responses or instabilities to the system (Figure 6.17 and Figure 6.18). For this reason, the gain scheduling method is applied in the system for force control, and the force response is significantly improved using the gain scheduling method, which has a maximum force error of 4N during the movement and the ITAE of a period of 5 seconds is around $40 N \cdot s^2$. The performance of the combined controller has been evaluated. The RMS error of the position during the movement is around 2.5mm and reduces to 0.8mm in steady state.

This performance is not only affected by the control gains, but also different control force and speed of the movement. Figure 6.43 shows the controlled force set point of 2N, which makes the force control more challenging in this system. In addition, it is found that the update rate has a significant effect on the system performance (Figure 6.44), and this application is limited by the update rate of the robot controller which is 4ms as maximum. In conclusion, although more work should be done to improve the performance of the combined position and force controller, it can control both the position of the coil and the contact force, which has met the robotic system has shown the significant advantage over the hand-held TMS system. The next chapter will discuss the software design of the system.

CHAPTER 7

SOFTWARE DESIGN

Previous chapters have discussed the major techniques employed in the robotic TMS system. This chapter presents the development of the software requirements which is a key component of the system since the realization and integration of all the techniques described previously depend on appropriate software design and integration. The QNX real-time operating system is employed for the control PC due to the excellence of its microkernel structure and inter-process communication mechanism. The software can be divided into two parts according to the different controller requirements. The major part is developed on the QNX real-time control PC using C++; a modular approach was adopted to match the functions of the system. A user interface was also built up using PhAB application builder to provide the experimenter with a user friendly interface to communicate with the system. The other software component is built on the CS8C robot controller using the VAL3 programming language. A real-time motion control function called ALTER is applied in order to externally control the robot movements. An Ethernet connection that communicates via TCP/IP socket is established between the two controllers for exchanging information.

7.1 QNX real-time operating system

Since several tasks are required to running simultaneously on the PC, an appropriate operating system is essential to ensure the robust and performance of the robotic system. The QNX operating system (full name: QNX Neutrino operating system) implements the standard POSIX API; version 6.4.0 is employed in this application. It provides for multi-tasking, multi-threads, priority-driven pre-emptive scheduling, with fast context-switching that are essential ingredients of a real-time system. More details of QNX system is in Appendix 2.1.

7.2 VAL3 Robot Language

The VAL3 (Staubli) robotic programming language is employed with the CS8C controller, and is a high level robot programming language. The VAL3 language is integrated as a part of the robot control system and it combines the basic features of a standard real-time high-level computer language with functionalities that are specific to the control of an industrial robot. The VAL3 language uses an easy to understand syntax and clear instruction set. The control applications can be written on the Manual Control Panel (MCP) or on a PC in a specific development environment and downloaded to the robot controller. VAL3 can continuously compute the trajectory, which allows complicated motions to be quickly executed and reduces the complexity of the overall system. More details on how to programming with VAL3 are in Appendix A 2.3. The tracking scheme requires the robot's movement to be controlled in real-time so that the robot can immediately track the subject's head movement. The real-time motion control function integrated in the Staubli robot controller is called ALTER. It allows the geometrical transformation (rotation and translation) applied to a robot path to be immediately effective. This feature makes it possible to modify a nominal path with external sensors (position and force sensors are employed in this application). There are six main instructions used in this application:

- *alterBegin()*: initialize the alter mode for the alterable path being executed.

- *alterEnd()*: exit the alter mode and make the current move not alterable any more.
- *alterMovej()*:register an alterable joint move command.
- *alterMovel()*:register an alterable linear move command.
- *alterMovec()*:register an alterable circular move command.
- *Alter()*: specify a new alteration of the alterable path.

The ALTER mode initiated with *alterBegin()* terminates with an *alterEnd()* command. When the end of an alterable path is reached, the ALTER mode remains active until *alterEnd()* is executed. The ‘altered’ coordinates are defined in the frame or tool specified with the *alterBegin()* instruction. When a frame is used as reference, the path modification is with respect to the world frame of the robot. This mode must be used when the deviation of the path is known or measured in the Cartesian space. When a tool coordinate frame is used as reference, the path modification is relative to the tool frame. This mode must be used when the deviation of the path is known or measured relative to the tool centre point (TCP). The *alterEnd()* instruction is needed to specify how to terminate the ALTER mode, either before the nominal move is complete, so that the next non-alterable move can be sequenced without stop; or after, so that it remains possible to move the arm with ALTER while the nominal move is stopped.

The tool is used as the reference frame in this application, and programming consists of following steps:

- Use *alterBegin()* enable the ALTER mode, and select the ALTER mode that relative to the tool frame.
- Define the nominal path which is programmed as standard moves with *alterMovej()*, *alterMovel()* and *alterMovec()* instruction.
- Specify a deviation to the nominal path with *alter()* instruction.
- End the alteration with *alterEnd()* if the alteration is finished.

The ALTER command must be synchronized with the communication period so that the path modification can continuously respond to the subject's movement. This is achieved using synchronous VAL3 task with *taskCreatSync()* instruction which has a maximum update rate of 4ms.

7.3 Design of the software system

There are following major requirements proposed for design of the software system:

- Deterministic operation. Since the most important property of the software system is to ensure the safety of the subject, it is necessary to design the software with predictable behaviour during the running time.
- Tolerance to system faults. Unpredictable system faults occurring during the stimulus operation may be of critical consequence to the subject's safety, for example: the communication between the host computer and the tracking system could disconnect; or the power of the tracking system could be lost during operation. The software system should respond immediately to these faults.
- Module-style-design. The overall software system should be divided into modules with each module handling a specific task. This method can help reduce a complex problem into simpler ones, and it because easier to identify the origin when faults occur.
- Optimal time plan of data acquisition. The robotic TMS application requires the latest data of the position and orientation of the coil and subject' head. The update rate should be as quick as possible, however, the rate is constrained by the performance of the hardware.

7.3.1 Overview of the software system

The architecture of the software system is illustrated in Figure 7.1. The position and force data are acquired and processed by the QNX control PC, and the ‘ALTER’ trajectory is transmitted to the robot controller to guide the robot movement.

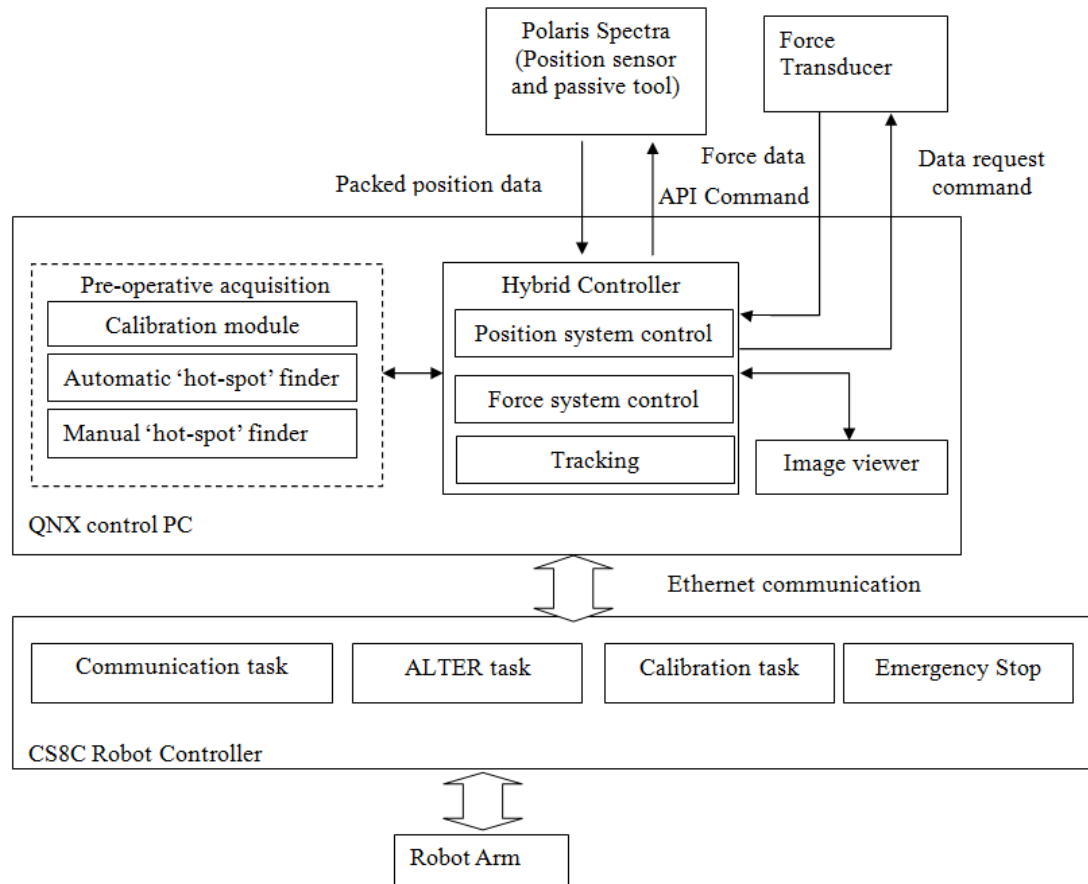


Figure 7.1 Software architecture.

There are three main modules integrated in QNX control PC: Pre-acquisition module, hybrid controller module, and image viewer. The pre-acquisition module is developed for the pre-operative mode of the system, it consists of three functions: calibration, manual ‘hot-spot’ finder and automatic ‘hot-spot’ finder. The hybrid controller module is the fundamental section in the system, and comprises three units: position sensor system control, force sensor system control and tracking units. The position and force sensor system control units provide the parsed position and force data to other modules. The tracking unit incorporates the tracking control algorithms proposed in Chapter 6, and generate the trajectory of the robot arm according to the

hybrid position and force controller. The image viewer module is used for displaying the MRI images of the subject. In addition, it can also display the current stimulation position after the registration of the MRI image and the real head of the subject.

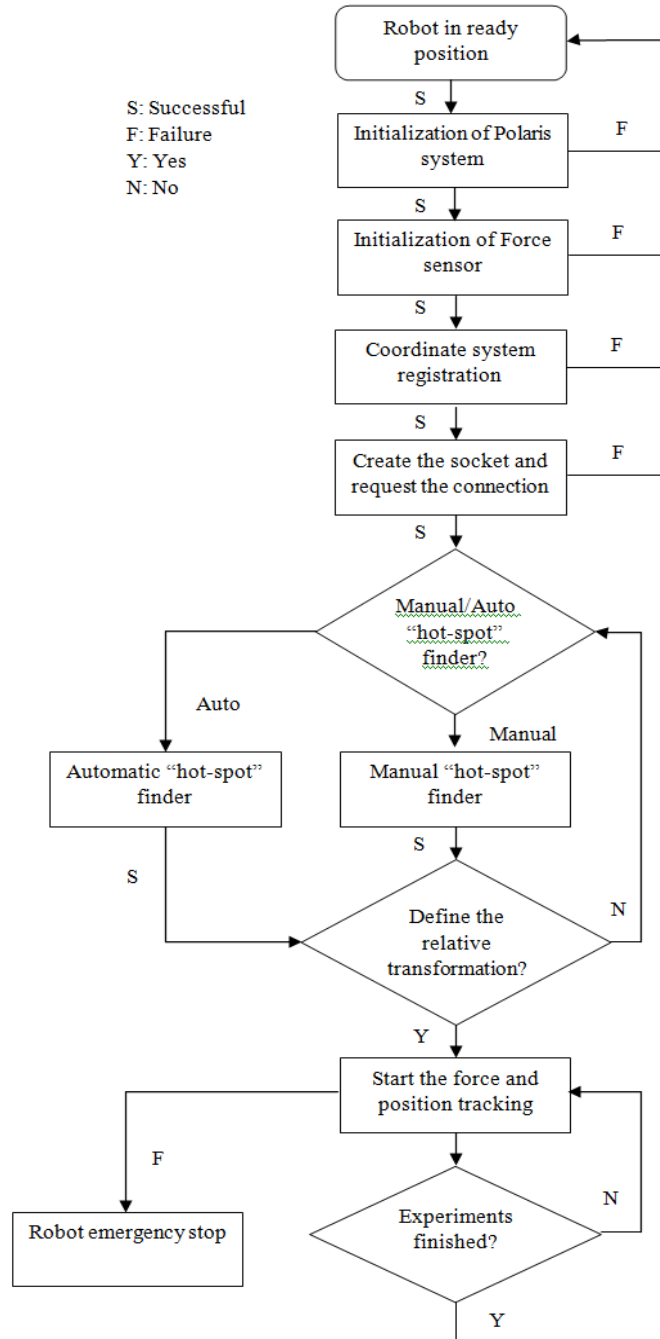


Figure 7.2 System operating sequence.

The operating sequence of the system is illustrated in Figure 7.2. The robot starts with a ready state, followed by initialization of the position tracking system and force sensor system. The initialization includes establishing communication, setting baudrate, sampling rate, channel numbers, passive tools. If the initializations are successful, the coordinate system registration is carried out. Several consecutive movements are made by the robot arm to establish the registration and transformations are calculated once these movements have finished. After registration, the ‘hot-spot’ (stimulation site) is established, using either the manual or automatic mode. In the manual mode, the system enables “tele-operation” so that the experimenter can grasp and position the coil manually to find the ‘hot-spot’. The latter mode is selected, once the robot has been initially positioned by the experimenter, and initiates the automatic finder routine. Once the ‘hot-spot’ is defined, the relative transformation between the coil and ‘hot-spot’ should be recorded, and the tracking mode runs until this experiment finished. If any component failure is detected during the tracking, the robot will be stopped immediately.

The CS8C robot controller receives information from the PC to guide the robot movement or modify its position. It has four major tasks to accomplish, namely *communication*, *calibration*, *ALTER* and *emergency stop* tasks. The *communication* task sets up the Ethernet communication and talks with the PC via socket. All information sent from the PC to CS8C is received and parsed by this task. The *calibration* task is used for collecting sets of robot positions and orientations to format the equation and finally obtain the transformation between the position sensor and the robot base frame. The *ALTER* task is used to control the robot motion in real-time. Finally, the *emergency stop* task detects any system faults and stops the robot immediately. The *communication* and *ALTER* task are created as synchronous tasks since they have to be scheduled at regular periods of 4ms for data acquisition and motion control. The other two tasks are asynchronous tasks, the *emergency stop* task are set at the highest priority to ensure the safety. The flow chart for the operating sequence of these four tasks is illustrated in Figure 7.3. To begin with, the robot arm is commanded to a ready position. The *calibration* task is then called if calibration has not been carried out. Once the calibration is complete, the socket link

is initialized; and the *communication* task will keep listening at the port until there is a connection request. After the connection is successfully established, the port can received the data sent from the PC and the ALTER task can initiate the robot movement. If any failure takes place during this process, the *emergency stop* task is called, and system stops immediately. If there is no failure, the system keeps running until the experiment finished. Finally, the robot returns to the ready position.

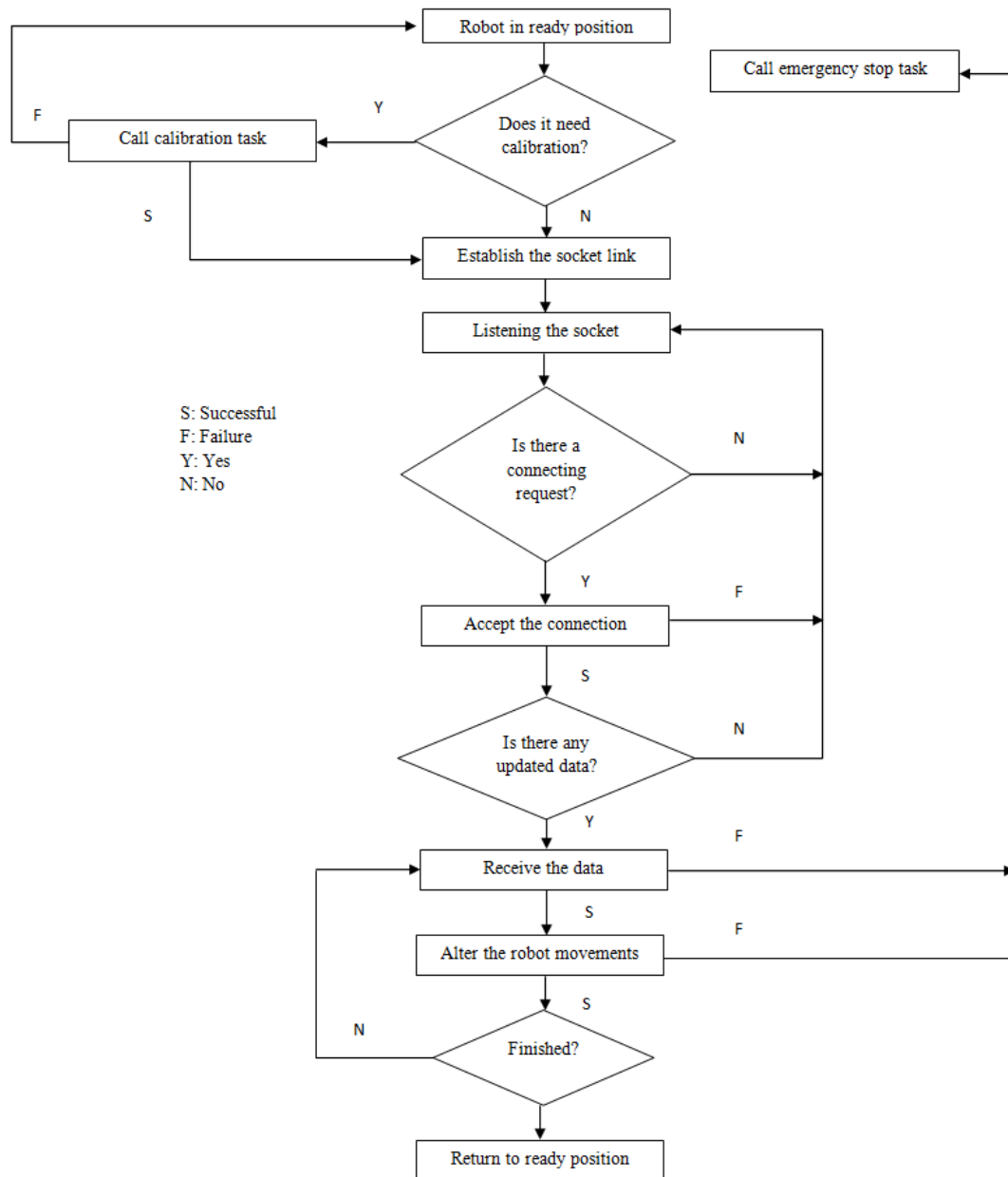


Figure 7.3 Flow chart of the application running on the CS8C.

7.3.2 Serial communications in QNX operating system

Serial communications in QNX is required for the Polaris system which is controlled through a RS232 interface. QNX considers the serial ports as *device files*, thus a serial port can be accessed by simply opening the corresponding *device file*. Each serial port on a QNX system has one device file (files in the /dev directory) associated with it; the device file of the serial port used in this application is /dev/ser0. Since a serial port is a file, the *open()* function is used to access it, such as:

```
fd = open("/dev/ser0", O_RDWR | O_NOCTTY);
```

The open option *O_RDWR* indicates that the port is open for both reading and writing. The *O_NOCTTY* flag tells QNX that this program does not want to be the "controlling terminal" for that port. If this is not specified, any input (such as keyboard abort signals) would affect the process. *write()* and *read()* function are used to transmit and receive data respectively:

```
n = write(fd, "ABC\r", 4);
size_read = read( fd, buffer, sizeof( buffer ) );
```

The *write()* function returns the number of bytes sent or -1 if an error occurs. When the port operates in raw data mode, each *read()* call will return the number of characters that are actually available in the serial input buffers. If no characters are available, the call will block (wait) until characters come in, which means that reading data from the serial port will freeze the execution of the program. In order to resolve this problem, the port should be always set to *non-block* mode before the data acquisition procedure, carried out by setting the *O_NONBLOCK* option when opening the serial port:

```
fd = open("/dev/ser0", O_RDWR | O_NOCTTY | O_NONBLOCK);
```

The *close()* function is used to close the serial port:

```
close(fd);
```

tcgetattr() and *tcsetattr()* are the two major functions to get and set terminal attributes, such as baud rate, number of data bits, parity, stop bits etc. A *termios* structure is used to contain all of the serial options in QNX. An example is listed as follows to configure the baud rate of the serial port to 9600, eight (8) data bits, no (N) parity bit, and one (1) stop bit.

```

/*****

struct termios tio; //define termio structure

speed_t speed; //define serial speed

//8,n,1
tio.c_cflag &=~CSIZE;
tio.c_cflag &=~CSTOPB;
tio.c_cflag &=~PARENB;
tio.c_cflag &=~ IHFLOW;
tio.c_cflag &=~OHFLOW;
tio.c_cflag |=CS8|CLOCAL|CREAD;
//8,n,1
speed=B9600;
rc = cfsetispeed(&tio, speed)//Set the input baud rate in a termios
structure
rc = cfsetospeed(&tio, speed);//Set the output baud rate in a
termios structure

rc = tcsetattr(fd,TCSANOW,&tio);

*****/

```

Since an API command: “TX” is required to be sent to the Polaris system continuously to acquire the latest transformations of the passive tools, a “FD handler” is setup using *PtAppAddFd()* in order to efficiently coordinate the transmit and receive data via the serial port. An “FD handler” is a function that is called by the main event loop when an I/O port is ready for input or output. This means whenever data is waiting in the receive buffer of the serial port, a function will be called automatically, the data then can be read and parsed. On the other hand, when the data read is completed and the port is ready to transmit, a TX function is called automatically. In this way, transmit and receive data can be executed at an optimal speed without consuming more system resource.

7.3.3 Ethernet communication using socket

The communication between the QNX PC and CS8C robot controller is established via Ethernet link using a socket, which provides an endpoint for bidirectional inter-process communication. There are two widely used socket types, namely, stream sockets and datagram sockets. Stream sockets treat communications as a continuous stream of characters, while datagram sockets have to read entire messages. Each uses its own communications protocol. Stream sockets use TCP (Transmission Control Protocol), and datagram sockets use UDP (Unix Datagram Protocol). However, the CS8C robot controller only supports TCP, and uses the client server model. As illustrated in Figure 7.4, these terms refer to the two inter-communication processes in which the client, connects to the other process, the server, typically to make a request for information. The server in turn typically offers its services to the client, which needs to know the existence and the address of the server, however the server does not need to know the address of the client prior to the connection being established. Once a connection is established, both sides are able to send and receive information to each other.

The steps involved in establishing a socket on the server side are as follows (Figure 7.4):

- Create a socket with the *socket()* system call;
- Bind the socket to an address using the *bind()* system call. For a server socket on the internet, an address consists of a port number on the host machine;
- Listen for connections with the *listen()* system call;
- Accept a connection with the *accept()* system call. This call typically blocks until a client connects with the server;
- Send and receive data;
- Socket closed once the communication thread is quit.

The steps involved in establishing a socket on the client side are as follows (Figure 7.4):

- Create a socket with the *socket()* system call;
- Connect the socket to the address of the server using the *connect()* system call;
- Send and receive data.
- Socket closed once the communication thread is quit.

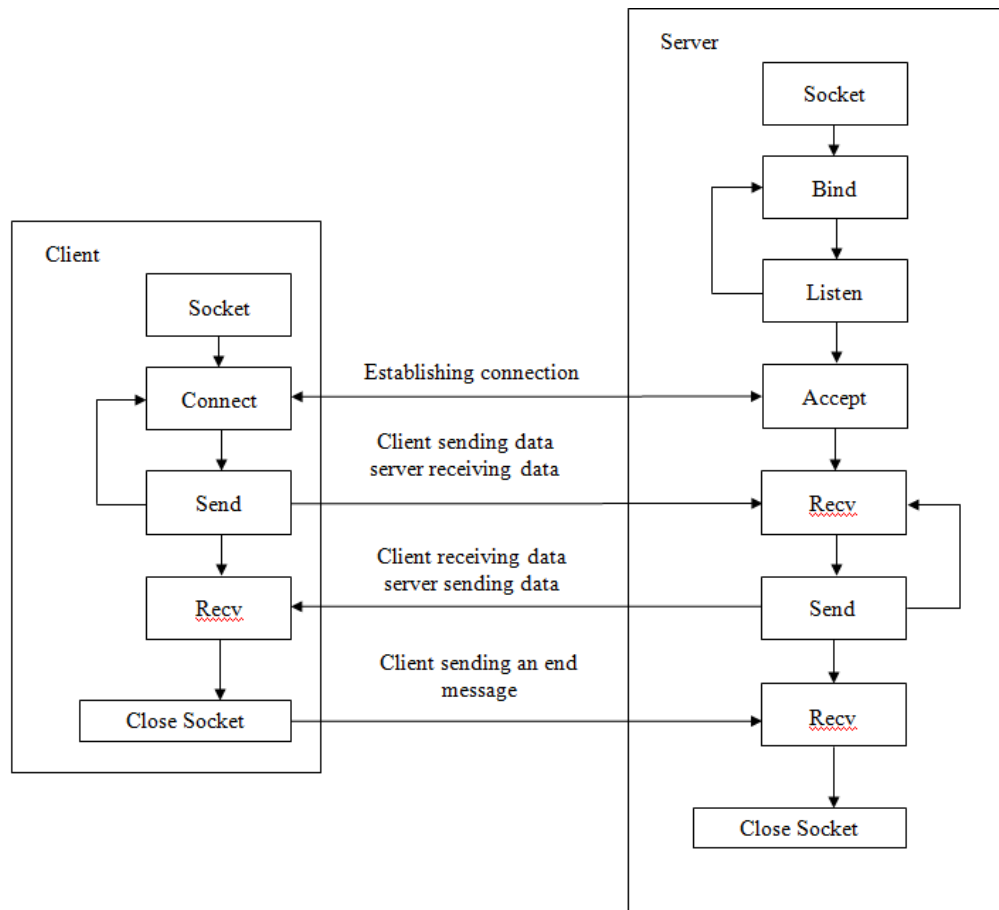


Figure 7.4 TCP socket flow diagram.

Here the QNX PC is used as the client (which makes requests), and the CS8C robot controller is set as the server (which responds to the requests). On the CS8C robot controller, the socket can be configured as client or server mode directly via the control panel. The server socket is activated in the CS8C each time a VAL3 program uses it, and is deactivated when the instruction *clearBuffer()* is called. In VAL3, The

sio type is used to link a VAL3 variable an Ethernet socket connection. There are two instruction types used to control the socket:

sioGet ():Reads a single character or an array of characters from the socket and returns the number of characters read. The reading sequence stops when the data array is full or when the input reading buffer is empty. For an Ethernet socket connection, *sioGet* first attempts to make a connection, if a timeout for input communication has been reached, *sioGet* returns -1. If the connection is active, but there are no data in the input reader buffer, *sioGet* waits until data is received or until the end of the waiting period has been reached.

sioSet():Writes a character or an array of characters to the socket and returns the number of characters written. For an Ethernet socket connection, *sioSet* first tries to make a connection if there is no active connection.

7.3.4 Calibration module

The calibration module is designed as part of the software system so that calibration can be easily carried out if the position of Polaris Spectra tracking sensor has changed prior to starting the TMS stimulation session. The software module is designed in two parts (Figure 7.5): the first part collects the robot positions, and the other solves the transformation equation ${}^iHX = {}^jHYB$.

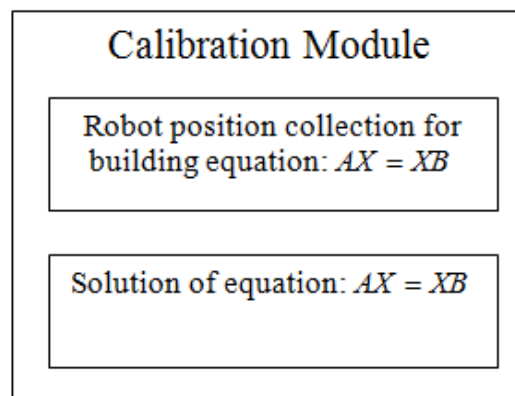


Figure 7.5 Calibration module: robot position collection and solution.

The operator interface of this module consists of five buttons (Figure 7.6). The top button opens the communication port between the robot controller and QNX PC. The next three buttons are used to successively command the robot to do consecutive movements, and to record the positions and orientations of the robot with respect to its base frame and corresponding positions and orientations of the passive tool, with respect to the tracking system frame. Seven positions are recorded by the system, and are used for building Eq. (5.19). The ‘cali’ button is used to build and generate the solution to Eq.(5.23) and Eq.(5.24).

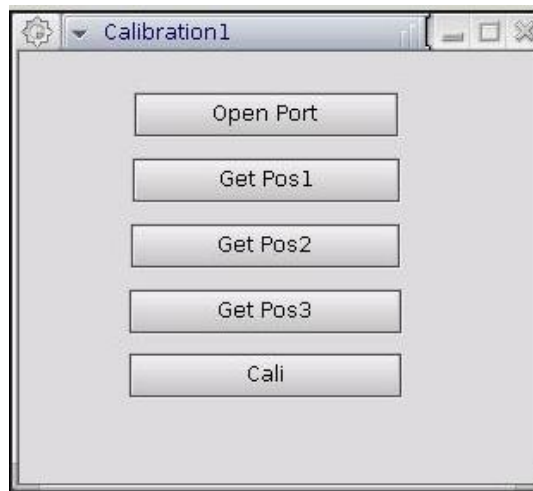


Figure 7.6 Control window of calibration module.

Singular Value Decomposition (SVD) method is used to analyse the solution of Eq. (5.23) since it is more accurate than the least squares method for linear optimization. For a given matrix $M \in \mathfrak{R}^{m \times n}$ decomposition is given by:

$$M = U \Sigma V^T \quad (7.1)$$

where

- U is a $m \times n$ matrix with orthogonal columns; the column vectors of U are the left singular vectors of M ;
- Σ is a $n \times n$ diagonal matrix with non-negative values; the diagonal values of Σ are the singular values of M ;

$$y_i = z_i / \xi_i \quad (7.5)$$

and the final result is given by:

$$x = Vy \quad (7.6)$$

here Eq.(5.23) is a linear equation of the form $Ax = 0$. Thus the minimization of $\|Ax\|$ needs to be carried out, and a constraint has to be imposed on x to ensure $x \neq 0$. Hence, the problem reduces to that of finding x which minimizes $\|Ax\|$ subject to $\|x\| = 1$.

Hence :

$$\|Ax\| = \|U \Sigma V^T x\| = \|\Sigma V^T x\| \quad (7.7)$$

substituting $y = V^T x$ to Eq. (7.7), the problem becomes the minimization of $\|\Sigma y\|$ subject to $\|y\| = 1$; where Σ is a diagonal matrix with elements in descending order, thus the minimal solution is $y = (0, 0, \dots, 0, 1)^T$, and x is equal to the last column of V using Eq. (7.6).

7.3.5 Hybrid controller module

This module comprises three submodules: force sensor control, position sensor control and tracking submodules. The force and position sensor system control submodules are used to obtain the force and position data from the sensors; and parse the data for further use. The tracking submodule is designed to utilise the data for tracking control using the combined force-position control method previously described in Chapter 6. These are controlled by three different threads, the purpose of which is to ensure the independency of each submodule functions. By using multi-threads, the sensors can continuously update the data, and the processed data can be transferred to the robot controller for further use. In this way, the data transferred to the robot is the most up to date, which ensures both the performance

and safety of the whole system. The algorithms for the tracking submodule are described in Chapter 6.

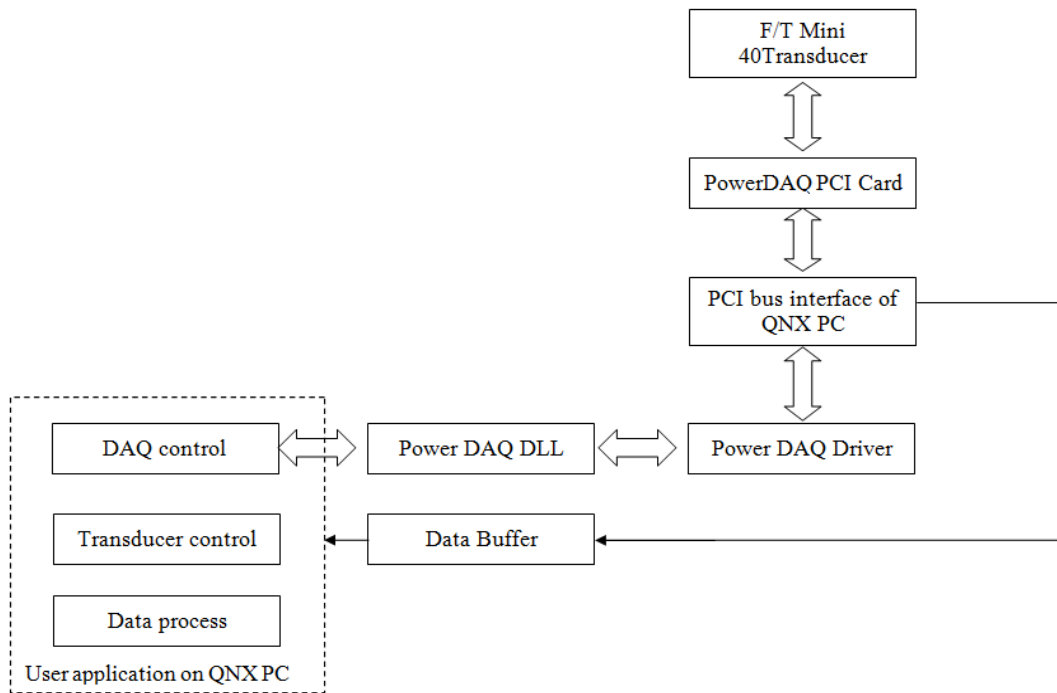


Figure 7.7 Structure of force sensing control application.

- **Force sensor control submodule**

The structure of the force sensing control submodule is illustrated in Figure 7.7, and comprises three parts: DAQ device control, transducer control and data process. The DAQ control application communicates with the DAQ PCI board via the PowerDAQ API (UEI, Inc, US), which is integrated into the PowerDAQ dynamic-link library (DLL). The PowerDAQ API includes a set of functions that allow user applications to get board-specific information, such as model, serial number and interrupt request (IRQ) line. These API can manipulate the PCI card through the DAQ driver. Data is transferred from the board through the PCI bus and stored in the user-level buffer.

The following steps are used to obtain a set of analog data from the force transducer through the DAQ card and convert them into force and torque values.

1. DAQ card initialization, using the following steps:

- *pd_find_devices()* is used to establish communication with the card, the number of channels, scan rate, input range, input mode can be configured.
 - *_PdAInReset()* is used to reset the board.
 - *_PdAInSetCfg()* is used to set up the configuration.
 - *_PdAInSetChList()* is used to set up the Channel List.
 - *_PdAInEnableConv()* is used to enable conversions.
 - *_PdAInSwStartTrig()* is used to issue the software-based Start trigger.
2. Transducer configuration, including: setting the calibration file, force and torque units, and tool transformation. The following ATI 'C' library functions are used to achieve the configuration;
 - *createCalibration* - Loads calibration information for a transducer from a calibration file.
 - *SetToolTransform* - Performs a 6-axis translation/rotation on the coordinate system of the transducer.
 - *SetForceUnits* - Sets the units of force output.
 - *SetTorqueUnits* - Sets the units of torque output.
 3. Configured the board to collect analog samples as required using the on board timer or a program loop using *_PdAInGetSamples()*.
 4. *Bias()* function are used to store a baseline reading (zero the transducer).
 5. The *ConvertToFT()* function is used to convert another array of voltages into forces and torques, and bias reading will be subtracted automatically.

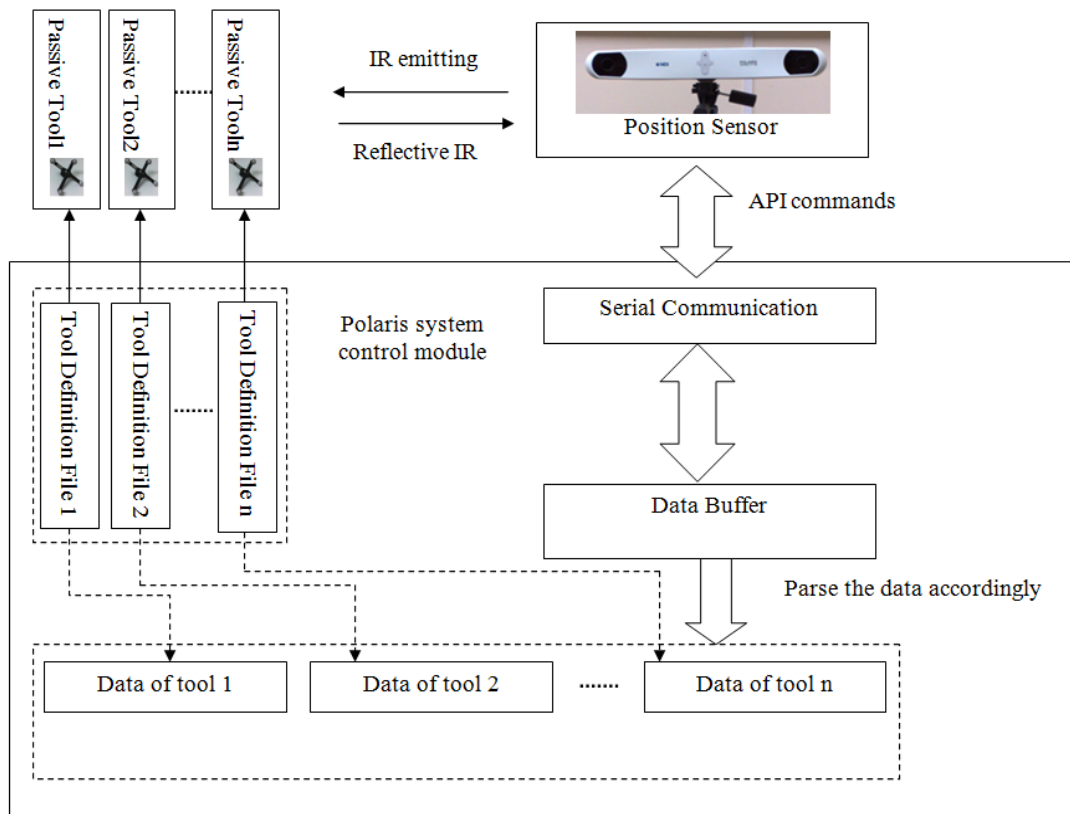


Figure 7.8 Software structure of position sensor system.

- **Position sensor system control module**

The structure of Polaris system control module is illustrated in Figure 7.8. The position sensor is located in the working cell, and the passive tools are mounted on the subjects or the robot arm. API commands are available, and a serial communication function is designed to transmit API commands and receive data from the position sensor. Tool definition files are loaded in the position sensor before acquiring the data from it and the sensor returns the different tool identifiers according to these files. After the positional and orientation raw data for these passive tools is acquired and buffered, they need to be further parsed into position and orientation data according to the number of tools. These separated data packets are then saved in the different buffers that are ready for use.

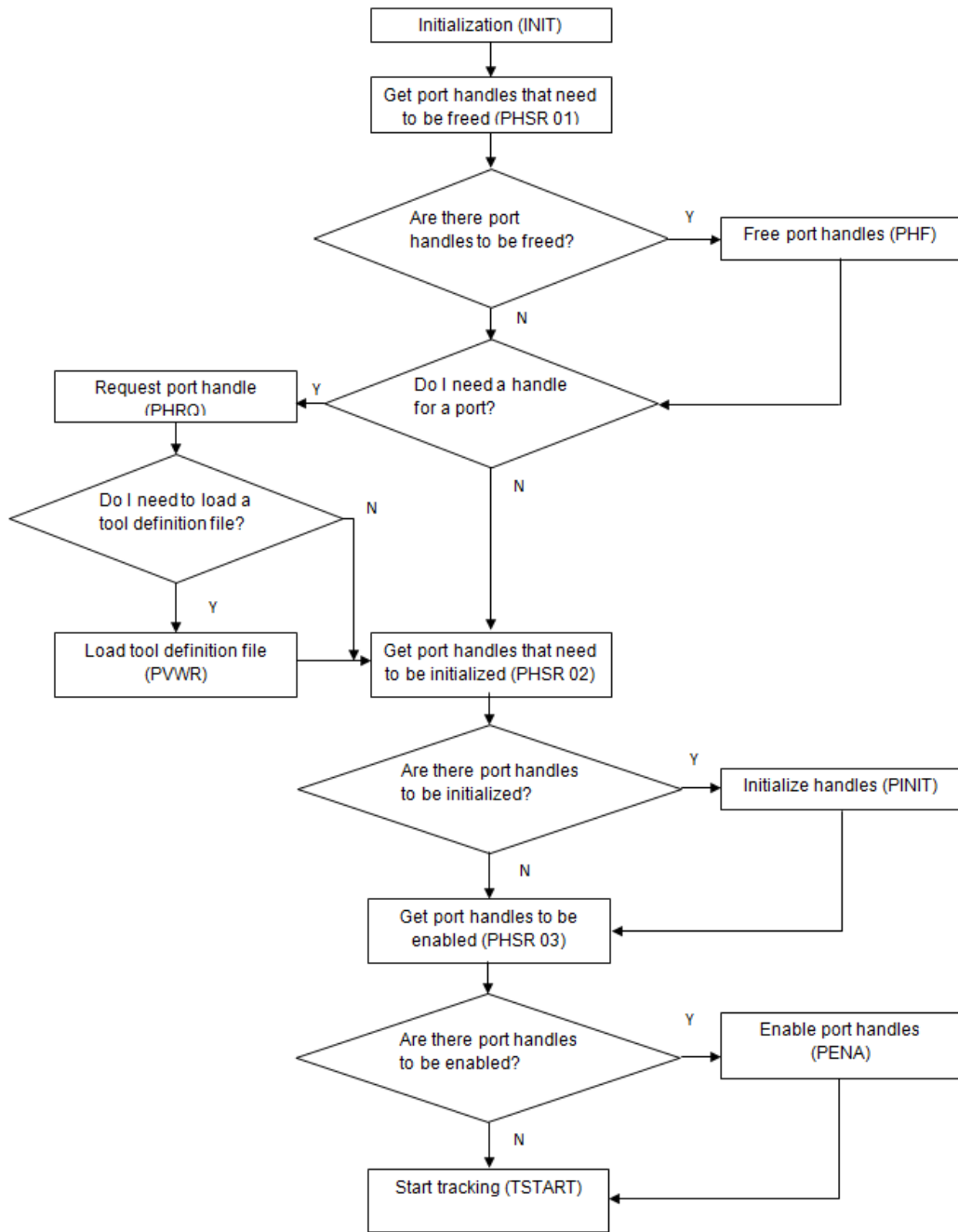


Figure 7.9 Flow chart for configuring the Polaris system.

The flowchart (Figure 7.9) illustrates the configuration process of the Polaris system. The system is initialized using the instruction *INIT* to ensure that it has the correct configuration, including firmware revisions and the characterized measurement volumes for which the position sensor has been calibrated. Once the system is successfully initialized, it reports the port handles that need to be freed using *PHSR 01*. A port handle is assigned to a tool definition file; the position sensor identifies

the tool definition file through the corresponding handle. If there are port handles that have been previous used, *PHF* is used, otherwise, *PHRQ* is directly used to request the port handles. The command *PVWR* is then used to assign the tool definition files to the port handles. Once the tool definition files are loaded successfully, the system is required to report the port handles that need to be initialized and enabled using *PHSR 02* and *PHSR 03* respectively. Commands *PINIT* and *PENA* are used to initialized and enable the reported port handles in order to track the tool. Finally, the command *TSTART* is used to start the tracking.

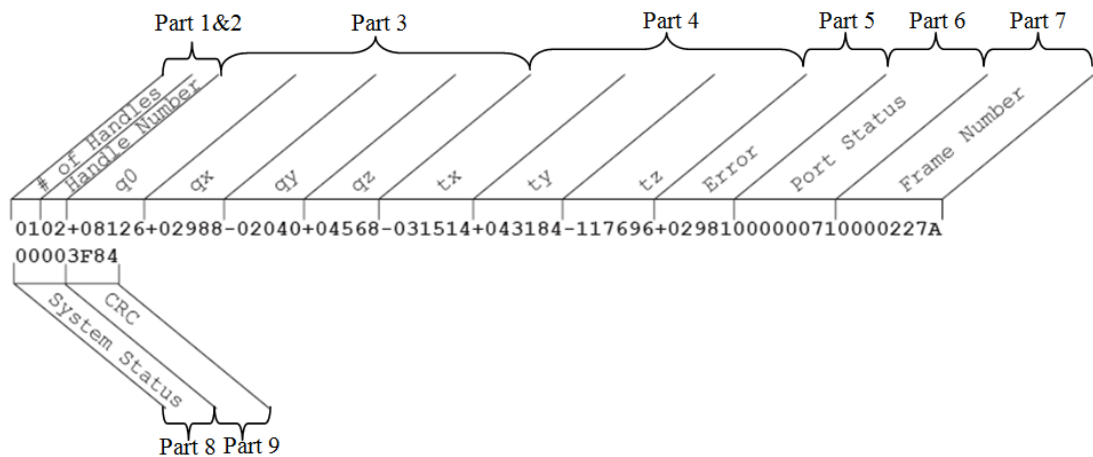


Figure 7.10 An example of the data packet returned by the position sensor.

The information returned by the position sensor is raw data, and an appropriate function is designed to parse and process the information according to the unique structure of the data. An example of the returned data is illustrated in Figure 7.10. All are hexadecimal characters, and the packet is divided into 9 parts:

1. Number of handles - 2 hexadecimal characters that indicate the number of port handles for which transformations are returned. In the example, the number of handle is 01, this indicates only one tool transformation is returned.
2. Handle Number which indicates the port handle whose transformation follows ('02' in the example).
3. Tool rotational component, given in quaternion format of $q = (q_0, q_x, q_y, q_z)$, each 6 characters long including a sign, and an implied decimal in the

position X.XXXX. The rotational part is $q = (+0.8126, +0.2988, -0.2040, +0.4568)$ in the example.

4. X,Y and Z translational component of the tool (unit: mm) with respect to the global coordinate system, a signed 6 decimal digits with an implied decimal point in the position XXXX.XX. In the above example, the translation of the tool is $T = (-315.14, +431.84, -1176.96)$.
5. An error value for the tool transformation. This RMS value, given in mm, is the result of the least squares minimization between the marker geometry in the tool definition file and data from the tool's markers measured by the system. It includes a sign, and 5 decimal digits with an implied decimal in the position X.XXXX, in the example is +0.2981.
6. Status of the tool, indicating whether the tool is out of volume, partially out of volume, or missing. It also indicates whether the port handle corresponding to each tool is enabled and initialized. It comprises 8 hexadecimal characters (32 bits), and each bit represents a status. In this example the port status (00000071) indicates that the tool is occupied, initialized, enabled, and out of volume. More detail of the bit field is given in Appendix 2.4.
7. Frame number for each tool transformation. The frame number is incremented by 1 at a rate of 60 Hz. The frame counter starts as soon as the system is powered on.
8. System status, which indicates the system errors. It contains 4 hexadecimal characters (16 bits), no system error (0000) is reported in the example. Detail of the bit field is in Appendix 2.4.
9. A 16bit Cyclic Redundancy Check (CRC) value calculated by the system. It is used to check and ensure the returned value is correct.

Since each part of the raw data has a fixed length, the information can be parsed using a pointer, which points to the start address of the buffer used to store a packet.

Each part then can be parsed using the start address of its buffer. A *nGetTransforms()* function was developed as shown in Appendix 2.5.

- **State machine of the tracking control module**

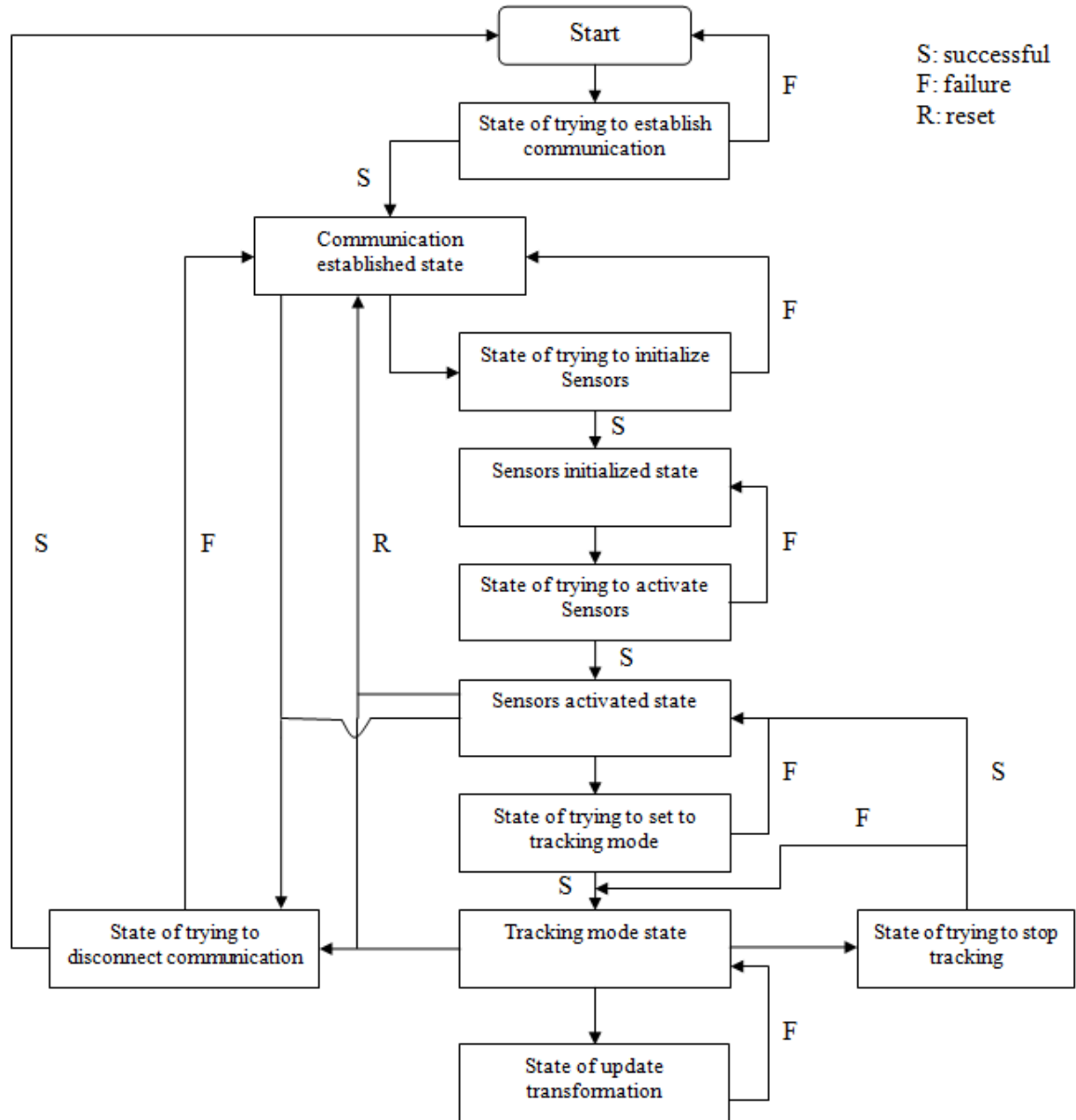


Figure 7.11 State machine of the tracking module.

A state machine is designed to ensure the system tracking control is always in a predictable configuration and to make the software deterministic. Several key states of the tracking control module are defined in the state machine as shown in Figure 7.11:

- System is *ready* to start. This state is established after the robot arm is in the *ready* state and there is no problem with the hardware of the system.
- *Communication established* state – established if the host computer successfully connects to the sensors with correct configuration of corresponding communication types.
- *Sensors initialized* state – Entered if commands for initializing both force and position sensors have been successfully completed. This is to ensure the configurations of both sensors are determined successfully.
- *Sensors activated* state – Sensors must be activated before they can capture data. Therefore, this state is entered if the initializations are accomplished successfully.
- *Tracking mode* state – The system enters this state if both sensors are activated and commands to begin tracking is successfully executed.
- *Update transformation* – This state updates the transformation and entered once the tracking mode is initiated and sensors have received the corresponding scan data.

7.3.6 Software Design of Image module

The image module displays the preoperative scans obtained from MRI or CT scans, and is used to assist the experimenter to obtain a view of the subjects' brain. After appropriate coordinate system registration between the subjects' head and the image frame, the output can be extended for planning and guiding the placement of the coil, and must meet the following requirements:

- Support popular medical image formats.
- Display both the scanned image and stimulation position. In order to give a straight forward visualization to the experimenter, this application not only displays the images but also displays the current stimulation position on the image.

- Coordinate registration. In order to find the current stimulation position on the image corresponding to the real head, a registration function needs to be integrated into the module.
- Communication interface. This provides for communication with other modules in the system.

In this application the DICOM and Analyze image formats, which are the most common formats, are used to store the scans. The Digital Imaging and Communications in Medicine (DICOM) format, produced by the National Electrical Manufacturers Association (NEMA), is the most common format used for storing medical scans, and support the viewing and distribution of medical scan images, such as MRI, CT scans. A single DICOM file contains both a header and the image data (Figure 7.12). The header is used to store information about the patient's name, image dimensions, and pixel spacing, etc.

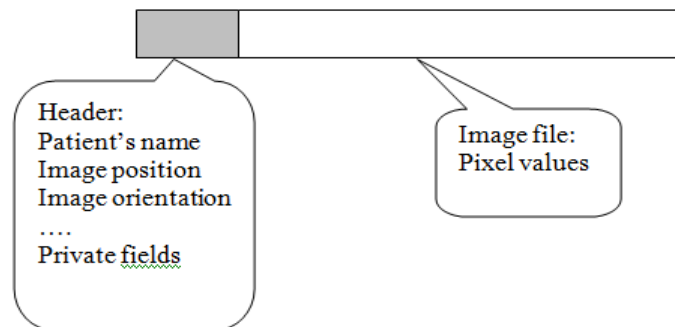


Figure 7.12 DICOM file structure.

The Analyze format is employed to normalize the scans for multi-dimensional display. Unlike the DICOM format, one data item of Analyze format comprises two files: the actual data in a binary format, with the filename extension .img, and another file (header with filename extension .hdr) with information about the data such as voxel size and number of voxel in each dimension. The format of the image file contains uncompressed pixel data for the images in one of several possible pixel formats:

- 1 bit packed binary (slices begin on byte boundaries).

- 8 bit (unsigned char) gray scale.
- 16 bit signed short.
- 32 bit signed integers or float.
- 24 bit RGB, 8 bits per channel.

The header file describes the history of the pixel data and dimensions, including three substructures:

- *Header_key* describing the header.
- *Image_dimension* for describing image sizes.
- *Data_history* which is optional.

These are represented as a structure (in ‘C’) and shown in Appendix 2.6. Based on each format, a programme can be written to decode and view the image in QNX.

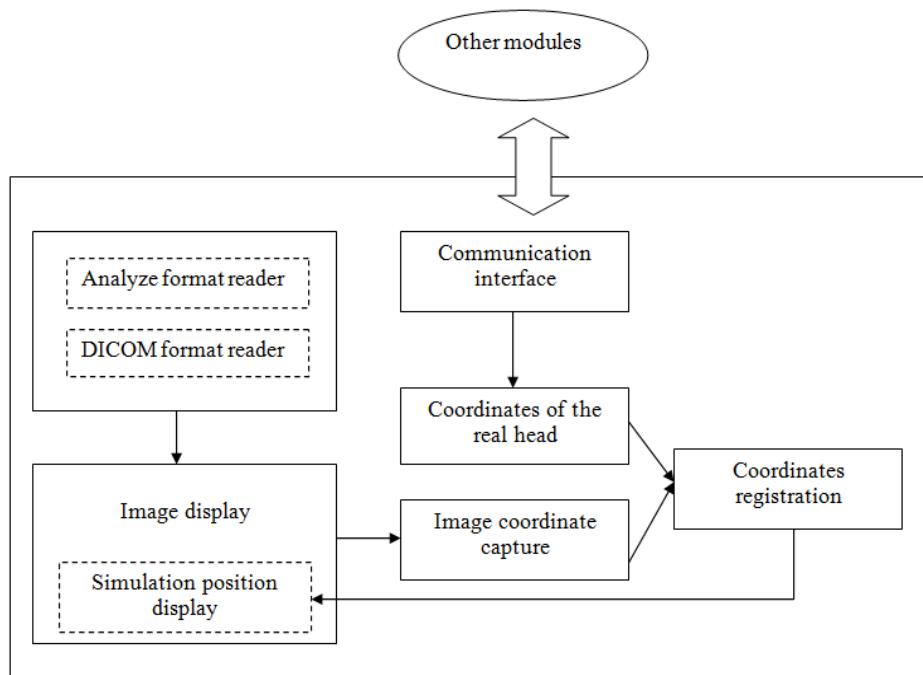


Figure 7.13 Image module structure.

The image module is designed according to the above requirements with the structure as shown in Figure 7.13. Since both Analyze and DICOM format readers are available, the first task is to establish which format the file supports, after which

the images are displayed on the monitor using the corresponding reader. If the user needs to display the current stimulation position, the coordinate registration should be carried out in advance. This can be achieved by registering the coordinates of the image and the corresponding coordinates on the subject's head, using the position sensor and integral pointer. The image coordinates are captured using a mouse by overlaying the positions on the displayed image. After registration, the stimulation position, i.e. the current coil centre position, is displayed as a yellow point on the image as shown in Figure 7.14.

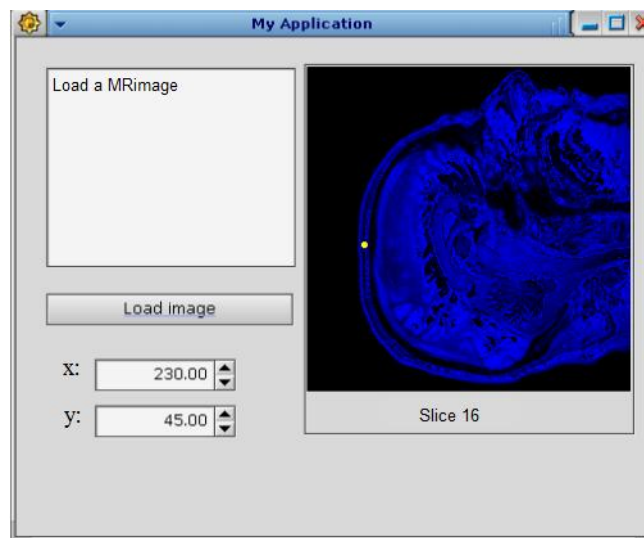


Figure 7.14 MRI image viewer. The picture in the viewer is a 2D image, and the yellow point indicates the current stimulation position of the head. The coordinates of the point (the origin is the left upper corner of the image) are (230,45,16). The Z-coordinate depends on the slice number of the image that is loaded.

The ‘Procrustes analysis’ method is applied to the image registration. This is a statistical shape analysis method that used to determine linear transformation (translation, orthogonal rotation, reflection and scaling) of the points in a matrix to conform them to the points in a target matrix. The criterion is to minimise the sum of squared errors of two sets of points. Suppose there are two objects made up from a finite number of n points in p dimensions, are called landmark points. Two representative matrices are $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$, where x_i and y_i ($i = 1 \dots n$) are vectors in p dimensional space. If $X = (x_1, x_2, \dots, x_n)$ is the target matrix, the orthogonal rotational matrix A and a translational vector b are used to

match the points in $Y = (y_1, y_2, \dots, y_n)$ to $X = (x_1, x_2, \dots, x_n)$ and can be obtained by minimizing Eq.(7.8).

$$D^2 = \sum_{i=1}^n \|x_i - Ay_i - b\|^2 = \sum_{i=1}^n \sum_{j=1}^p (x_{ij} - (Ay_i + b)_j)^2 \quad (7.8)$$

The rotational and translational part can be found separately. The translation part is obtained by centring the two matrices by calculating the mean values of the columns in the matrices. Denoting $z_i = Ay_i + b$, Eq.(7.8) can be written as:

$$D^2 = \sum_{i=1}^n \|x_i - Ay_i - b\|^2 = \sum_{i=1}^n \|x_i - z_i\|^2 = \sum_{i=1}^n \|x_i - \bar{x}\|^2 + \sum_{i=1}^n \|z_i - \bar{z}\|^2 + n\|\bar{x} - \bar{z}\|^2 \quad (7.9)$$

to minimise D in Eq.(7.9), $\|\bar{x} - \bar{z}\| = 0$, thus the translational vector is found by:

$$b = \bar{x} - A\bar{y} \quad (7.10)$$

after subtracted the corresponding mean from each column of X and Y, the remaining problem is to find the rotational matrix, and thus Eq.(7.8) becomes to:

$$\begin{aligned} D^2 &= \sum_{i=1}^n \|x_i - Ay_i\|^2 = \text{trace}((X - YA)(X - YA)^T) \\ &= \text{trace}(XX^T) + \text{trace}(YAA^TY) - 2\text{trace}(X^TYA) = \text{trace}(XX^T) + \text{trace}(YY^T) - 2\text{trace}(X^TYA) \end{aligned} \quad (7.11)$$

since A is an orthogonal matrix, then $AA^T = I$.

In Eq.(7.11), only the last term is dependent on A, and this should be maximized to minimise D. In order to obtain A using this condition, a symmetric matrix of constraints by $\frac{1}{2}\Lambda$ can be defined, then the problem becomes to maximize Eq.(8.12),

$$\Gamma = \text{trace}(X^TYA - \frac{1}{2}\Lambda(AA^T - I)) \quad (7.12)$$

we calculate the first derivatives of Eq.(7.12) with respect to matrix A using the following lemmas:

$$trace(\Lambda AA^T) = \sum_{m=1}^p \sum_{i=1}^p \lambda_{mi} \sum_{k=1}^p a_{ik} a_{mk}, \quad \frac{\partial trace(\Lambda AA^T)}{\partial a_{ij}} = \sum_{k=1}^p \lambda_{ik} a_{kj} + \sum_{m=1}^p \lambda_{mi} a_{mj} \longrightarrow$$

$$\frac{\partial trace(\Lambda AA^T)}{\partial A} = 2\Lambda A$$

and lemmas:

$$trace(BA) = \sum_{j=1}^p \sum_{i=1}^n b_{ji} a_{ij}, \quad \frac{\partial trace(BA)}{\partial a_{ij}} = b_{ji} \longrightarrow \frac{\partial trace(BA)}{\partial A} = B^T$$

we then equate the first derivatives of Eq.(7.12) with respect to matrix A to 0. Thus,

$$Y^T X = \Lambda A \tag{7.13}$$

The SVD introduced in the previous section is applied to solve Eq.(7.13). Transforming Eq.(7.13) with SVD, then becomes:

$$Y^T X X^T Y = \Lambda A A^T \Lambda \longrightarrow (U \Sigma V^T)(V \Sigma U^T) = \Lambda^2 \longrightarrow (U \Sigma^2 U^T) = \Lambda^2 \longrightarrow$$

$$\Lambda = U \Sigma U^T \tag{7.14}$$

by substituting Eq.(7.14) to Eq.(7.13) A can be found by,

$$U \Sigma V^T = U \Sigma U^T A \longrightarrow UV^T = A$$

where U and V are $p \times p$ orthogonal matrices.

Having obtained both the translational and rotational parts, the procedure for establishing registration is as follows:

- Load the MRI images in the application.
- Use the Polaris tracking pointer to identify a desired location on the subject's head and record the coordinates with respect to the head coordinate system. The coordinates of the corresponding point on the image are recorded.
- Repeat previous step until sufficient points have been collected.

- Use the Procrustes method to complete the registration.

7.4 Summary

This chapter describes the software design of the robotic TMS system. A modular approach has been adopted; the functionalities are all designed as a module and integrated as a complete software system. The key features of QNX including microkernel, IPC, threads and PhAB have been introduced, and based on these properties, the control modules were designed using a multi-threads approach. The corresponding user interfaces were also designed using PhAB. The VAL3 robot language is also described, along with the ALTER (real-time motion control function). The conclusion and future work are presented in the following chapter.

CHAPTER 8

CONCLUSION AND FUTURE WORK

Transcranial Magnetic Stimulation (TMS) is an important technique for investigating human neurophysiology and has the potential to be a routine treatment tool for various neurological and psychiatric disorders, a robotic TMS system is developed to hold and place the stimulus coil to improve the accuracy of the stimulation results and the comfort of the subjects. The emphasis of this work is placed on the development of the architecture of a robotic TMS system; different techniques are integrated into the system to realize the basic but necessary functionalities to assist the physiologist accomplish the TMS experiments. This chapter is the final chapter of the dissertation and summarises all the research work presented, including the achievements, along with discussions for future work that would enhance the performance of the system.

8.1 Conclusions

The hypothesis of this research was that it should be possible to develop a robotic TMS system to conduct TMS experiments. It should be able to position and hold the stimulus coil over a fixed location and at a fixed orientation on the unconstrained

head of the subject; and the robot should be able to track both small head movements and maintain a small contact force between the head and coil. This goal has been achieved as follows:

A prototype robotic TMS system has been developed to cope with some of the disadvantages in the current hand-held system. It is an integration of three subsystems: force sensor, position sensor and the robot system. It also integrates several techniques to accomplish the specified tasks, including coordinate system registration, position control using external position sensor, combined position and force control using real-time path control.

To establish the adequacy of the design of the robotic system for the general procedures of TMS, a series of preliminary TMS experiments were conducted on real subjects using the hand-held system. In addition to familiarize the procedure and obtain an understanding of the problems experienced in setting up a manual TMS system, these experiments also aim to investigate the difference between Chinese and Caucasians on various parameters measured from responses to TMS. The results provide the first objective evidence that there are differences between TMS measurements across different racial groups, supporting anecdotal accounts from several laboratories. The subject's movements in terms of movement volume and speed, in addition to the contact force were addressed in the experiment. It was found that the movements were relatively small during the stimulation session, within a 300*300*300mm cube.

The operating procedure was planned for the robotic TMS system: first of all, system initialization, including initialization of the sensor and robot systems. Second, the stimulation site is defined using the robotic system. Thirdly, robot tracks both the force and movement during the stimulation session. Finally, robot arm returns to start position after experiments finished. Two operating modes are designed for the system based on the TMS procedure, a pre-operative acquisition mode and a tracking mode. Pre-operative acquisition mode is designed to fulfil the first two experiment procedures mentioned above, and the tracking mode is used during the stimulation session. The stimulation site is manually established by the neuroscientist in most of current TMS systems, and is a convenient method used by experienced

experimenters. A real-time force control algorithm allows the experimenter to manually manipulate the TMS coil. Using this ‘tele-operated’ mode, the operator can search for and acquire the desirable stimulation site just as they do with the current manual TMS equipment. In addition, an automatic or image based ‘hot-spot’ finder has also been integrated in the system.

Another key task in pre-operative acquisition mode is the registration between the coordinate systems of Polaris Spectra system and robot arm. A calibration was initially carried out for the Polaris position sensor, and the RMS accuracy was found to be 0.41mm and RMS repeatability was 0.057mm for both tools used in this application. A mathematical method yields a matrix equation of the form $AX = XB$ along with its generated equation of the form $HX = HYB$ are applied for the coordinate system registration. They were both experimentally validated in a series of tests and precise calibration. It has been demonstrated that the latter equation has better performance (around 0.55mm in translation) than the former one (around 1.45mm in translation) since the movements used to construct the equation are more independent.

After coordinate system registration, the robot movement can be guided by the external position sensor. A position tracking algorithm has been developed and validated during the test, so that a moving object can be tracked using an external position tracking system using the ALTER real-time path control available in VAL3. Furthermore, a wrist mounted force transducer was used to achieve the combined position and force control during the stimulation session. An adaptive combined position and force control architecture was developed based on a hybrid position and force control method [Raibert and Craig, 1981]. In this application, the controller is applied to tracking a moving object rather than following a trajectory on a fixed surface. A gain scheduling PI controller was implemented to the combined position and force control architecture, and its performance was validated during the experiments. It has been shown that the proposed method can be used to compensate the subject’s motion during the stimulation session. Furthermore, other control methods can also be applied in the proposed control architecture, which allows it to be extended or improved. The test result has shown that the system has met the

specifications proposed in Chapter 3. And it also showed the significant advantage and more accuracy of the robotic system over the hand-held TMS system (Figure 6.45).

An open architecture software control system along with a user interface was developed to integrate all the subsystems, using a real-time operating system, QNX 6.4.0. The code was designed to be portable so that each module of system can be easily integrated into other systems, and external software modules can also be easily adopted into this system for extension. A VAL3 application was designed for the CS8C robot controller to accomplish the robot movement, as a supervisory control from the QNX PC via TCP/IP communication. The real-time path control protocol ALTER is used to control the motion of the robot arm, and it has been demonstrated that it can control the robot motion according to the tracked object's movement.

A summary of achievements is as follows:

- This study creates the first objective evidence that there are differences between TMS measurements across different racial groups. According to these results, it would be unsafe to use normative data on these measures gathered from healthy volunteers in a single racial group to diagnose abnormality in a wider range of patients. Furthermore, gathering normative data in heterogenous populations without consideration of race will increase the variability in the measures and hence reduce their sensitivity to detect abnormality.
- The subject's movements in terms of movement volume and speed, in addition to the contact force were addressed for the TMS experiment.
- Design of a prototype robotic TMS system to assist the neuroscientist to conduct experiments, including hardware, software design, and operational procedures of the robotic TMS system.
- Application and evaluation of both equations: $AX = XB$ and $HX = HYB$ for the coordinate system registration between a robot arm and an optical position tracking system, and the latter demonstrates better performance.
- Development of a combined position and force robot control architecture using an external position and force sensing.

- Application of a gain scheduling PI force controller within the proposed position and force control architecture for simultaneous tracking of both the contact force and displacement of a moving object.
- Portable control system development, using a pseudo-RTOS for integration of new sub-systems.

8.2 Future work

A Staubli industrial robot has been used in this application, and whilst there are obvious advantages in terms of robustness of software and hardware designs are implicit, and it is a cost effective solution. However, there is one significant disadvantage of employing an industrial robot in a medical application: safety. There are two major reasons for this. Firstly, the industrial robots are capable of very high speeds, which can cause serious injury to a subject if an unpredictable collision happens. Secondly, although the robot arm has a large workspace, the workspace is not flexible enough to include all possible stimulation targets for TMS for a static position of the head. For example, it is impossible to stimulate the central prefrontal cortex and the occipital cortex in one session without changing the subject's position. On the other hand, the large workspace includes useless points for TMS stimulation and some of these points lie inside the subject's body. This is an inherent safety problem and increases the risk, because extra work needs to be done to ensure the robot does not move to such positions and collide with the subject. Although appropriate software design can mitigate the risk, improvements of hardware is the most security method to solve the problem. Robot with spherical structures (Figure 8.1) proposed by Lebosse et al.[2007] and Ramrath et al.[2007] are more suitable solutions for the TMS application. The spherical workspace is matched to the spherical surface of the subject's head, which can efficiently reduce the risk. Up to now, both systems have not been completed yet, and the performance is an open question. The limited resources did not allow to design and build a robot ourselves, for the future, it is better to seek the collaboration with one of the group to make a purposed-build TMS robot.

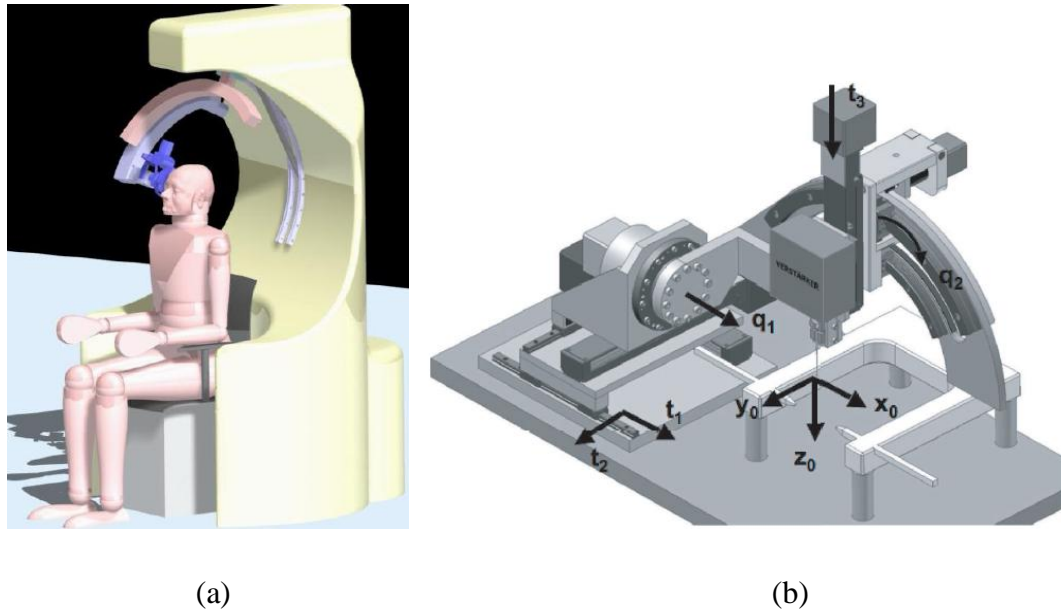


Figure 8.1 The CAD model of the robots. (a) is the model designed for TMS proposed by Lebosse et al.[2007]. (b) is the model designed by Ramrath et al.[2007] for Stereotactic Surgery.

The CS8C robot controller has a maximum update rate of 4ms, which limits the performance in terms of real-time. This can be improved using PC-based motion controller (Servotogo, SIMOTION, etc), in this way, the CS8C robot controller can be replaced by the QNX PC controller which has excellent real-time performance. Furthermore, the system is also simplified as only one PC controller is needed for the control. The position sensor used in this application has a highest tracking speed of 60Hz, which also limits the performance of the system. Tracking systems such as: optotrak (NDI, Waterloo,CA) and auctrack 500 (Atracsys, Ch. De) can achieve a fast update rate of 4000Hz. If a faster robot controller along with a sensor with higher tracking speed are employed in the system, the real-time performance of the system could be significantly improved.

There is no learning capability in this current system. The system presented a combined position and force control architecture to adopt different control methods. The gain scheduling PI controller is employed in the work, and the gains were tuned by experiments. Adding a learning capability to the position and force controller is an important improvement for the robot system to enable the robot to perform its task better, and help it adapt to any changes in its environment. There are several examples of learning paradigms applied to hybrid vision/force controller using fuzzy

control and neural networks [Kumar et al., 2011, Hsu and Fu, 1995, Visioli et al., 2010].

8.3 Presentations and publications

1. Presentations

YI, X. Presentation of Design of a robotic transcranial magnetic stimulation system. On 4th IEEE conference on CIS&RAM. Grand Copthorne Waterfront Hotel, Singapore, 28-30 June, 2010.

YI, X. Presentation of Differences between Han Chinese and Caucasians in Transcranial Magnetic Stimulation Parameters. On Conference of British Society for clinical neurophysiology. Salford royal hospital, Manchester, 4th March, 2011.

2. Publications

YI, X. & BICKER, R. 2010. Design of a robotic transcranial magnetic stimulation system. Conference on Robotics, Automation and Mechatronics (RAM), 2010 Singapore. IEEE, 78-83. (Withdraw in 2011 for preparation of patent application)

YI, X., FISHER, K. M., LAI, H. M., BICKER, R. & BAKER, S. N. 2011. Differences between Han Chinese and Caucasians in Transcranial Magnetic Stimulation Parameters. *Clin Neurophysiol*, (Under review).

REFERENCES

- ADLER JR, J. R., MURPHY, M. J., CHANG, S. D. & HANCOCK, S. L. 1999. Image-guided Robotic Radiosurgery. *Neurosurgery*, 44, 1299-1360.
- ALEXANDER, G. E., DELONG, M. R. & STRICK, P. L. 1986. Parallel Organization of Functionally Segregated Circuits Linking Basal Ganglia and Cortex. *Annual Reviews in Neuroscience*, 9, 357-381.
- ANDERSON, R. J. & SPONG, M. W. 1988. Hybrid impedance control of robotic manipulators. *IEEE Journal of Robotics and Automation*, 4, 549-556.
- ATI 2010. *Manual of the F/T Sensors*.
- BAKER, S. N. Expecting submit in 2012 Automatic 'hot-spot' finder using Bayesian inference Newcastle University.
- BAKER, S. N., OLIVIER, E. & LEMON, R. N. 1995. Task-related modulation in the amplitude of the direct volley evoked by transcranial magnetic stimulation of the motor cortex and recorded from the medullary pyramid in the monkey. *J Physiol*, 487, 69P.
- BARKER, A. T. 1999. The history and basic principles of magnetic nerve stimulation. *Electroencephalogr Clin Neurophysiol Suppl*, 51, 3-21.
- BARKER, A. T., JALINOUS, R. & FREESTON, I. L. 1985. Non-invasive magnetic stimulation of human motor cortex. *Lancet*, 1, 1106-1107.
- BASTINGS, E. P., GAGE, H. D., GREENBERG, J. P., HAMMOND, G., HERNANDEZ, L., SANTAGO, P., HAMILTON, C. A., MOODY, D. M., SINGH, K. D. & RICCI, P. E. 1998. Co-registration of cortical magnetic stimulation and functional magnetic resonance imaging. *NeuroReport*, 9, 1941-1946.
- BEN-SHACHAR, D., GAZAWI, H., RIBOYAD-LEVIN, J. & KLEIN, E. 1999. Chronic repetitive transcranial magnetic stimulation alters β -adrenergic and 5-HT₂ receptor characteristics in rat brain. *Brain Research*, 816, 78-83.
- BIRKPELLNER, W., WATZINGER, F., WANSCHITZ, F., ENISLIDIS, G., KOLLMANN, C., RAFOLT, D., NOWOTNY, R., EWERS, R. & BERGMANN, H. 1998. Systematic distortions in magnetic position digitizers. *Medical Physics*, 25, 2242-2248.
- BOHNING, D. E., SHASTRI, A., MCCONNELL, K. A., NAHAS, Z., LORBERBAUM, J. P., ROBERTS, D. R., TENEBACK, C., VINCENT, D. J. & GEORGE, M. S. 1999. A combined TMS/fMRI study of intensity-dependent TMS over motor cortex. *Biological Psychiatry*, 45, 385-394.
- BOHNING, D. E., SHASTRI, A., NAHAS, Z., LORBERBAUM, J. P., ANDERSEN, S. W., DANNELS, W. R., HAXTHAUSEN, E. U., VINCENT, D. J. & GEORGE, M. S. 1998. Echoplanar BOLD fMRI of Brain Activation Induced by Concurrent Transcranial Magnetic Stimulation. *Investigative radiology*, 33, 336-340.
- BONIFACE, S. J., MILLS, K. R. & SCHUBERT, M. 1991. Responses of single apinal motoneurons to magnetic brain stimulation in healthy subjects and patients with multiple sclerosis. *Brain*, 114, 643-662.
- BOROOJERDI, B., FOLTYS, H., KRINGS, T., SPETZGER, U., THRON, A. & TÖPPER, R. 1999. Localization of the motor hand area using transcranial magnetic stimulation and functional magnetic resonance imaging. *Clinical Neurophysiology*, 110, 699-704.

- BRAINSIGHT. 2010. *Brainsight TMS:Neuronavigation* [Online]. Available: <http://www.rogue-resolutions.com/neuronavigation/1061>.
- BURGNER, J., ZHANG, Y., RACZKOWSKY, J., WOERN, H., EGGERS, G. & MUEHLING, J. Methods for end-effector coupling in robot assisted interventions. Proceedings of IEEE International Conference on Robotics and Automation 2008. IEEE, 3395-3400.
- CANTELO, R., ROSSI, S., VARRASI, C., ULIVELLI, M., CIVARDI, C., BARTALINI, S., VATTI, G., CINCOTTA, M., BORGHERESI, A. & ZACCARA, G. 2007. Slow repetitive TMS for drug-resistant epilepsy: clinical and EEG findings of a placebo-controlled trial. *Epilepsia*, 48, 366-374.
- CANTELO, R., TARLETTI, R. & CIVARDI, C. 2002. Transcranial magnetic stimulation and Parkinson's disease. *Brain Research Reviews*, 38, 309-327.
- CHEAH, C. C., HOU, S. P., ZHAO, Y. & SLOTINE, J. J. E. 2010. Adaptive vision and force tracking control for robots with constraint uncertainty. *IEEE/ASME Transactions on Mechatronics*, 15, 389-399.
- CHEN, R. 1997a. Depression of motor cortex excitability by low-frequency transcranial magnetic stimulation. *Neurology*, 48, 1398-1403.
- CHEN, R. 1997b. Impaired inhibition in writer's cramp during voluntary muscle activation. *Neurology*, 49, 1054-1059.
- CHEN, R. 2000. Studies of human motor physiology with transcranial magnetic stimulation. *Muscle & nerve*, 23, 26-32.
- CHEN, R., LOZANO, A. M. & ASHBY, P. 1999. Mechanism of the silent period following transcranial magnetic stimulation Evidence from epidural recordings. *Experimental Brain Research*, 128, 539-542.
- CHISTYAKOV, A. V., SOUSTIEL, J. F., HAFNER, H., TRUBNIK, M., LEVY, G. & FEINSOD, M. 2001. Excitatory and inhibitory corticospinal responses to transcranial magnetic stimulation in patients with minor to moderate head injury. *Journal of Neurology, Neurosurgery & Psychiatry*, 70, 580-587.
- CICINELLI, P., TRAVERSA, R. & ROSSINI, P. M. 1997. Post-stroke reorganization of brain motor output to the hand: a 2-4 month follow-up with focal magnetic transcranial stimulation. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, 105, 438-450.
- CLASSEN, J., SCHNITZLER, A., BINKOFSKI, F., WERHAHN, K. J., KIM, Y. S., KESSLER, K. R. & BENECKE, R. 1997. The motor syndrome associated with exaggerated inhibition within the primary motor cortex of patients with hemiparetic. *Brain*, 120, 605-619.
- COHEN, L. G., BANDINELLI, S., TOPKA, H. R., FUHR, P., ROTH, B. J. & HALLETT, M. 1991. Topographic maps of human motor cortex in normal and pathological conditions: mirror movements, amputations and spinal cord injuries. *Electroencephalogr Clin Neurophysiol Suppl*, 43, 36-50.
- COLCHESTER, A. C. F., ZHAO, J., HOLTON-TAINTER, K. S., HENRI, C. J., MAITLAND, N., ROBERTS, P. T. E., HARRIS, C. G. & EVANS, R. J. 1996. Development and preliminary evaluation of VISLAN, a surgical planning and guidance system using intra-operative video imaging. *Medical Image Analysis*, 1, 73-90.

- CONWAY, J. H., SMITH, D. A. & DIXON, G. 2004. On quaternions and octonions: their geometry, arithmetic, and symmetry. *The Mathematical Intelligencer*, 26, 75-77.
- COUTURIER, J. L. 2005. Efficacy of rapid-rate repetitive transcranial magnetic stimulation in the treatment of depression: a systematic review and meta-analysis. *J Psychiatry Neuroscience*, 30, 83-90.
- DANIILIDIS, K. 1999. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18, 286.
- DAVIES, B. L. 1996. A discussion of safety issues for medical robots. *Computer-Integrated Surgery*, 287-296.
- DAVIES, B. L., HIBBERD, R. D., NG, W. S., TIMONEY, A. G. & WICKHAM, J. E. A. A surgeon robot for prostatectomies. Proceedings of Fifth International Conference on Advanced Robotics, 1991. IEEE, 871-875.
- DAY, B. L., ROTHWELL, J. C., THOMPSON, P. D., DICK, J. P. R., COWAN, J. M. A., BERARDELLI, A. & MARSDEN, C. D. 1987. Motor cortex stimulation in intact man: 2. Multiple descending volleys. *Brain*, 110, 1191-1209.
- DESIATO, M. T., PALMIERI, M. G., GIACOMINI, P., SCALISE, A., ARCIPRETE, F. & CARAMIA, M. D. 1999. The effect of riluzole in amyotrophic lateral sclerosis: a study with cortical stimulation. *J Neurol Sci*, 169, 98-107.
- DI LAZZARO, V., RESTUCCIA, D., OLIVIERO, A., PROFICE, P., FERRARA, L., INSOLA, A., MAZZONE, P., TONALI, P. & ROTHWELL, J. C. 1998. Effects of voluntary contraction on descending volleys evoked by transcranial stimulation in conscious humans. *J Physiol*, 508 (Pt 2), 625-33.
- DORNAIKA, F. & HORAUD, R. 1998. Simultaneous robot-world and hand-eye calibration. *IEEE transactions on Robotics and Automation*, 14, 617 - 622.
- DOWLER, N. J. 1995. Applying software dependability principles to medical robotics. *Computing & Control Engineering Journal*, 6, 222-225.
- EDGLEY, S. A., EYRE, J. A., LEMON, R. N. & MILLER, S. 1990. Excitation of the corticospinal tract by electromagnetic and electrical stimulation of the scalp in the macaque monkey. *The Journal of Physiology*, 425, 301-320.
- ENGEL, D., KORB, W., RACZKOWSKY, J., HASSFELD, S. & WÖRN, H. Location decision for a robot milling complex trajectories in craniofacial surgery. 2003. Proceedings of the 17th International Congress and Exhibition on Computer Assisted Radiology and Surgery, 760-765.
- ESCUDERO, J. V., SANCHO, J., BAUTISTA, D., ESCUDERO, M. & LOPEZ-TRIGO, J. 1998. Prognostic Value of Motor Evoked Potential Obtained by Transcranial Magnetic Brain Stimulation in Motor Function Recovery in Patients With Acute Ischemic Stroke. *Stroke*, 29, 1854-1859.
- ETTINGER, G. J. 1997. *Hierarchical three-dimensional medical image registration*. Massachusetts Institute of Technology.
- ETTINGER, G. J., GRIMSON, W. E. L., LEVENTON, M. E., KIKINIS, R., GUGINO, V., COTE, W., KARAPELOU, M., AGLIO, L., SHENTON, M. & POTTS, G. Non-invasive functional brain mapping using registered transcranial magnetic stimulation. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 1996 San Francisco, CA. 332-41.

- ETTINGER, G. J., LEVENTON, M. E., GRIMSON, W. E. L., KIKINIS, R., GUGINO, L., COTE, W., SPRUNG, L., AGLIO, L., SHENTON, M. E. & POTTS, G. 1998. Experimentation with a transcranial magnetic stimulation system for functional brain mapping. *Medical Image Analysis*, 2, 133-142.
- FEI, B., NG, W. S., CHAUHAN, S. & KWOH, C. K. 2001. The safety issues of medical robotics. *Reliability Engineering and system safety*, 73, 183-192.
- FEINSOD, M., KREININ, B., CHISTYAKOV, A. & KLEIN, E. 1998. Preliminary evidence for a beneficial effect of low-frequency, repetitive transcranial magnetic stimulation in patients with major depression and schizophrenia. *Depress Anxiety*, 7, 65-8.
- FERBERT, A., PRIORI, A., ROTHWELL, J. C., DAY, B. L., COLEBATCH, J. G. & MARSDEN, C. D. 1992. Interhemispheric inhibition of the human motor cortex. *The Journal of Physiology*, 453, 525-546.
- FIGIEL, G. S., EPSTEIN, C., MCDONALD, W. M., AMAZON-LEECE, J., FIGIEL, L., SALDIVIA, A. & GLOVER, S. 1998. The Use of Rapid-Rate Transcranial Magnetic Stimulation (rTMS) in Refractory Depressed Patients. *J Neuropsychiatry Clin Neurosci*, 10, 20-25.
- FINKE, M., FADINI, T., KANTELHARDT, S., GIESE, A., MATTHAUS, L. & SCHWEIKARD, A. Brain-mapping using robotized TMS. 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008. 3929-3932.
- FINLAY, P. A. & ORNSTEIN, M. H. 1995. Controlling the movement of a surgical laparoscope. *Engineering in Medicine and Biology Magazine, IEEE*, 14, 289-291.
- FOX, P. T., NARAYANA, S., TANDON, N., SANDOVAL, H., FOX, S. P., KOCHUNOV, P. & LANCASTER, J. L. 2004. Column-based model of electric field excitation of cerebral cortex. *Human Brain Mapping*, 22, 1-14.
- FREGNI, F., OTACHI, P. T. M., DO VALLE, A., BOGGIO, P. S., THUT, G., RIGONATTI, S. P., PASCUAL-LEONE, A. & VALENTE, K. D. 2006. A randomized clinical trial of repetitive transcranial magnetic stimulation in patients with refractory epilepsy. *Annals of neurology*, 60, 447-455.
- GEORGE, M. S., NAHAS, Z., KOZEL, F. A., LI, X., DENSLOW, S., YAMANAKA, K., MISHORY, A., FOUST, M. J. & BOHNING, D. E. 2002. Mechanisms and State of the Art of Transcranial Magnetic Stimulation. *The Journal of ECT*, 18, 170-181.
- GEORGE, M. S., WASSERMANN, E. M., WILLIAMS, W. A., CALLAHAN, A., KETTER, T. A., BASSER, P., HALLETT, M. & POST, R. M. 1995. Daily repetitive transcranial magnetic stimulation (rTMS) improves mood in depression. *NEUROREPORT-OXFORD*, 6, 1853-1856.
- GHABRA, M. B., HALLETT, M. & WASSERMANN, E. M. 1999. Simultaneous repetitive transcranial magnetic stimulation does not speed fine movement in PD. *Neurology*, 52, 768-770.
- GHOSH, B. K., XI, N. & TARN, T. J. 2000. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Transactions on Control Systems Technology*, 8, 635-645.
- GIULIANI, M., LENZ, C., MÜLLER, T., RICKERT, M. & KNOLL, A. 2010. Design principles for safety in human-robot interaction. *International Journal of Social Robotics*, 2, 253-274.

- GLOSSOP, N. D. 2009. Advantages of Optical Compared with Electromagnetic Tracking. *The Journal of Bone and Joint Surgery*, 91, 23.
- GOMES, P. 2011. Surgical robotics: Reviewing the past, analysing the present, imagining the future. *Robotics and Computer-Integrated Manufacturing*, 27, 261-266.
- GRIMSON, W. E. L., ETTINGER, G. J., WHITE, S. J., LOZANO-PEREZ, T., WELLS III, W. M. & KIKINIS, R. 1996. An automatic registration method for frameless stereotaxy, imageguided surgery, and enhanced reality visualization. *Medical Imaging, IEEE Transactions on*, 15, 129-140.
- GRISARU, N., CHUDAKOV, B., YAROSLAVSKY, Y. & BELMAKER, R. H. 1998. Transcranial magnetic stimulation in mania: a controlled study. *Am J Psychiatry*, 155, 1608-1610.
- GUGGISBERG, A. G., DUBACH, P., HESS, C. W., WÜTHRICH, C. & MATHIS, J. 2001. Motor evoked potentials from masseter muscle induced by transcranial magnetic stimulation of the pyramidal tract: the importance of coil orientation. *Clinical Neurophysiology*, 112, 2312-2319.
- HADDADIN, S., ALBU-SCHÄFFER, A. & HIRZINGER, G. 2009. Requirements for safe robots: measurements, analysis and new insights. *The international journal of robotics research*, 28, 1507-1527.
- HALLETT, M. 1998. The neurophysiology of dystonia. *Arch Neurol*, 55, 601-603.
- HALLETT, M. 2000. Transcranial magnetic stimulation and the human brain. *Nature*, 406, 147-150.
- HALLETT, M. 2007. Transcranial Magnetic Stimulation: A Primer. *Neuron*, 55, 187-199.
- HAMILTON, D. L., VISINSKY, M. L., BENNETT, J. K., CAVALLARO, J. R. & WALKER, I. D. Fault tolerant algorithms and architectures for robotics. Proceedings of 7th Mediterranean Electrotechnical Conference, 1994. 1034-1036.
- HAMILTON, M. 1960. A rating scale for depression. *J Neurol Neurosurg Psychiatry*, 23, 56-62.
- HAMILTON, W. R. On a new species of imaginary quantities connected with a theory of quaternions. Proceedings of the Royal Irish Academy, 1844. 424-434.
- HANAJIMA, R., UGAWA, Y., OKABE, S., YUASA, K., SHIIO, Y., IWATA, N. K. & KANAZAWA, I. 2001. Interhemispheric interaction between the hand motor areas in patients with cortical myoclonus. *Clinical Neurophysiology*, 112, 623-626.
- HAWES, N., WYATT, J. & SLOMAN, A. 2006. An architecture schema for embodied cognitive systems. *SCHOOL OF COMPUTER SCIENCE RESEARCH REPORTS-UNIVERSITY OF BIRMINGHAM CSR*, 12.
- HERWIG, U., PADBERG, F., UNGER, J., SPITZER, M. & SCHÖNFELDT-LECUONA, C. 2001a. Transcranial magnetic stimulation in therapy studies: examination of the reliability of "standard" coil positioning by neuronavigation. *Biological Psychiatry*, 50, 58-61.
- HERWIG, U., SATRAPI, P. & SCHÖNFELDT-LECUONA, C. 2003. Using the international 10-20 EEG system for positioning of transcranial magnetic stimulation. *Brain topography*, 16, 95-99.

- HERWIG, U., SCHÖNFELDT-LECUONA, C., WUNDERLICH, A. P., VON TIESENHAUSEN, C., THIELSCHER, A., WALTER, H. & SPITZER, M. 2001b. The navigation of transcranial magnetic stimulation. *Psychiatry Research: Neuroimaging*, 108, 123-131.
- HESS, C. W., MILLS, K. R., MURRAY, N. M. F. & SCHRIEFER, T. N. 1987. Magnetic brain stimulation: Central motor conduction studies in multiple sclerosis. *Annals of Neurology*, 22, 744-752.
- HOGAN, N. Impedance control: An approach to manipulation. American Control Conference, 1985. IEEE, 304-313.
- HOGAN, N. Stable execution of contact tasks using impedance control. Proceedings of IEEE International Conference on Robotics and Automation, 1987. IEEE, 1047-1054.
- HORAUD, R. & DORNAIKA, F. 1995. Hand-eye calibration. *The international journal of robotics research*, 14, 195.
- HOSODA, K., IGARASHI, K. & ASADA, M. 1998. Adaptive hybrid control for visual and force servoing in an unknown environment. *Robotics & Automation Magazine, IEEE*, 5, 39-43.
- HSU, F. Y. & FU, L. C. A new design of adaptive fuzzy hybrid force/position controller for robot manipulators. 1995. IEEE, 863-868 vol. 1.
- HUTCHINSON, S., HAGER, G. D. & CORKE, P. I. 1996. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12, 651-670.
- INGHILLERI, M., BERARDELLI, A., CRUCCU, G. & MANFREDI, M. 1993. Silent period evoked by transcranial stimulation of the human cortex and cervicomedullary junction. *The Journal of Physiology*, 466, 521-534.
- JASPER, H. H. 1958. The 10-20 electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology*, 10, 371-375.
- KALFAS, I. H., KORMOS, D. W., MURPHY, M. A., MCKENZIE, R. L., BARNETT, G. H., BELL, G. R., STEINER, C. P., TRIMBLE, M. B. & WEISENBERGER, J. P. 1995. Application of frameless stereotaxy to pedicle screw fixation of the spine. *Journal of Neurosurgery*, 83, 641-647.
- KARAS, C. S. & CHIOCCA, E. A. 2007. Neurosurgical robotics: a review of brain and spine applications. *Journal of Robotic Surgery*, 1, 39-43.
- KAUS, M., ING, D., STEINMEIER, R., SPORER, T., INF, D., GANSLANDT, O. & FAHLBUSCH, R. 1997. Technical Accuracy of a Neuronavigation System Measured with a High-precision Mechanical Micromanipulator. *Neurosurgery*, 41, 1431-1437.
- KHADEM, R., YEH, C. C., SADEGHI-TEHRANI, M., BAX, M. R., JOHNSON, J. A., WELCH, J. N., WILKINSON, E. P. & SHAHIDI, R. 2000. Biomedical Paper Comparative Tracking Error Analysis of Five Different Optical Tracking Systems. *Computer Aided Surgery*, 5, 98-107.
- KHEDR, E. M., AHMED, M. A., FATHY, N. & ROTHWELL, J. C. 2005. Therapeutic trial of repetitive transcranial magnetic stimulation after acute ischemic stroke. *Neurology*, 65, 466-468.
- KHODABANDEHLOO, K. 1996. Analyses of robot systems using fault and event trees: case studies. *Reliability Engineering & System Safety*, 53, 247-264.
- KIERS, L., CROS, D., CHIAPPA, K. H. & FANG, J. 1993. Variability of motor potentials evoked by transcranial magnetic stimulation.

Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section, 89, 415-423.

- KIM, V. B., CHAPMAN III, W. H. H., ALBRECHT, R. J., BAILEY, B. M., YOUNG, J. A., NIFONG, L. & CHITWOOD JR, W. R. 2002. Early experience with telemanipulative robot-assisted laparoscopic cholecystectomy using da Vinci. *Surgical Laparoscopy Endoscopy & Percutaneous Techniques*, 12, 33-40.
- KIMBRELL, T. A., LITTLE, J. T., DUNN, R. T., FRYE, M. A., GREENBERG, B. D., WASSERMANN, E. M., REPELLA, J. D., DANIELSON, A. L., WILLIS, M. W. & BENSON, B. E. 1999. Frequency dependence of antidepressant response to left prefrontal repetitive transcranial magnetic stimulation (rTMS) as a function of baseline cerebral glucose metabolism. *Biological Psychiatry*, 46, 1603-1613.
- KIMISKIDIS, V. K., PAPAGIANNOPOULOS, S., SOTIRAKOGLU, K., KAZIS, D. A., KAZIS, A. & MILLS, K. R. 2005. Silent period to transcranial magnetic stimulation: construction and properties of stimulus-response curves in healthy volunteers. *Experimental Brain Research*, 163, 21-31.
- KLEINJUNG, T., EICHHAMMER, P., LANGGUTH, B., JACOB, P., MARIENHAGEN, J., HAJAK, G., WOLF, S. R. & STRUTZ, J. 2005. Long-term effects of repetitive transcranial magnetic stimulation (rTMS) in patients with chronic tinnitus. *Otolaryngology-Head and Neck Surgery*, 132, 566-569.
- KNAPPE, P., GROSS, I., PIECK, S., WAHRBURG, J., KUENZLER, S. & KERSCHBAUMER, F. Position control of a surgical robot by a navigation system. Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.
- KOBAYASHI, M. & PASCUAL-LEONE, A. 2003. Transcranial magnetic stimulation in neurology. *Lancet Neurol*, 2, 145-56.
- KOCH, G., FRANCA, M., ALBRECHT, U. V., CALTAGIRONE, C. & ROTHWELL, J. C. 2006. Effects of paired pulse TMS of primary somatosensory cortex on perception of a peripheral electrical stimulus. *Experimental Brain Research*, 172, 416-424.
- KORB, W., ENGEL, D., BOESECKE, R., EGGERS, G., KOTRIKOVA, B., MARMULLA, R., RACZKOWSKY, J., WÖRN, H., MÜHLING, J. & HASSFELD, S. 2003. Development and First Patient Trial of a Surgical Robot for Complex Trajectory Milling. *Computer Aided Surgery*, 8, 247-256.
- KORB, W., MARMULLA, R., RACZKOWSKY, J., MÜHLING, J. & HASSFELD, S. 2004. Robots in the operating theatre—chances and challenges. *International Journal of Oral & Maxillofacial Surgery*, 33, 721-732.
- KOZEL, F. A., NAHAS, Z., DEBRUX, C., MOLLOY, M., LORBERBAUM, J. P., BOHNING, D., RISCH, S. C. & GEORGE, M. S. 2000. How coil-cortex distance relates to age, motor threshold, and antidepressant response to repetitive transcranial magnetic stimulation. *J Neuropsychiatry Clin Neurosci*, 12, 376-384.
- KRAGIC, D. 2001. *Visual servoing for manipulation: Robustness and integration issues*. Phd, KTH, Numerical Analysis and Computer Science.
- KRINGS, T., BUCHBINDER, B. R., BUTLER, W. E., CHIAPPA, K. H., JIANG, H. J., ROSEN, B. R. & COSGROVE, G. R. 1997. Stereotactic transcranial

- magnetic stimulation: correlation with direct electrical cortical stimulation. *Neurosurgery*, 41, 1319.
- KRINGS, T., CHIAPPA, K. H., FOLTYS, H., REINGES, M. H., COSGROVE, R. G. & THRON, A. 2001. Introducing navigated transcranial magnetic stimulation as a refined brain mapping methodology. *Neurosurgical Review*, 24, 171-179.
- KUJIRAI, T., CARAMIA, M. D., ROTHWELL, J. C., DAY, B. L., THOMPSON, P. D., FERBERT, A., WROE, S., ASSELMAN, P. & MARSDEN, C. D. 1993. Corticocortical inhibition in human motor cortex. *The Journal of Physiology*, 471, 501-519.
- KUMAR, N., PANWAR, V., SUKAVANAM, N., SHARMA, S. P. & BORM, J. H. 2011. Neural network based hybrid force/position control for robot manipulators. *International Journal of Precision Engineering and Manufacturing*, 12, 419-426.
- KWOH, Y. S., HOU, J., JONCKHEERE, E. A., HAYATI, S., RES, C. T., CENTER, M. M. & BEACH, L. 1988. A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35, 153-160.
- LANCASTER, J. L., NARAYANA, S., WENZEL, D., LUCKEMEYER, J., ROBY, J. & FOX, P. 2004. Evaluation of an Image-Guided, Robotically Positioned Transcranial Magnetic Stimulation System. *Human Brain Mapping*, 22, 329-340.
- LEBOSSE, C., RENAUD, P., BAYLE, B., DE MATHELIN, M., PICCIN, O. & FOUCHER, J. A robotic system for automated image-guided transcranial magnetic stimulation. Proceedings of IEEE/NIH Life Science Systems and Applications Workshop, 2007 LISA. 55-58.
- LEBOSSÉ, C., RENU, P., BAYLE, B., DE MATHELIN, M., PIECIN, O., LAROCHE, E. & FOUCHER, J. 2006. Robotic image-guided transcranial magnetic stimulation. *International Journal of Computer Assisted Radiology and Surgery*, 1, 137.
- LEE, J. H. & VAN DONKELAAR, P. 2006. The human dorsal premotor cortex generates on-line error corrections during sensorimotor adaptation. *Journal of Neuroscience*, 26, 3330-3334.
- LEITE, A. C. & LIZARRALDE, F. 2009. Hybrid Adaptive Vision—Force Control for Robot Manipulators Interacting with Unknown Surfaces. *The international journal of robotics research*, 28, 911-926.
- LISANBY, S. H., GUTMAN, D., LUBER, B., SCHROEDER, C. & SACKEIM, H. A. 2001. Sham TMS: intracerebral measurement of the induced electrical field and the induction of motor-evoked potentials. *Biological Psychiatry*, 49, 460-463.
- MAEDA, F., KEENAN, J. P., TORMOS, J. M., TOPKA, H. & PASCUAL-LEONE, A. 2000. Modulation of corticospinal excitability by repetitive transcranial magnetic stimulation. *Clinical Neurophysiology*, 111, 800-805.
- MANWARING, K. H., MANWARING, M. L. & MOSS, S. D. 1994. Magnetic field guided endoscopic dissection through a burr hole may avoid more invasive craniotomies. A preliminary report. *Acta neurochirurgica. Supplement*, 61, 34-39.

- MARMULLA, R., HILBERT, M. & NIEDERDELLMANN, H. 1997. Inherent precision of mechanical, infrared and laser-guided navigation systems for computer-assisted surgery. *Journal of Cranio-Maxillofacial Surgery*, 25, 192-197.
- MARTIN, P. I., NAESER, A., MARIATORMOS, J., NICHOLAS, M., KURLAND, J., FREGNI, F., SEEKINS, H., DORON, K. & PASCUAL-LEONE, A. 2004. Transcranial Magnetic Stimulation as a Complementary Treatment for Aphasia. *Clinical Neurophysiology*, 15, 16-26.
- MATTHAUS, L., GIESE, A., WERTHEIMER, D. & SCHWEIKARD, A. 2005. Planning and Analyzing Robotized TMS Using Virtual Reality. *Studies in Healthy Technology and Informatics*, 119, 373-378.
- MATTHAUS, L., TRILLENBERG, P., BODENSTEINER, C., GIESE, A. & SCHWEIKARD, A. 2006. Robotized TMS for motion compensated navigated brain stimulation. *International Journal of Computer Assisted Radiology and Surgery*, 1, 139-143.
- MENKES, D. L., BODNAR, P., BALLESTEROS, R. A. & SWENSON, M. R. 1999. Right frontal lobe slow frequency repetitive transcranial magnetic stimulation (SF r-TMS) is an effective treatment for depression: a case-control pilot study of safety and efficacy. *J Neurol Neurosurg Psychiatry*, 67, 113-115.
- MERTON, P. A. & MORTON, H. B. 1980. Stimulation of the cerebral cortex in the intact human subject. *Nature*, 285, 227-227.
- MILLS, K. R. & NITHI, K. A. 1997. Corticomotor threshold is reduced in early sporadic amyotrophic lateral sclerosis. *Muscle & nerve*, 20, 1137-1141.
- MORGAN, P. S., CARTER, T., DAVIS, S., SEPEHRI, A., PUNT, J., BYRNE, P., MOODY, A. & FINLAY, P. The application accuracy of the Pathfinder neurosurgical robot. Proceedings of the 17th International Congress and Exhibition on Computer Assisted Radiology and Surgery, 2003. Elsevier, 561-567.
- NDI 2005. Manual of Polaris Spectra.
- NEGGERS, S. F. W., LANGERAK, T. R., SCHUTTER, D., MANDL, R. C. W., RAMSEY, N. F., LEMMENS, P. J. J. & POSTMA, A. 2004. A stereotactic method for image-guided transcranial magnetic stimulation validated with fMRI and motor-evoked potentials. *Neuroimage*, 21, 1805-1817.
- NELSON, B. J., MORROW, J. D. & KHOSLA, P. K. Improved force control through visual servoing. Proceedings of the American Control Conference, 1995. IEEE, 380-386 vol. 1.
- NOIRHOMME, Q., FERRANT, M., VANDERMEEREN, Y., OLIVIER, E., MACQ, B. & CUISENAIRE, O. 2004. Registration and real-time visualization of transcranial magnetic stimulation with 3-D MR images. *IEEE Transactions on Biomedical Engineering*, 51, 1994-2005.
- OKAMOTO, M., DAN, H., SAKAMOTO, K., TAKEO, K., SHIMIZU, K., KOHNO, S., ODA, I., ISOBE, S., SUZUKI, T. & KOHYAMA, K. 2004. Three-dimensional probabilistic anatomical cranio-cerebral correlation via the international 10–20 system oriented for transcranial functional brain mapping. *Neuroimage*, 21, 99-111.
- PASCUAL-LEONE, A. 1994. Akinesia in Parkinson's disease. II. Effects of subthreshold repetitive transcranial motor cortex stimulation. *Neurology*, 44, 892-898.

- PASCUAL-LEONE, A., DAVEY, N. J., ROTHWELL, J., WASSERMAN, E. M. & PURI, B. K. 2002. *Handbook of transcranial magnetic stimulation*, Arnold London.
- PASCUAL-LEONE, A., RUBIO, B., PALLARDO, F. & CATALA, M. D. 1996. Rapid-rate transcranial magnetic stimulation of left dorsolateral prefrontal cortex in drug-resistant depression. *Lancet-London*, 233-237.
- PASCUAL-LEONE, A., VALLS-SOLE, J., WASSERMANN, E. M. & HALLETT, M. 1994. Responses to rapid-rate transcranial magnetic stimulation of the human motor cortex. *Brain*, 117, 847-858.
- PELIZZARI, C. A., CHEN, G. T. Y., SPELBRING, D. R., WEICHSELBAUM, R. R. & CHEN, C. T. 1989. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. *Journal of computer assisted tomography*, 13, 20-26.
- PRESS, W. H. 2007. *Numerical recipes: the art of scientific computing*, Cambridge university press.
- PRIORI, A., BERARDELLI, A., INGHILLERI, M., ACCORNERO, N. & MANFREDI, M. 1994. Motor cortical inhibition and the dopaminergic system: Pharmacological changes in the silent period after transcranial brain stimulation in normal subjects, patients with Parkinson's disease and drug-induced parkinsonism. *Brain*, 117, 317-323.
- QNX 2008. QNX Neutrino Programmer Guide.
- RAD 2009. *Manual of Ultimatic™ Pneumatic Collision Sensors*.
- RAIBERT, M. H. & CRAIG, J. J. 1981. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102, 126-133.
- RAMRATH, L., HOFMANN, U. G. & SCHWEIKARD, A. Spherical assistant for stereotactic surgery. 2007. IEEE, 859-864.
- REINHARDT, H. F., TRIPPEL, M., WESTERMANN, B., HORSTMANN, G. A. & GRATZL, O. 1996. Computer assisted brain surgery for small lesions in the central sensorimotor region. *Acta neurochirurgica*, 138, 200-205.
- RICHTER L., MATTHÄUS L., SCHLAEFER A. & A, S. Fast robotic compensation of spontaneous head motion during Transcranial Magnetic Stimulation (TMS). Proceedings of UKACC International Conference on CONTROL, 2010 UK. 872-877.
- RIDDING, M. C. & ROTHWELL, J. C. 1997. Stimulus/response curves as a method of measuring motor cortical excitability in man. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, 105, 340-344.
- ROGERS, D. F. & ADAMS, J. A. 1989. *Mathematical elements for computer graphics*, McGraw-Hill Higher Education.
- ROSSINI, P. M., BARKER, A. T., BERARDELLI, A., CARAMIA, M. D., CARUSO, G., CRACCO, R. Q., DIMITRIJEVIC, M. R., HALLETT, M., KATAYAMA, Y. & LUCKING, C. H. 1994. Non-invasive electrical and magnetic stimulation of the brain, spinal cord and roots: basic principles and procedures for routine clinical application. Report of an IFCN committee. *Electroencephalogr Clin Neurophysiol*, 91, 79-92.
- ROSSINI, P. M. & ROSSI, S. 1998. Clinical applications of motor evoked potentials. *Electroencephalography and Clinical Neurophysiology*, 106, 180-194.

- SANDERSON, A. C. & WEISS, L. E. 1980. Image-based visual servo control using relational graph error signals. *Proc. IEEE*, 1074-1077.
- SANGER, T. D., GARG, R. R. & CHEN, R. 2001. Interactions between two different inhibitory systems in the human motor cortex. *The Journal of Physiology*, 530, 307-317.
- SCHÖNFELDT-LECUONA, C., THIELSCHER, A., FREUDENMANN, R. W., KRON, M., SPITZER, M. & HERWIG, U. 2005. Accuracy of stereotaxic positioning of transcranial magnetic stimulation. *Brain topography*, 17, 253-259.
- SCHWARTZ, D., LEMOINE, D., POISEAU, E. & BARILLOT, C. 1996. Registration of MEG/EEG data with 3D MRI: Methodology and precision issues. *Brain Topography*, 9, 101-116.
- SEBORG, D., EDGAR, T. F. & MELLICHAMP, D. 2006. *Process dynamics & control*, Wiley-India.
- SHAOHUA, T., HANG, C. C. & CHAI, J. S. 1997. Gain Scheduling: from Conventional to Neuro-fuzzy. *Automatica*, 33, 411-419.
- SHIMIZU, T., HOSAKI, A., HINO, T., SATO, M., KOMORI, T., HIRAI, S. & ROSSINI, P. M. 2002. Motor cortical disinhibition in the unaffected hemisphere after unilateral cortical stroke. *Brain*, 125, 1896-1907.
- SHIU, Y. & AHMAD, S. Finding the mounting position of a sensor by solving a homogeneous transform equation of the form $AX = XB$. Proceedings of IEEE International Conference on Robotics and Automation, 1987.
- SHIU, Y. C. & AHMAD, S. 1989. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$. *IEEE Transactions on Robotics and Automation*, 5, 16-29.
- SIEBNER, H. R., LEE, L. & BESTMANN, S. 2003. Interleaving TMS with functional MRI: now that it is technically feasible how should it be used? *Clinical Neurophysiology*, 114, 1997-1999.
- SIEBNER, H. R. & ROTHWELL, J. 2003. Transcranial magnetic stimulation: new insights into representational cortical plasticity. *Experimental brain research*, 148, 1-16.
- SIEBNER, H. R., TORMOS, J. M., BAUMANN, A. O. C., AUER, C., CATALA, M. D., CONRAD, B. & PASCUAL-LEONE, A. 1999. Low-frequency repetitive transcranial magnetic stimulation of the motor cortex in writer's cramp. *Neurology*, 52, 529-529.
- SJÖ, K., LÓPEZ, D. G., PAUL, C., JENSFELT, P. & KRAGIC, D. 2008. Object search and localization for an indoor mobile robot. *Journal of Computing and Information Technology*, 17, 67-80.
- SPARING, R., BUELTE, D., MEISTER, I. G., PAUS, T. & FINK, G. R. 2008. Transcranial magnetic stimulation and the challenge of coil placement: A comparison of conventional and stereotaxic neuronavigational strategies. *HUMAN BRAIN MAPPING*, 29, 82-96.
- STROBL, K. H. & HIRZINGER, G. 2006. Optimal hand-eye calibration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- TAYLOR, R. H., MITTELSTADT, B. D., PAUL, H. A., HANSON, W., KAZANZIDES, P., ZUHARS, J. F., WILLIAMSON, B., MUSITS, B. L., GLASSMAN, E. & BARGAR, W. L. 1994. An image-directed robotic

- system for precise orthopaedic surgery. *IEEE Transactions on Robotics and Automation*, 10, 261-275.
- TAYLOR, R. H., PAUL, H. A., KAZANZIDES, P., MITTELSTADT, B. D., HANSON, W., ZUHARS, J., WILLIAMSON, B., MUSITS, B., GLASSMAN, E. & BARGAR, W. L. Taming the bull: safety in a precise surgical robot. Proceedings of Fifth International Conference on Advanced Robotics, 1991. 865-870.
- TERGAU, F., WASSERMANN, E. M., PAULUS, W. & ZIEMANN, U. 1999. Lack of clinical improvement in patients with Parkinson's disease after low and high frequency repetitive transcranial magnetic stimulation. *Electroencephalogr Clin Neurophysiol Suppl*, 51, 281-8.
- THIELSCHER, A. & KAMMER, T. 2002. Linking physics with physiology in TMS: a sphere field model to determine the cortical stimulation site in TMS. *Neuroimage*, 17, 1117-1130.
- TINGS, T., LANG, N., TERGAU, F., PAULUS, W. & SOMMER, M. 2005. Orientation-specific fast rTMS maximizes corticospinal inhibition and facilitation. *Experimental Brain Research*, 164, 323-333.
- TOFTS, P. S. 1990. The distribution of induced currents in magnetic stimulation of the nervous system. *Physics in Medicine and Biology*, 35, 1119-1128.
- TRIGGS, W. J., MACDONELL, R. A. L., CROS, D., CHIAPPA, K. H., SHAHANI, B. T. & DAY, B. J. 1992. Motor inhibition and excitation are independent effects of magnetic cortical stimulation. *Annals of Neurology*, 32, 345-351.
- TSAI, R. Y., LENZ, R. K., CENTER, I. & HEIGHTS, Y. 1989. A new technique for fully autonomous and efficient 3D robotic hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5, 345-358.
- UGAWA, Y., ROTHWELL, J. C., DAY, B. L., THOMPSON, P. D. & MARSDEN, C. D. 1991. Percutaneous electrical stimulation of corticospinal pathways at the level of the pyramidal decussation in humans. *Ann Neurol*, 29, 418-27.
- VALLS-SOLE, J., PASCUAL-LEONE, A., BRASIL-NETO, J. P., CAMMAROTA, A., MCSHANE, L. & HALLETT, M. 1994. Abnormal facilitation of the response to transcranial magnetic stimulation in patients with Parkinson's disease. *Neurology*, 44, 735.
- VARLEY, P. 1999. Techniques for development of safety-related software for surgical robots. *IEEE Transactions on Information Technology in Biomedicine*, 3, 261-267.
- VARMA, T. R. K., ELDRIDGE, P. R., FORSTER, A., FOX, S., FLETCHER, N., STEIGER, M., LITTLECHILD, P., BYRNE, P., SINNOTT, A. & TYLER, K. 2003. Use of the NeuroMate stereotactic robot in a frameless mode for movement disorder surgery. *Stereotact Funct Neurosurg*, 80, 132-135.
- VISIOLI, A., ZILIANI, G. & LEGNANI, G. 2010. Iterative-learning hybrid force/velocity control for contour tracking. *IEEE Transactions on Robotics and Automation*, 26, 388-393.
- WAGNER, T., VALERO-CABRE, A. & PASCUAL-LEONE, A. 2007. Noninvasive Human Brain Stimulation. *ANNUAL REVIEW OF BIOMEDICAL ENGINEERING*, 9, 527-565.
- WAGNER, T. A., ZAHN, M., GRODZINSKY, A. J. & PASCUAL-LEONE, A. 2004. Three-dimensional head model simulation of transcranial magnetic stimulation. *IEEE Transactions on Biomedical Engineering*, 51, 1586-1598.

- WAHRBURG, J., GROSS, I., KNAPPE, P., PIECK, S., KÜNZLER, S. & KERSCHBAUMER, F. An interactive mechatronic assistance system to support surgical interventions. Proceedings of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery, 2004. Elsevier, 431-436.
- WAHRBURG, J. & KERSCHBAUMER, F. Using robots to increase the accuracy of surgical interventions. Industrial Electronics Society, 1998. IECON'98. Proceedings of the 24th Annual Conference of the IEEE, 1998.
- WANG, B., TORO, C., ZEFFIRO, T. A. & HALLETT, M. 1994. Head surface digitization and registration: a method for mapping positions on the head onto magnetic resonance images. *Brain topography*, 6, 185-192.
- WANG, C. C. 1992. Extrinsic calibration of a vision sensor mounted on a robot. *IEEE transactions on Robotics and Automation*, 8, 161-175.
- WANG, H., XUE, Y. & WANG, S. Spatio-temporal Dynamics of Brain Potential Induced by Magnetic Stimulation. IFMBE Proceedings, 2007. Springer, 2770.
- WASSERMANN, E. M. 1998. Risk and safety of repetitive transcranial magnetic stimulation: report and suggested guidelines from the International Workshop on the Safety of Repetitive Transcranial Magnetic Stimulation, June 5-7, 1996. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 108, 1-16.
- WASSERMANN, E. M., COHEN, L. G., FLITMAN, S. S., CHEN, R. & HALLETT, M. 1996a. Seizures in healthy people with repeated "safe" trains of transcranial magnetic stimuli. *Lancet(British edition)*, 347, 825-826.
- WASSERMANN, E. M., WANG, B., ZEFFIRO, T. A., SADATO, N., PASCUAL-LEONE, A., TORO, C. & HALLETT, M. 1996b. Locating the motor cortex on the MRI with transcranial magnetic stimulation and PET. *Neuroimage*, 3, 1-9.
- WATT, A. & WATT, M. 1992. *Advanced animation and rendering techniques*, ACM Press, ISBN 0-201-54412-1.
- WEHAGE, R. A. 1984. Quaternions and Euler parameters. A brief exposition. *Computer Aided Analysis and Optimization of Mechanical Systems Dynamics*, Springer Verlag.
- WELLS, W. M. 1997. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21, 63-98.
- WONGRATANAPHISAN, T. & COLE, M. 2009. Robust impedance control of a flexible structure mounted manipulator performing contact tasks. *Robotics, IEEE Transactions on*, 25, 445-451.
- WU, T., SOMMER, M., TERGAU, F. & PAULUS, W. 2000. Lasting influence of repetitive transcranial magnetic stimulation on intracortical excitability in human subjects. *Neuroscience Letters*, 287, 37-40.
- YOSHIKAWA, T. 1987. Dynamic hybrid position/force control of robot manipulators--description of hand constraints and calculation of joint driving force. *IEEE Journal of Robotics and Automation*, 3, 386-392.
- YOSHIKAWA, T. Force control of robot manipulators. ICRA'00, 2000. Proceedings of IEEE International Conference on Robotics and Automation, 2000. IEEE, 220-226 vol. 1.

- YOSHIKAWA, T., SUGIE, T. & TANAKA, M. 1988. Dynamic hybrid position/force control of robot manipulators-controller design and experiment. *IEEE Journal of Robotics and Automation*, 4, 699-705.
- ZAKARIA, W. W. Expecting submit in 2012. *Phd thesis, Force Control of a Robotic System for Transcranial Magnetic Stimulation*. Newcastle University.
- ZANGEN, A., ROTH, Y., VOLLER, B. & HALLETT, M. 2005. Transcranial magnetic stimulation of deep brain regions: evidence for efficacy of the H-Coil. *Clinical Neurophysiology*, 116, 775-779.
- ZENTNER, J. 1991. Motor evoked potential monitoring during neurosurgical operations on the spinal cord. *Neurosurgical Review*, 14, 29-36.
- ZHAO, Y. & CHEAH, C. C. Hybrid vision-force control for robot with uncertainties. ICRA'04, 2004. Proceedings of 2004 IEEE International Conference on Robotics and Automation, 2004. IEEE, 261-266
- ZHAO, Z. Y., TOMIZUKA, M. & ISAKA, S. 1993. Fuzzy gain scheduling of PID controllers. *Systems, Man and Cybernetics, IEEE Transactions on*, 23, 1392-1398.
- ZIEMANN, U., HALLETT, M. & COHEN, L. G. 1998. Mechanisms of Deafferentation-Induced Plasticity in Human Motor Cortex. *Journal of Neuroscience*, 18, 7000-7007.
- ZIEMANN, U., LOENNECKER, S., STEINHOFF, B. J. & PAULUS, W. 1996. Effects of antiepileptic drugs on motor cortex excitability in humans: A transcranial magnetic stimulation study. *Annals of Neurology*, 40, 367-378.

Appendix 1

This section gives some features of the Polaris Spectra system and the tools. The information is adapted from the Passive Polaris Spectra User Guide, Polaris Tool Design Guide and Polaris Spectra Tool Kit Guide. Since these features are important during the developing process of the system, the information is included here for completeness.

A 1.1 Determining the segment angle

There are two methods for determining the segment angle. The one is to use the 3D coordinates of the markers, the other is to use the segment lengths.

Table A 1.1 determining the segment angle using segment lengths

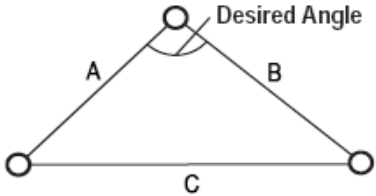
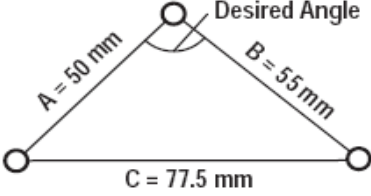
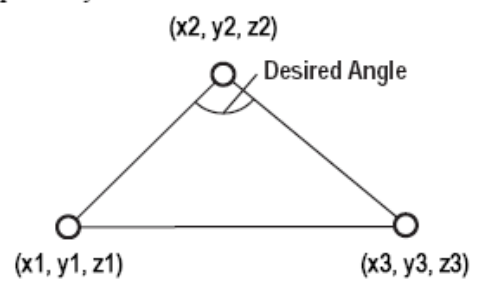
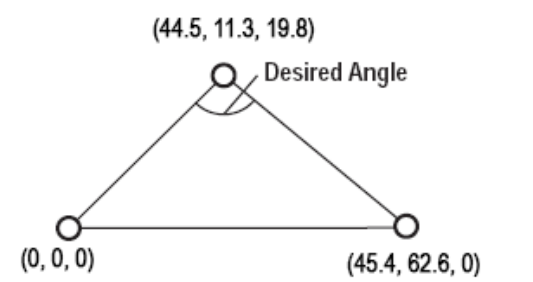
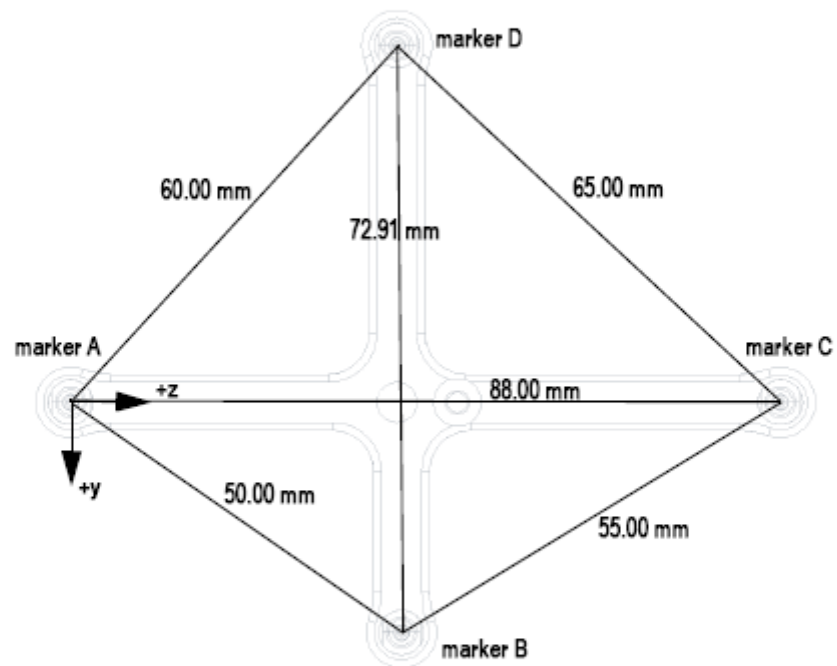
To Determine the Segment Angle:	Example:
1. The segment angles are given by A, B and C. Segments A and B meet at the desired segment angle.	The segment lengths are as shown:
	
2. Use the law of cosines to determine the segment angle. $\text{Segment angle} = \text{acos}\left(\frac{A^2 + B^2 - C^2}{2 \cdot A \cdot B}\right)$	$\text{Segment angle} = \text{acos}\left(\frac{50^2 + 55^2 - 77.5^2}{2 \cdot 50 \cdot 55}\right)$ $\text{Segment angle} = 95.02^\circ$

Table A 1.2 determining the segment angle using 3D coordinates

To Determine the Segment Angle:	Example:
1. The 3D marker coordinates are given by (x_1, y_1, z_1) , (x_2, y_2, z_2) , and (x_3, y_3, z_3) for markers 1, 2 and 3, respectively.	The 3D coordinates are as shown:
	
2. Use the 3D coordinates of the three markers that define the angle to determine the dot product. Dot product = $(x_1 - x_2) \cdot (x_3 - x_2)$ $+ (y_1 - y_2) \cdot (y_3 - y_2)$ $+ (z_1 - z_2) \cdot (z_3 - z_2)$	Dot product = $(0 - 44.5) \cdot (45.4 - 44.5)$ $+ (0 - 11.3) \cdot (62.6 - 11.3)$ $+ (0 - 19.8) \cdot (0 - 0)$ Dot product = -239.49
3. Determine the maximum dot product magnitude. Maximum dot product magnitude = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$ $\cdot \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}$	Maximum dot product magnitude = $\sqrt{(0 - 44.5)^2 + (0 - 11.3)^2 + (0 - 19.8)^2}$ $\cdot \sqrt{(45.4 - 44.5)^2 + (62.6 - 11.3)^2 + (0 - 19.8)^2}$ Maximum dot product magnitude = 2737.94
4. Use the dot product and the maximum dot product magnitude to determine the normalized dot product. Normalized Dot Product = $\frac{\text{Dot product}}{\text{Maximum dot product magnitude}}$	Normalized dot product = $\frac{-239.49}{2737.94}$ Normalized dot product = -0.087
5. Use the normalized dot product to calculate the segment angle. Segment angle = $\text{acos}(\text{normalized dot product})$	Segment angle = $\text{acos}(-0.087)$ Segment angle = 95.02°

A 1.2 Tools specifications

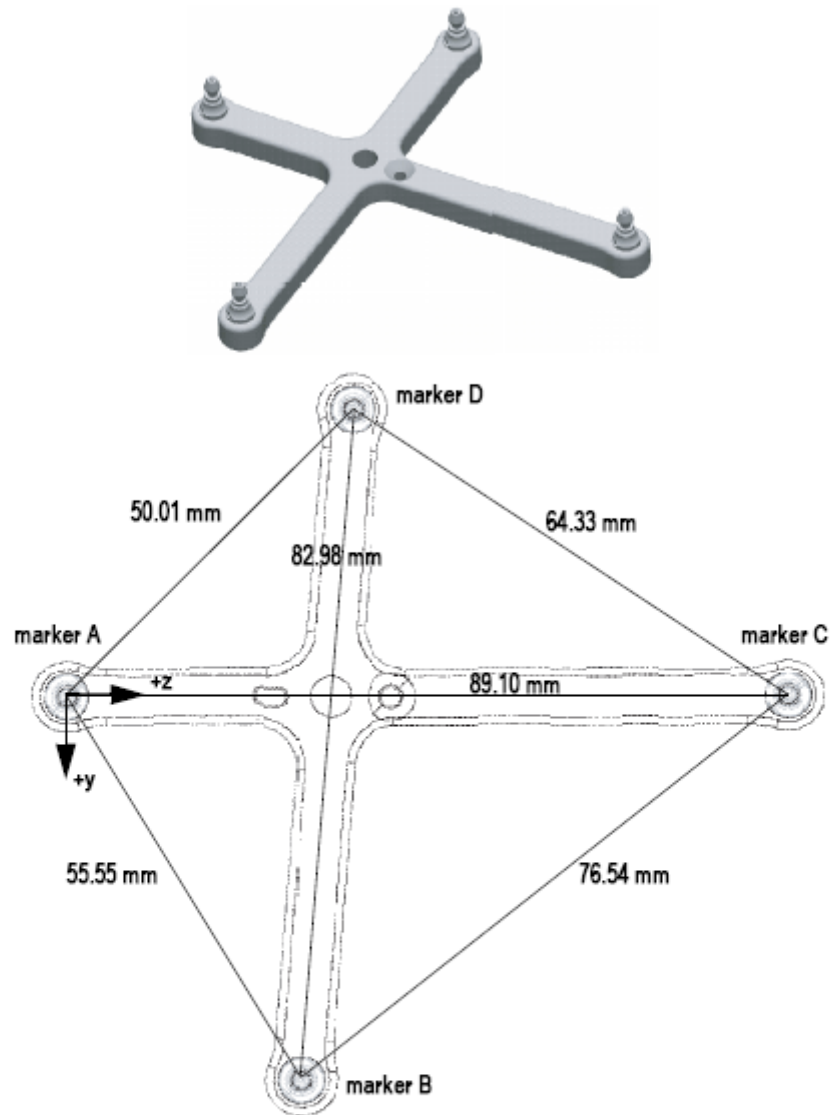
A 1.2.1 Tool 1



Specifications

Geometry	x (mm)	y (mm)	z (mm)
Marker A	0.00	0.00	0.00
Marker B	0.00	28.59	41.02
Marker C	0.00	0.00	88.00
Marker D	0.00	-44.32	40.45

A 1.2.2 Tool 2



Specifications

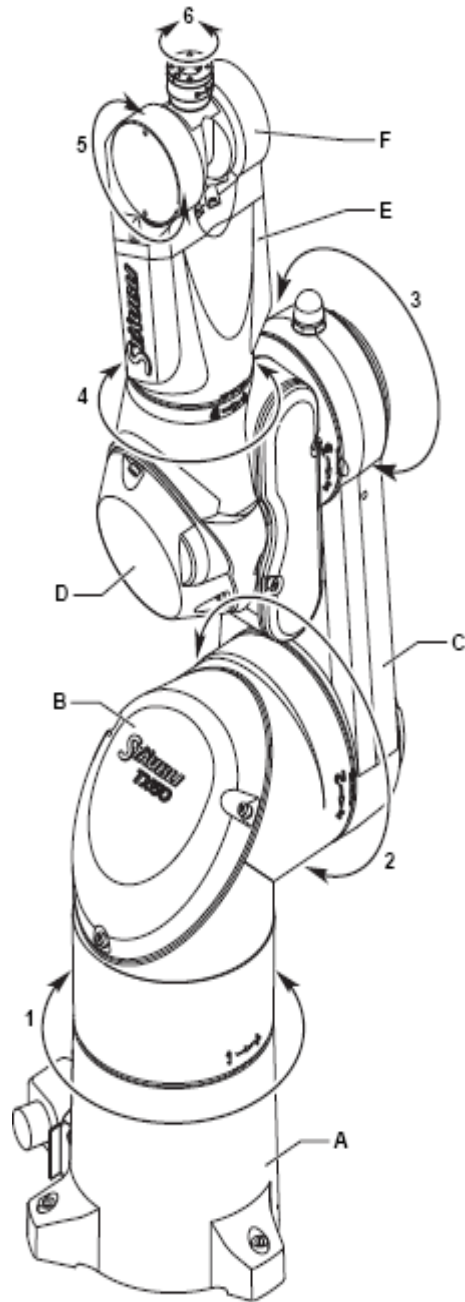
Geometry	x (mm)	y (mm)	z (mm)
Marker A	0.00	0.00	0.00
Marker B	0.00	47.38	28.99
Marker C	0.00	0.00	89.10
Marker D	0.00	-35.36	35.36

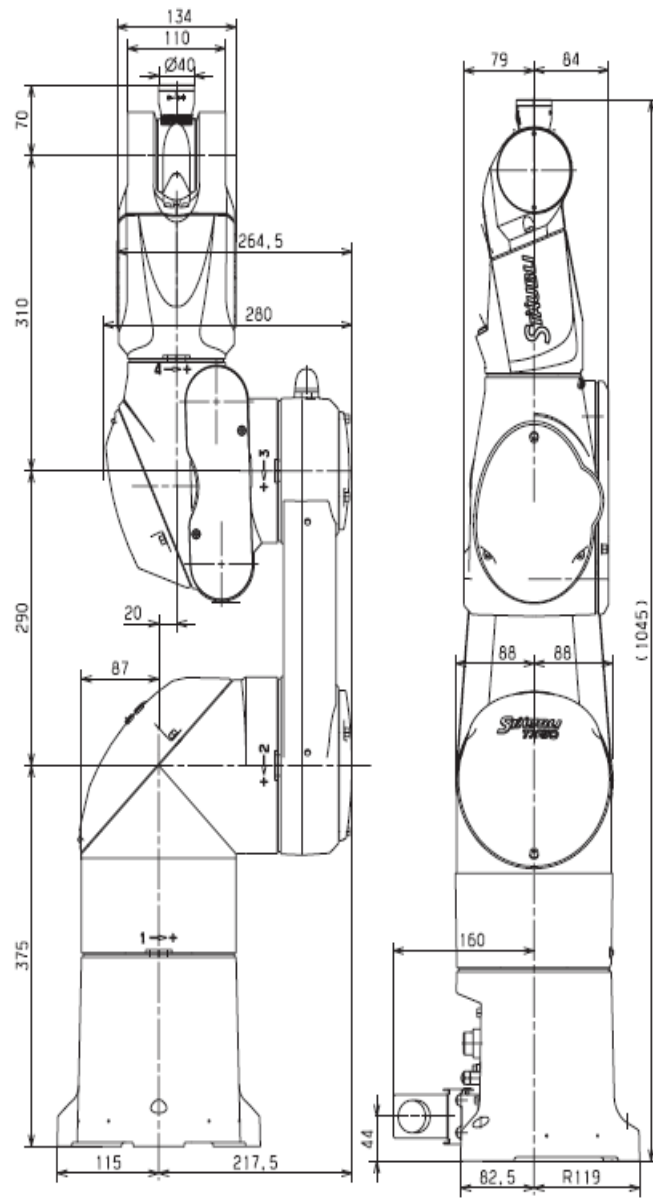
A 1.3 Robot characteristics

A 1.3.1 Robot specification

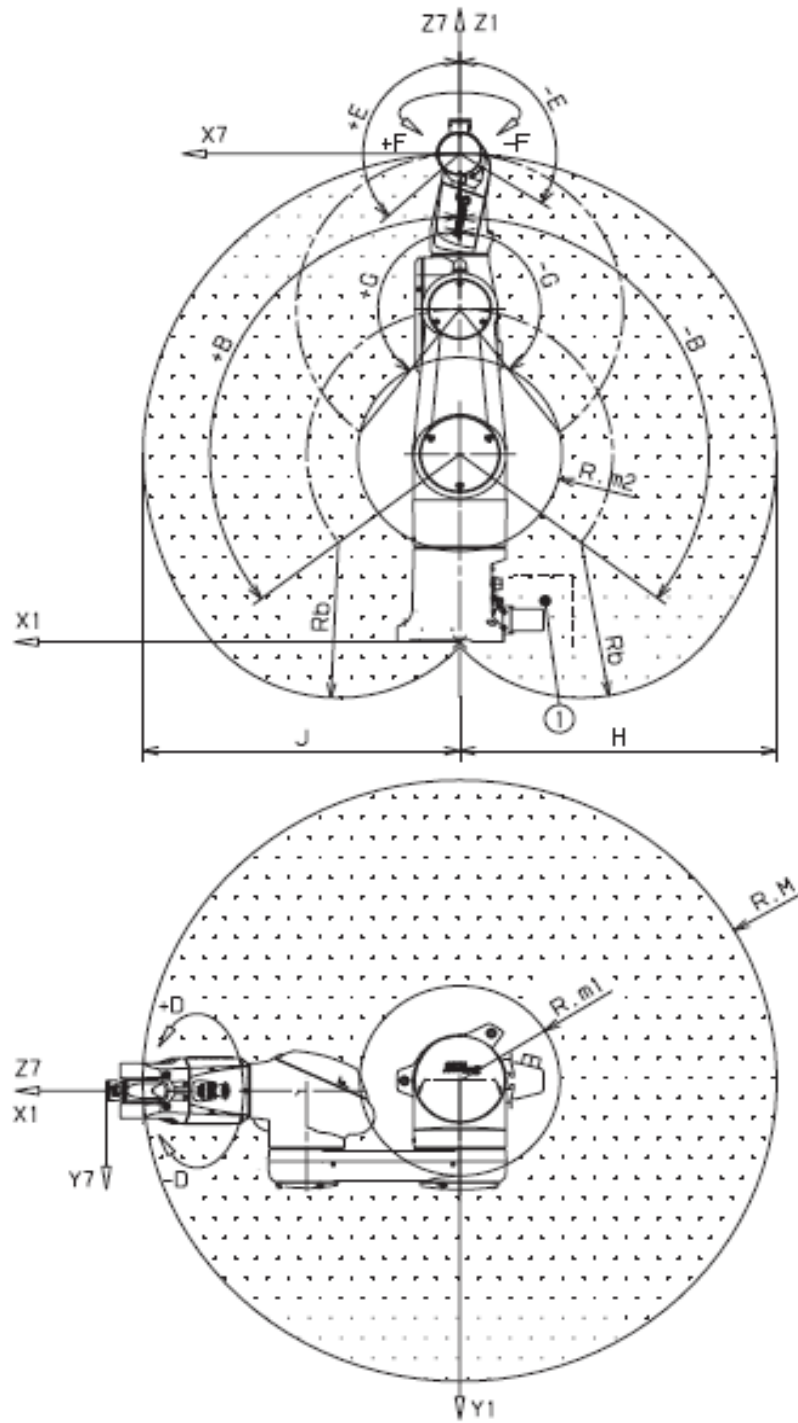
MODEL		TX60
Characteristics		
Maximum load		9 kg
Nominal load		3.5 kg
Reach (between axis 1 and 6)		670 mm
Number of degrees of freedom		6
Repeatability - ISO 9283		$\pm 0,02$ mm
Motion range	Axis 1 (A)	$\pm 180^\circ$
	Axis 2 (B)	$\pm 127,5^\circ$
	Axis 3(C)	$\pm 142,5^\circ$
	Axis 4 (D)	$\pm 270^\circ$
	Axis 5 (E)	$+133.5\% - 122.5^\circ$
	Axis 6 (F)	$\pm 270^\circ$
Work envelope	Maximum reach between axis 1 and 5 (R. M)	600 mm
	Minimum reach between axis 1 and 5 (R. m1)	190 mm
	Minimum reach between axis 2 and 5 (R. m2)	189 mm
	Minimum reach between axis 3 and 5 (R. b)	310 mm
Maximum speed	Axis 1	435 %s
	Axis 2	410 %s
	Axis 3	540 %s
	Axis 4	995 %s
	Axis 5	1065 %s
	Axis 6	1445 %s
Maximum speed at load gravity center		8 m/s
Maximum inertias	Axis 5	0.325 kg.m^2
	Axis 6	0.1 kg.m^2
Weight		51.4 kg
Brakes		All axis

A 1.3.2 Motion range and demesions (Unit : mm)

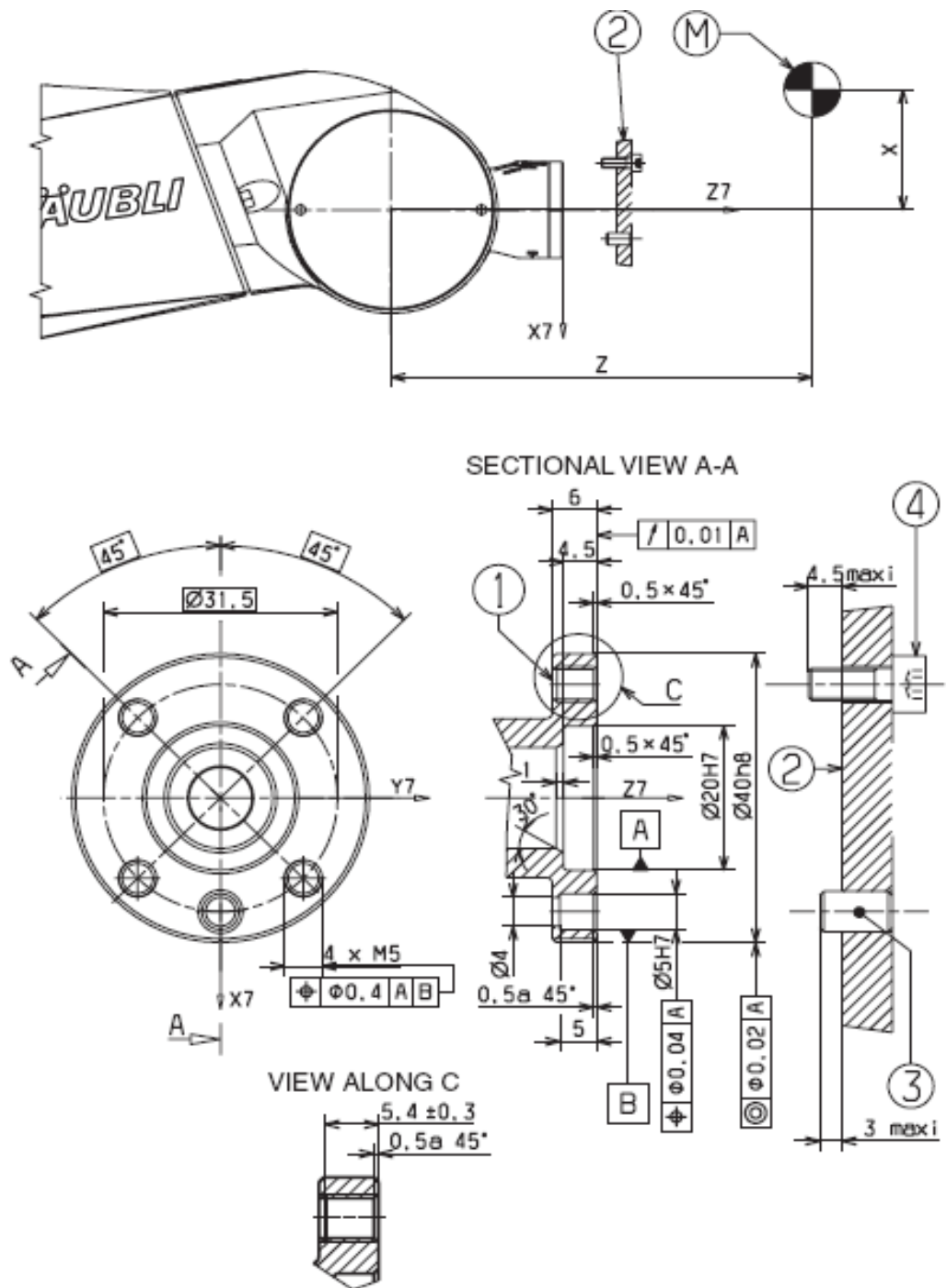




A 1.3.3 Work envelope (Unit : mm)

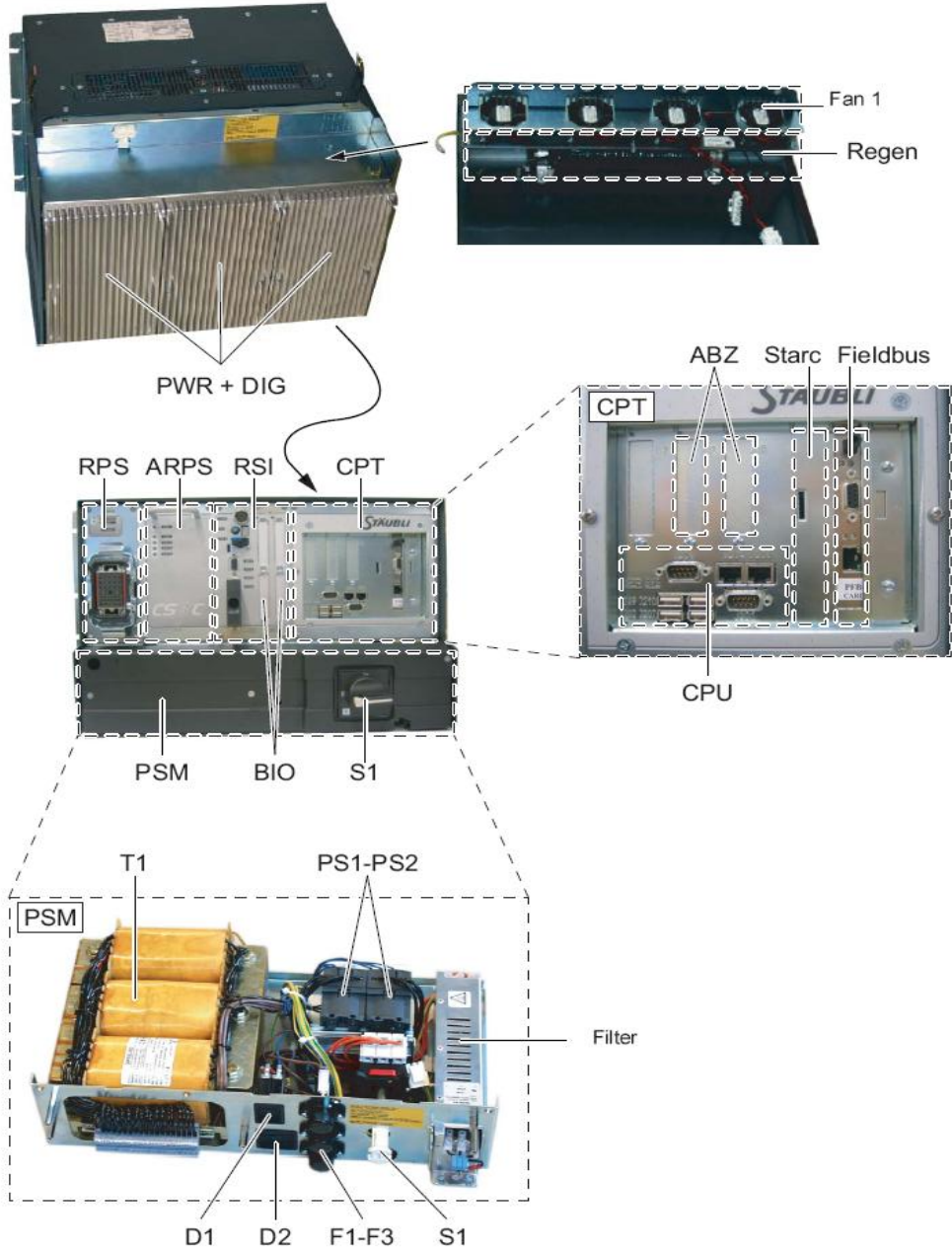


A 1.3.4 Twist (Unit : mm)



- ① Mechanical interface
- ② End-effector

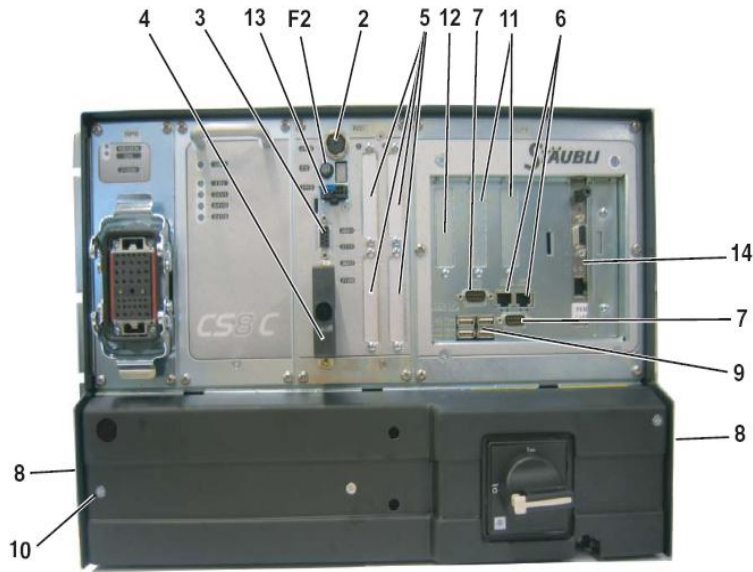
A 1.3.5 Components locations



ABZ	ABZ Encoder board	ABZ Encoder board
ARPS	Auxiliary Robot Power Supply (logical voltages)	Auxiliary Robot Power Supply (logical voltages)
BIO	Basic Inputs Outputs	Basic Inputs Outputs
BRB	Brake Release Board	Brake Release Board
BRK	Brake	Brake
COD	Arm Encoder	Arm Encoder
CPT	Computer	Computer
DIG	Digital part of the amplifier	Digital part of the amplifier
DOOR	Door contact	Door contact
DSI	Dual Sensor Interface board in Arm	Dual Sensor Interface board in Arm
EV	Solenoid valve	Solenoid valve
FAN	Fan	Fan
Fieldbus	Fieldbus	Fieldbus
IC	Interconnect Cable	Interconnect Cable
LSW	Limit Switch	Limit Switch
MCP	Manual Control Pendant	Manual Control Pendant
MCPES	Manual Control Pendant Emergency Stop	Manual Control Pendant Emergency Stop
MOT	Motor	Motor
PSM	Power Supply Module (power voltage)	Power Supply Module (power voltage)
PWR	Power part of the amplifier	Power part of the amplifier
REGEN	Regeneration resistance	Regeneration resistance
RPS	Robot Power Supply	Robot Power Supply
RSI	Robot Safety Interface	Robot Safety Interface
S1	Main switch	Main switch
STARC	Stäubli Advanced Robot Control	Stäubli Advanced Robot Control
Th	Thermo Sensor	Thermo Sensor
UESA	User Emergency Stop	User Emergency Stop
UESB	User Safety Stop	User Safety Stop
USEREN	User validation	User validation
WMS	Working Modes Selection Control Panel	Working Modes Selection control panel Working Modes Selection front panel
WMSES	Working Modes Selection Emergency Stop	Working Modes Selection Emergency Stop

A 1.3.6 Signal connections

The input/output signals are connected via the connectors located on the front panel of the controller.



- 2: Connector for MCP
- 3: Fast Inputs/Outputs
- 4: Connection with cell (emergency stop, door etc.)
- 5: Digital I/O options
- 6: Ethernet links
- 7: Serial links
- 9: USB links
- 10: Anti-static wrist strap
- 11: Optional encoder input
- 12: CAN output for Scara robot
- 13: Connector for WMS front panel
- 14: Optional fieldbus board

A 1.3.7 MCP and WMS

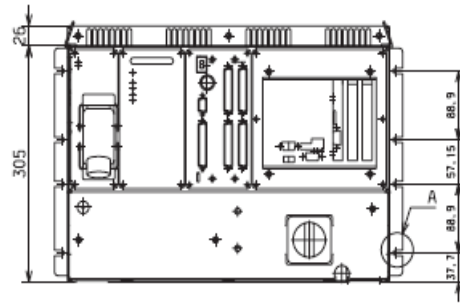
MCP (manual control pendent) can be used to enable power supply and control its movements.



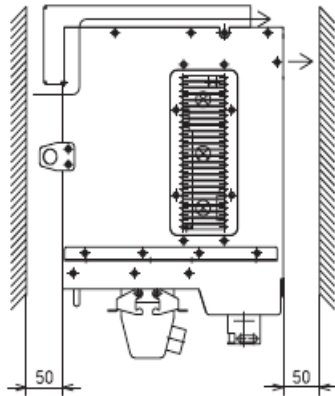
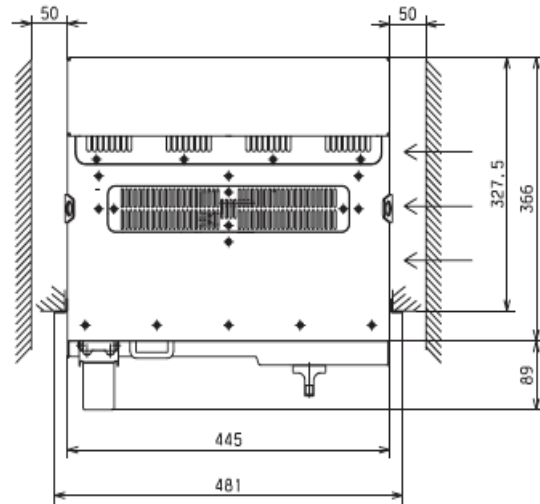
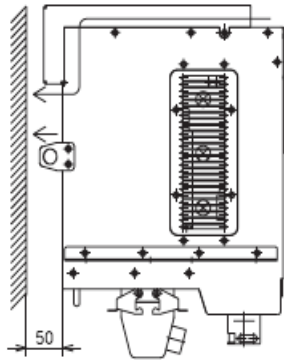
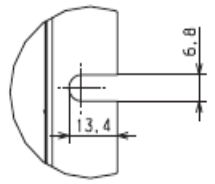
WMS (working mode selection mode) is used for select the working mode of the robot. There are three working modes: manual mode, local mode and remote mode.



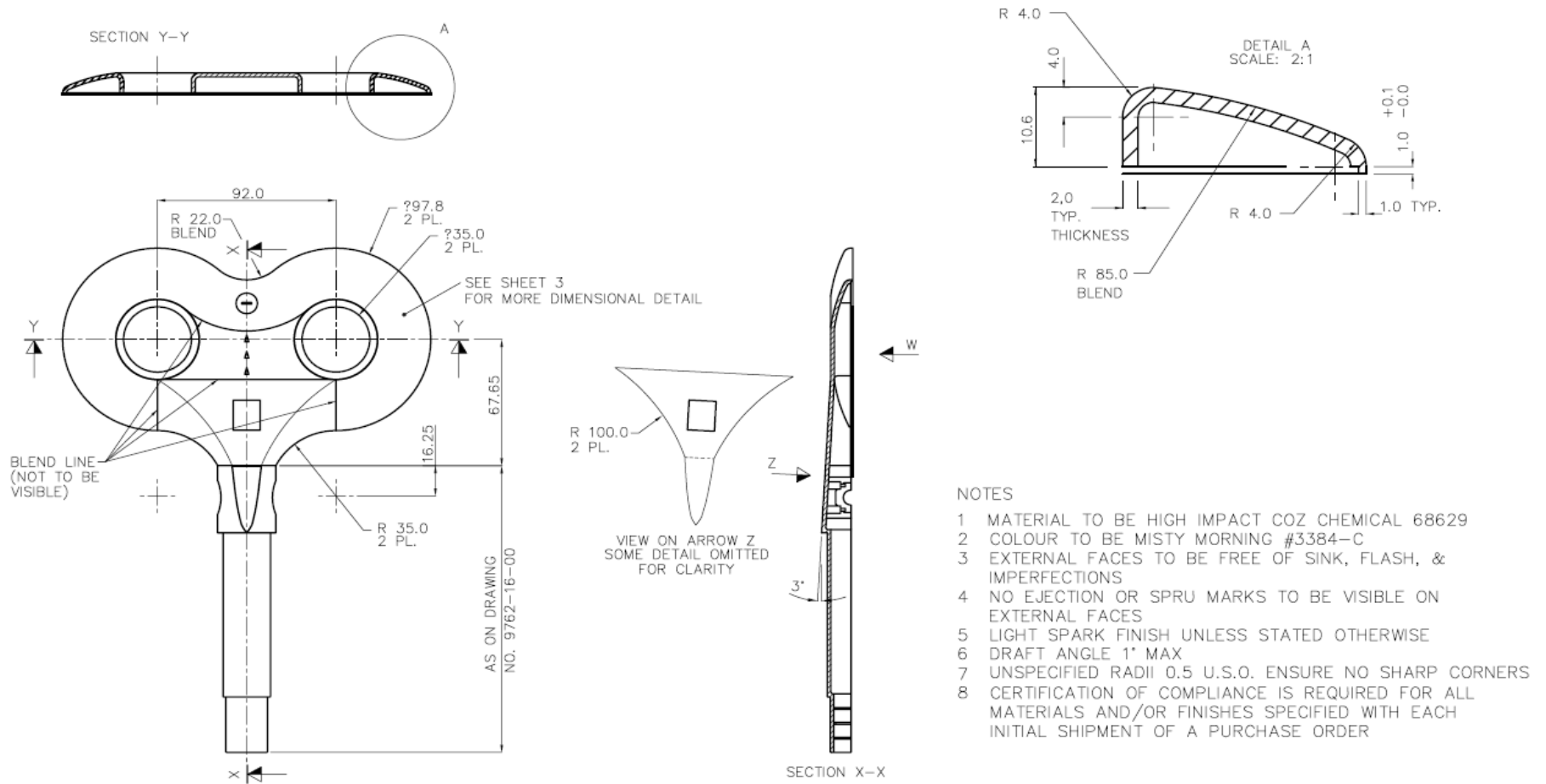
A 1.3.8 Three demension of the robot controller



Detail: A
Scale: 1:1



A 1.4 Geometrical information of the coil



A 1.5 Quaternions

This section gives some properties and operation of quaternions. The information is adapted from <http://en.wikipedia.org/wiki/Quaternion> and [Conway et al., 2004]. Since these features are essential during the developing process of the system, the information is included here for completeness.

Irish mathematician Sir William Rowan Hamilton initially induced quaternions in 1843 and applied to mechanics in three-dimensional space. They can be used in both theoretical and applied mathematics, especially for calculating three-dimensional rotations. The algebra of quaternions is often denoted by \mathbf{H} (for *Hamilton*).

A quaternion q can be represented by the following:

$$q = a_0 + a_1\vec{i} + a_2\vec{j} + a_3\vec{k}$$

where a_0, a_1, a_2, a_3 are real numbers, $a_1\vec{i} + a_2\vec{j} + a_3\vec{k}$ is the imaginary part which can represent a vector in 3D space. It is also write as: $q = (a_0, (a_1, a_2, a_3))$

The properties of the imaginary components are as followed:

$$\vec{i}^2 = \vec{j}^2 = \vec{k}^2 = -1$$

$$\vec{i} * \vec{j} = \vec{k}, \vec{j} * \vec{k} = \vec{i}, \vec{k} * \vec{i} = \vec{j}$$

$$\vec{j} * \vec{i} = -\vec{k}, \vec{k} * \vec{j} = -\vec{i}, \vec{i} * \vec{k} = -\vec{j}$$

If there is a $q = b_0 + b_1\vec{i} + b_2\vec{j} + b_3\vec{k}$, the multiplication of two quaternions is as:

$$q * p = (b_0a_0 - b_1a_1 - b_2a_2 - b_3a_3) + \vec{i}(b_0a_1 + b_1a_0 + b_2a_3 - b_3a_2) \\ + \vec{j}(b_0a_2 - b_1a_3 + b_2a_0 + b_3a_1) + \vec{k}(b_0a_3 + b_1a_2 - b_2a_1 + b_3a_0)$$

And the multiplication operation is not commutative.

A conjugate quaternion of q is given as followed:

$$q^* = a_0 - a_1\bar{i} - a_2\bar{j} - a_3\bar{k}$$

Which has the same real part but negated imaginary part.

The norm of a quaternion is defined like:

$$\|q\| = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_3^2}$$

And a multiplicative inverse of a non-zero quaternion is

$$q^{-1} = \frac{q^*}{\|q\|} = \frac{q^*}{qq^*}$$

Therefore, if the quaternion is a unit quaternion whose norm $\|q\|=1$, the inverse of the quaternion equals to its conjugate $q^{-1} = q^*$.

As we know, a rotation in two dimensions can be defined with the multiplication of two unit complex numbers. Similarity, a multiplication of two unit quaternion can be used to model a rotation in three dimensions. Since a point $t = (t_x, t_y, t_z)$ in 3D space can be represented by a purely imaginary quaternion, the rotation of this point about a rotation axis r with angle θ can be defined as multiplication of the quaternions.

If define two quaternions:

$$q = \left(\cos \frac{\theta}{2}, r * \sin \frac{\theta}{2}\right)$$

$$q_t = (0, t)$$

Hence

$$q_t' = qq_t q^{-1} = qq_t q^*$$

Where q_t' is the point after rotation, and q is a unit quaternion.

The matrix notation for last equation is

$$\begin{bmatrix} t_x' \\ t_y' \\ t_z' \end{bmatrix} = \begin{bmatrix} a_0^2 + a_1^2 - a_2^2 - a_3^2 & -2a_0a_3 + 2a_1a_2 & 2a_0a_2 + 2a_1a_3 \\ 2a_0a_3 + 2a_1a_2 & a_0^2 - a_1^2 + a_2^2 - a_3^2 & -2a_0a_1 + 2a_2a_3 \\ -2a_0a_2 + 2a_1a_3 & 2a_0a_1 + 2a_2a_3 & a_0^2 - a_1^2 - a_2^2 + a_3^2 \end{bmatrix} * \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

This also shows the conversion from a quaternion to a rotation matrix.

The following equations for converting a rotation matrix to a unit quaternion:

If a homogenous matrix $R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, we can obtain

$$a_0 = \pm \frac{1}{2} \sqrt{R_{11} + R_{22} + R_{33} + 1}$$

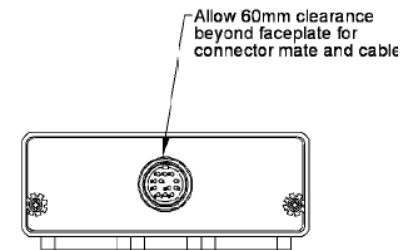
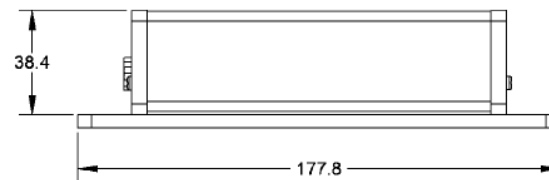
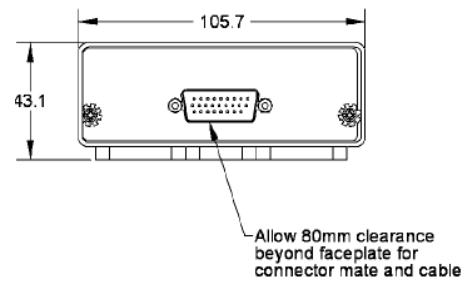
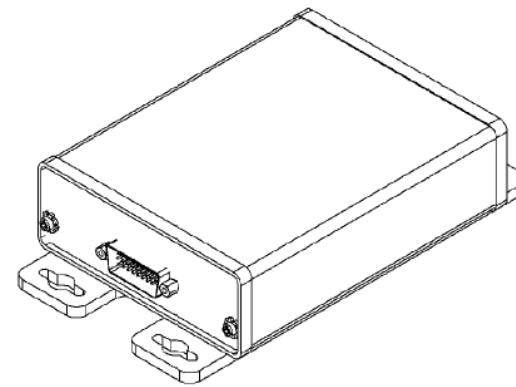
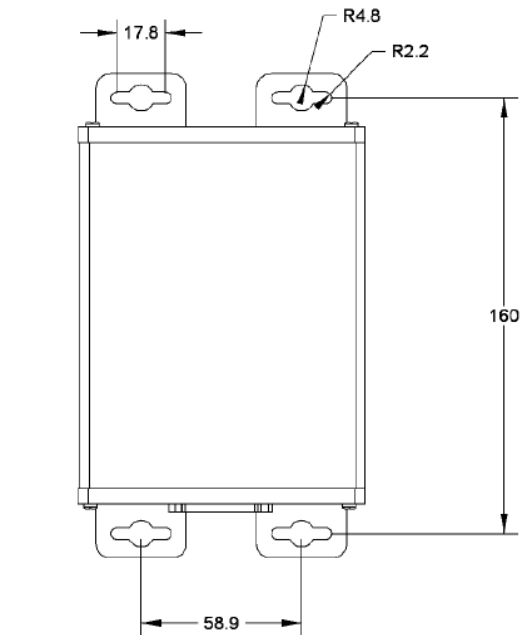
$$a_1 = \frac{R_{32} - R_{23}}{4a_0}$$

$$a_2 = \frac{R_{13} - R_{31}}{4a_0}$$

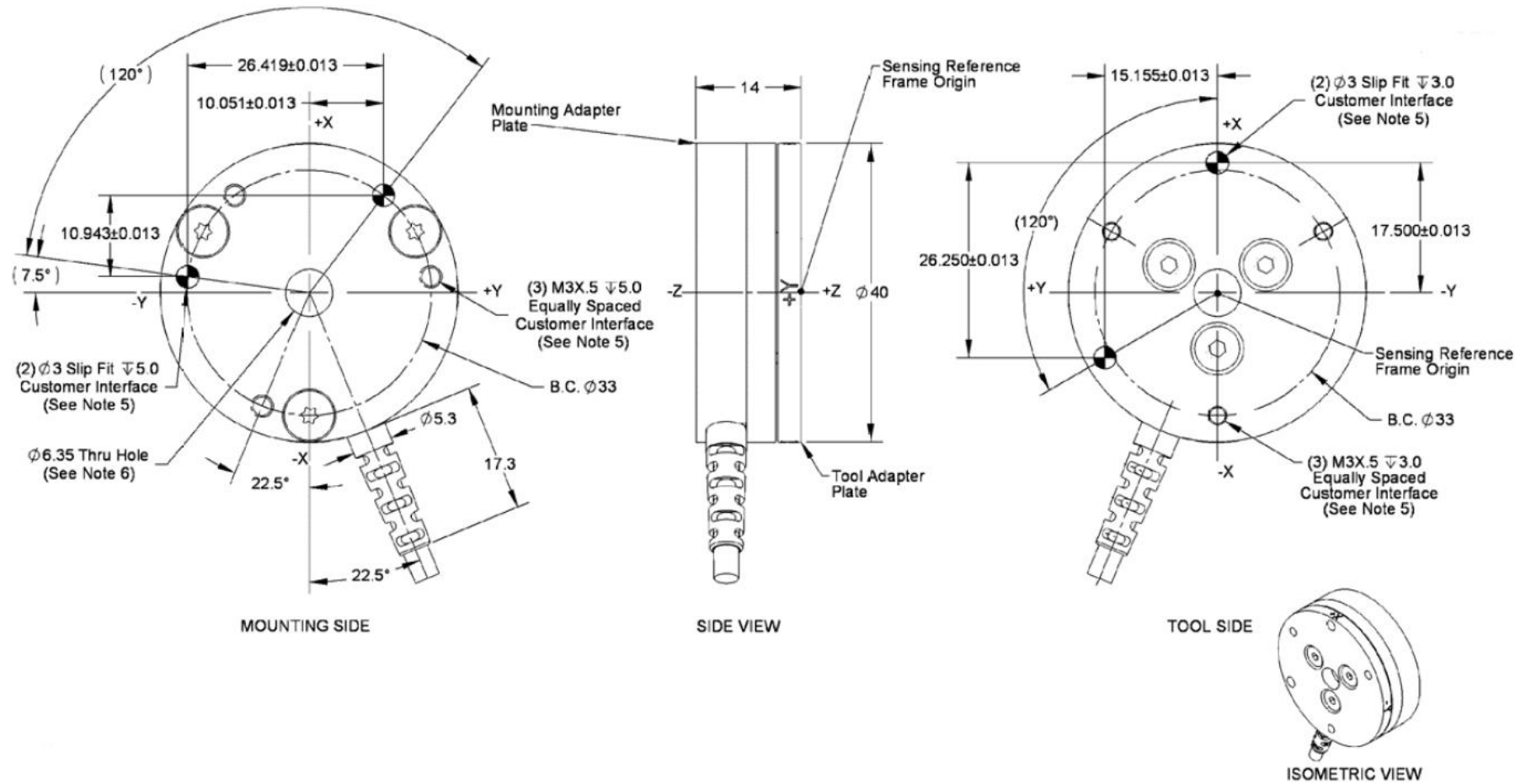
$$a_3 = \frac{R_{21} - R_{12}}{4a_0}$$

A 1.6 Dimensions of components of force sensor system

A 1.6.1 Dimension of DAQ power supply



A 1.6.2 Dimension of the transducer



A 1.6.3 Properties of the transducer

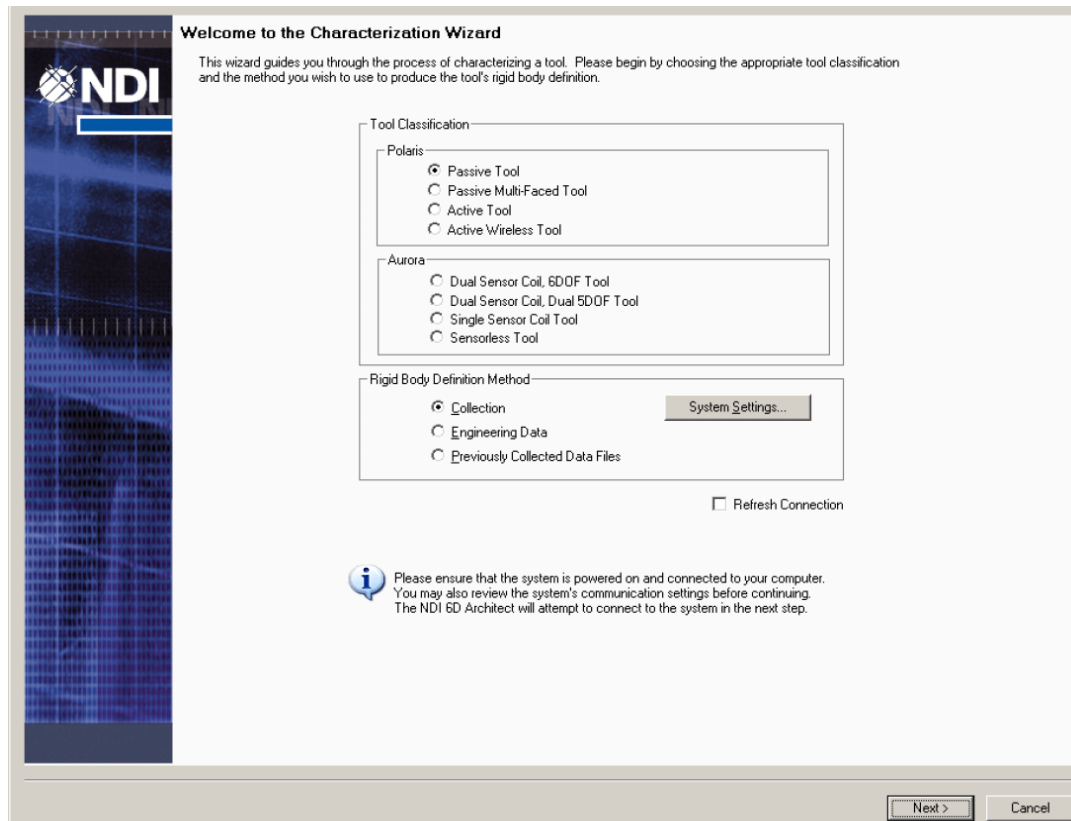
SI (Metric)						
Calibration	F _x ,F _y	F _z †	T _x ,T _y , T _z	F _x ,F _y	F _z †	T _x ,T _y , T _z
SI-20-1	±20 N	±60 N	±1 Nm	2 N/V	6 N/V	0.1 Nm/V
SI-40-2	±40 N	±120 N	±2 Nm	4 N/V	12 N/V	0.2 Nm/V
SI-80-4	±80 N	±240 N	±4 Nm	8 N/V	24 N/V	0.4 Nm/V
			Analog Output Range		Analog ±10V Sensitivity‡	

A 1.7 Method of creating a tool definition file for the passive tools

The tool definition file of a passive tool is created and characterized using NDI 6D Architect software. This appendix introduces the steps to make a tool definition file.

Step 1: select tool type and method

When the characterization wizard started, the following dialog appears. This first screen acts as a portal to the rest of the wizard's functionality. Tasks and procedures that the wizard will guide through are the customized according to the kind of tool that are characterizing.



1. In the Tool Classification section, select the type of Passive tool.
2. In the rigid body definition method section, select the type of Engineering Data.
3. Click next.

Step 2: Enter tool parameters

Once the type of tool have been chosen, the characterization wizard will ask to define the basic tool parameters including a part number, a manufacturer name, and serial number setting. These parameters become part of the tool definition file when it is produced at the end of the characterization procedure.

Tool Parameters

Please define the tool properties.

Tool Description

Part Number: Probe1
Manufacturer: NDI Revision: 1

Tool Type: Probe
Tool Subtype: Removable Tip
Marker Type: NDI930 Active Marker - ceramic

Serial Number Settings

Date (mm/dd/yyyy): 4/20/2007
Sequence Number: 0

Enhanced Tracking Algorithm

Lock on using only 3 markers
 Unique geometry tracking

Tool Tracking Parameters

Maximum Marker Angle: 60 degrees
Minimum Number of Markers: 3
Maximum 3D Error: 2 mm

Store to Defaults

< Back Next > Cancel

1. Enter the user-specified Part Number for the tool.
2. Enter the Manufacturer Name for the tool.
3. Enter the Revision Number for the tool.
4. From the drop-down list provided, select an appropriate type:

Tool Type	Description
Reference	A tool whose local coordinate system is used as a frame of reference in which other tools are tracked
Probe	A commonly used tool with a tip, meant to be used as a pointer
Button Box/Foot Switch	A markerless tool meant to be used as a switch or a foot pedal. This option is not available for passive tools.
Software Defined	Select this option if you want your application software to identify and acknowledge the tool type instead of storing this information on the tool's SROM device
Microscope Tracker	A special type of reference tool, usually attached to a microscope
Calibration Device	A special type of reference tool used to verify the calibration of an instrument's tip location, and possibly the instrument's diameter and vector
Isolation Box	A special type of tool that maintains electrical isolation between the tool and the Tool Interface Unit
C-Arm Tracker	A special type of reference tool generally attached to fluoroscope devices
GPIO Device	A device that incorporates four configurable input/output lines.

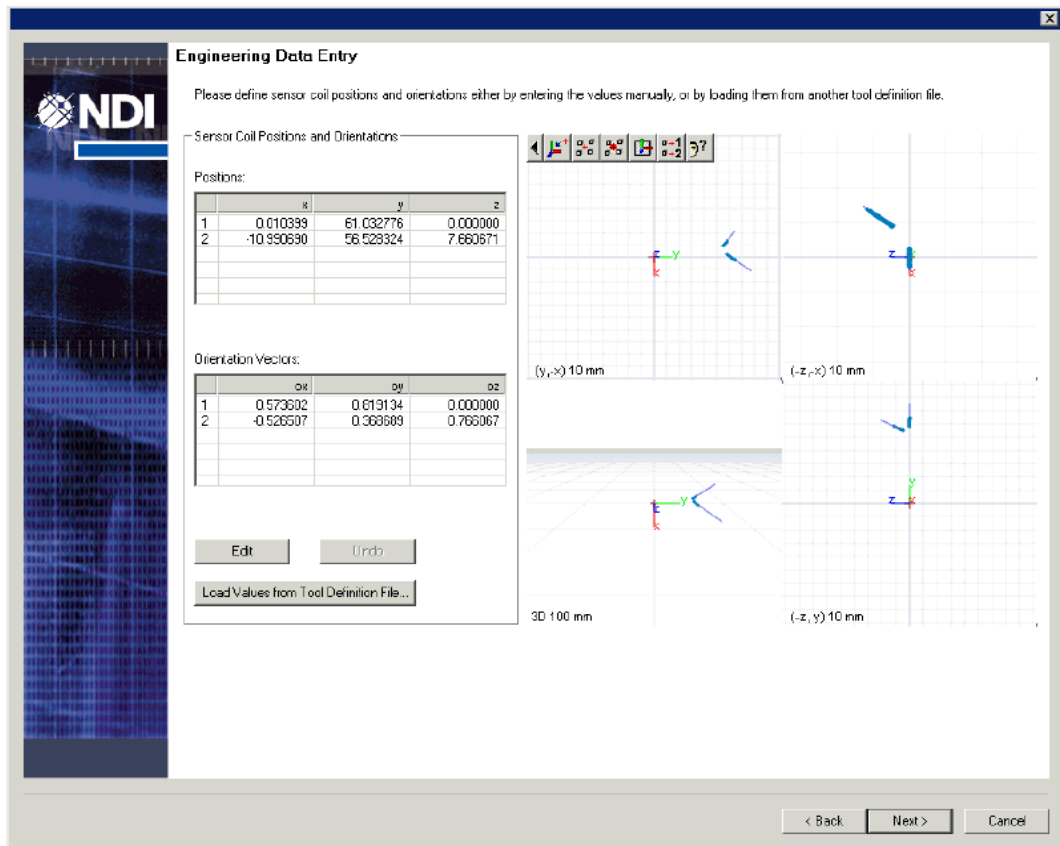
5. From the drop-down list provided, select the appropriate Tool Subtype.
This selection is made according to the kind of tip the tool has: Undefined, Fixed Tip, or Removable Tip.
6. Select the Marker Type used in the design.
7. Set tracking constrains for the tool using Tolerance Parameter below:

Setting	Description
Maximum Marker Angle	The maximum angle that the marker can be facing away from the Position Sensor. This field will accept an integer between 0 and 90. The default value for passive markers is 90°; the default value for active markers is 60°. Note that increasing the default setting will affect accuracy.
Minimum Number of Markers	The number of markers per face that must be detected before an NDI measurement system can return a transformation. This field's default value is 3. Note that you can also enter 0 for a markerless tool.
Maximum 3D Error	The maximum difference (in millimetres) between the marker's expected and measured positions. If the 3D error for a marker exceeds this value, its measurement data will not be used to calculate the tool's transformation. This field will accept any floating point value between 0.05 and 10, with a default of 0.5.

8. Click Next.

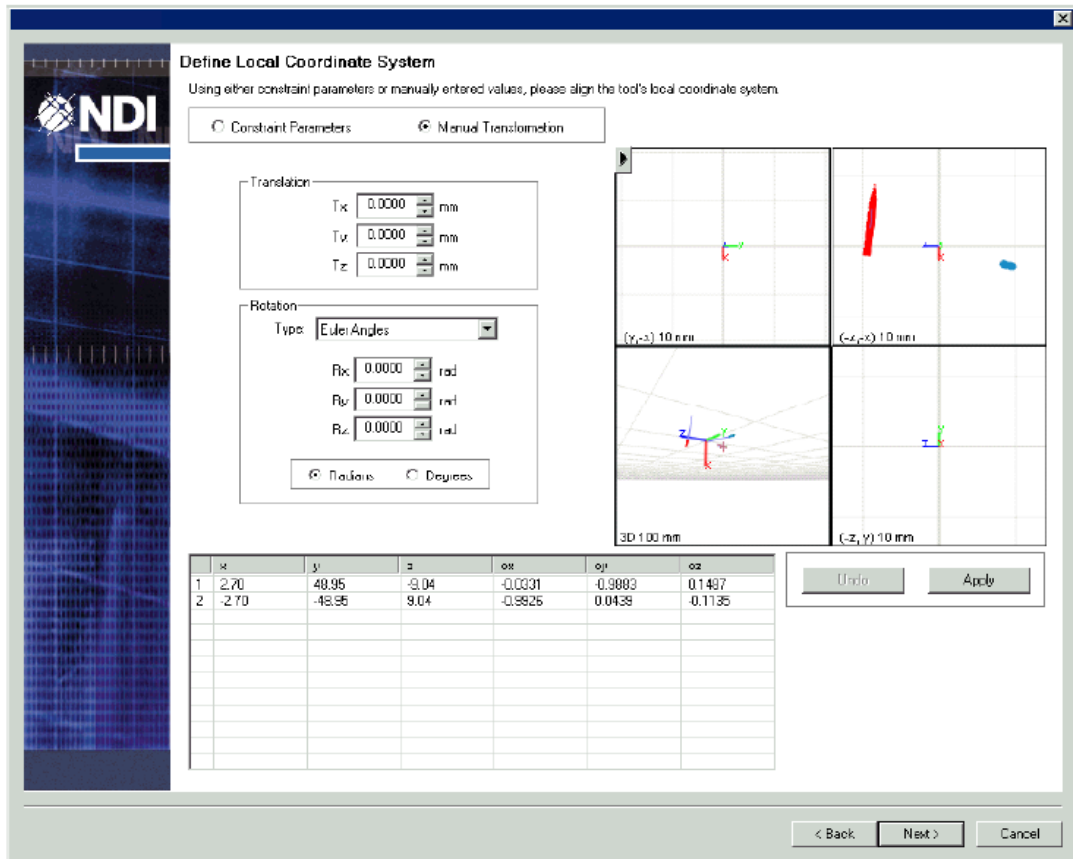
Step 3: Produce a rigid body definition with engineer data

This step is to create the rigid body definition according to the dimensions of the rigid body, and they can be input to the wizard directly, and click next.



Step 4: Align the local coordinate system of the tool

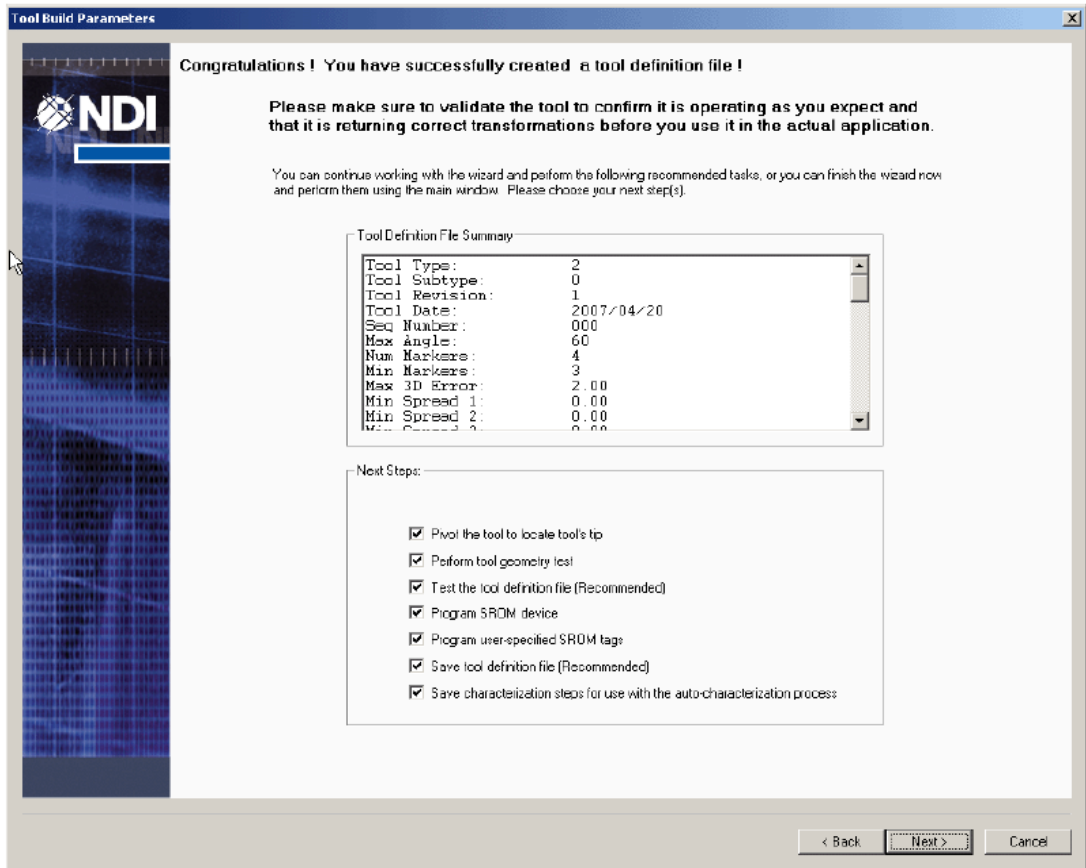
This step in the characterization process is to ensure the tool's local coordinate system is correctly aligned. The coordinate values can be entered manually, and the wizard also provides a graphic representation of each axis and origin value entered to help to review the adjustment.



1. Click Manual Transformation.
2. In the translation section, use the coordinate fields provided to translate the origin of the local coordinate system.
3. In the Rotation section, use the fields provided to rotate the local coordinate system.
4. Click apply once it finished and click next.

Step 5: Complete the characterization

There are several selections for final tasks before the characterization process complete. If nothing is selected, the wizard closes and the software populates the main window with the tool definition file.



Appendix 2

This section describes some features of programming using both QNX and VAL3. Since these features are important during the developing process of the system, the information is included here for completeness.

A 2.1 System architecture of QNX

The architecture of the system consists of a microkernel and optional cooperating processes as shown in Figure A2.1. It acts as a kind of "software bus" that the optional cooperating processes can dynamically plugin/plugin OS modules whenever they are needed, which ensures the small size of the OS. The microkernel implements only the key services such as threads, signals, message passing, synchronization, scheduling and timer services; additional functionality is implemented within these cooperative processes. Compared with other OS, QNX Neutrino can achieve a unique degree of efficiency, modularity and simplicity, which relies on two fundamental features: microkernel architecture and message based inter-process communication (IPC). The microkernel and IPC adopted in this application are briefly described in the following sections.

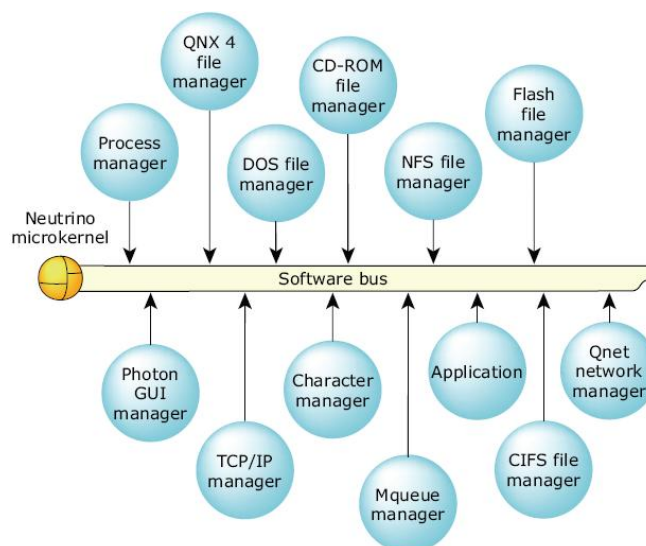


Figure A2.1 QNX Neutrino Architecture[QNX, 2008].

A 2.1.1 QNX Neutrino Microkernel

The *kernel* is the heart of any operating system. In some systems, e.g. Windows, the *kernel* comprises so many functions that for all intents and purposes, in this way it becomes the entire operating system. The QNX Neutrino microkernel implements the core POSIX features used in embedded real-time systems, along with the fundamental QNX Neutrino message-passing services. It is a tiny *kernel*, and there are only several objects contained within it: thread, dispatch, timer, pulse, vector, channel and connection (Figure A2.2). Channel and connection objects are two special objects adopted by QNX, such that message passing is not directed from thread to thread but directed towards channels and connections. A thread that wishes to receive messages first creates a channel; another thread that wishes to send a message to that thread must first make a connection by attaching to that channel. In this way, the message passing is more efficient since the connection and channel have been determined and there is no need to decide where to send the message again.

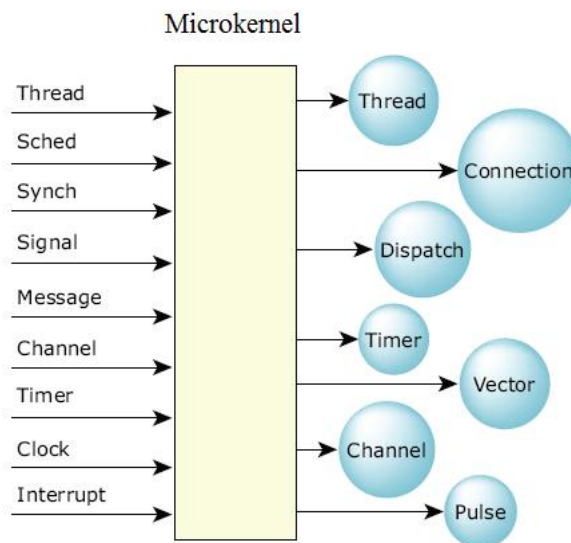


Figure A2.2 QNX Neutrino Microkernel[QNX, 2008].

The QNX *kernel* (unlike the *kernels* of other OS which comprise numerous functions) lacks a file-systems and several other services that are normally expected of an OS. QNX only support a few key services. These high-level service routines operated at a

higher level and manipulate the contained objects which run at lowest level of the microkernel, as follows:

- Thread service for threads management and creation.
- Signal service for software and hardware signals management.
- Synchronization service for synchronizing relative threads.
- Message passing service. The microkernel handles all messages between all threads throughout the entire system.
- Scheduling service which is used to schedule the threads for execution.
- Timer service for providing various timer services.
- Process management service.

A 2.1.2 Inter-process Communication

Inter-process Communication (IPC) plays a fundamental role in the transformation of QNX Neutrino from an embedded real-time kernel into a full-scale POSIX operating system. IPC is used to connect these optional cooperating processes to the microkernel. Message passing is the primary form of IPC in QNX Neutrino, and is implemented in three function: *MsgSend()*, *MsgReceive()* and *MsgReply()*. It works as a “server-client” connection, in which the state of a client thread in the send-receive-reply transaction works in the sequence illustrated in Figure A2.3.

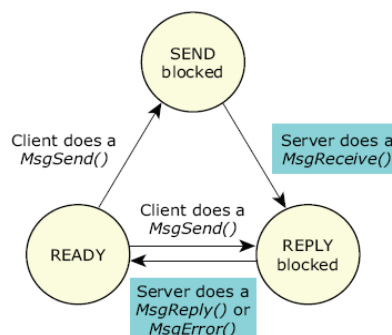


Figure A2.3 State of a client thread in the send-receive-reply transaction[QNX, 2008].

- The thread is in the READY state if it can use the CPU.
- If the client thread calls *MsgSend()*, and the server thread has not called *MsgReceive()*, the client thread becomes SEND blocked. Once the server thread calls *MsgReceive()*, the client thread's state will be changed to REPLY blocked, which means that server thread has receive the message and must reply. When the server calls the *MsgReply()*, the client becomes READY.
- If the client thread calls *MsgSend()*, and the server thread is already blocked on the *MsgReceive()*, then the client thread skip the SEND blocked state to REPLY blocked immediately.
- If the server thread fails, exits, or disappears, the client thread becomes READY with *MsgSend()* indicating an error.

A server thread only has two states (Figure A2.4), and the server is normally waiting to receive a message from a client in a typical server loop.

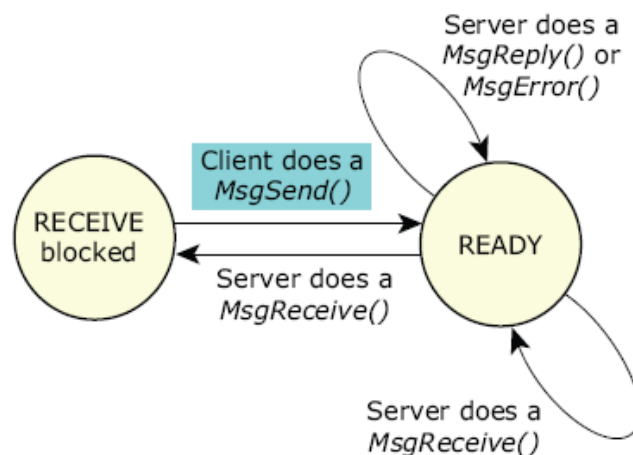


Figure A2.4 State of a server thread in the send-receive-reply transaction[QNX, 2008].

- The thread is in the READY state if it can use the CPU.
- If the server thread calls *MsgReceive()*, and no other thread has been sent to it, then the server thread becomes RECEIVE blocked. When another thread sends to it, the server thread becomes READY.

- If the server thread calls *MsgReceive()*, and another thread has already sent to it, then *MsgReceive()* returns immediately with the message, and the server thread does not block.
- If the server thread calls *MsgReply()*, it does not become blocked.

From the working sequence of the message passing, it can be established that this not only allows threads to pass data to each other, but also provides a method of synchronizing the execution of several threads. Take the client thread for example, after the client thread calls the *MsgSend()*, the client thread will not be executed until it receives a reply from the server. This ensures the server thread can fully execute all the necessary commands before the client thread continues, and the microkernel is able to schedule all threads as efficiently as possible according to their state.

A 2.1.3 Threads scheduling

Since there are several algorithms within the application to execute concurrently, multi-threads are applied to achieve the concurrency. These threads are tightly coupled; the communications and synchronization is achieved using the IPC mechanism mentioned above. Furthermore, every thread is assigned a priority. The thread scheduler selects the next thread to run by looking at the priority assigned to every thread that is READY (capable of using the CPU), and the thread with the highest priority is selected to run. Figure A2.5 shows the ready queue for five threads (B-F) that are READY. Thread A is currently running. All other threads (G-Z) are BLOCKED. Threads A, B, and C are at the highest priority, so they will share the processor based on the running scheduling algorithm of the thread. The OS supports a total of 256 scheduling priority levels, and the threads on the ready queue are ordered by priority. The ready queue is actually implemented as 256 separate queues, one for each priority. Most of the time, threads are queued in a FIFO order in the queue of their priority. The first thread in the highest-priority queue is selected to run.

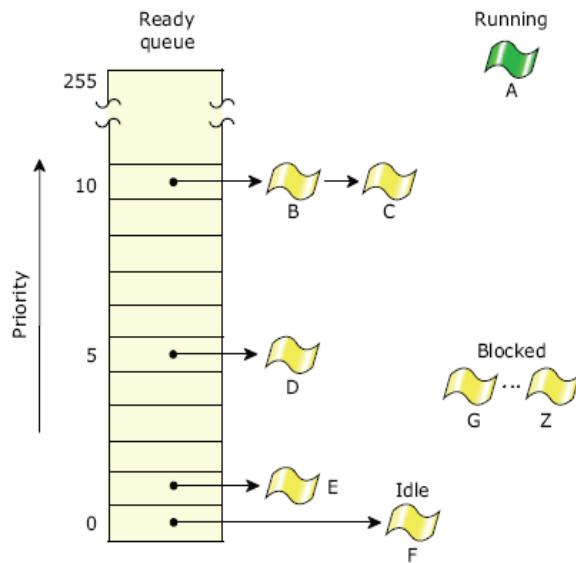


Figure A2.5 The ready queue of threads[QNX, 2008].

A 2.1.4 Photon Application Builder (PhAB)

The user interface for this application is built using Photon Application Builder (PhAB), which is a visual design tool that generates the underlying C and/or C++ code to implement the user interface. PhAB can be used for designing and creating modules (e.g. windows, menus, dialogs, and icons), and widgets (e.g. buttons and labels). It also helps to create widget callbacks, special resources that connect a widget to the application's code or link a widget to a PhAB module. The main user interface built using PhAB is illustrated in Figure A2.6, each button links to the corresponding callback, and text widgets are used for displaying the information of the system. Details of building an application with PhAB are described in Appendix 2.2.

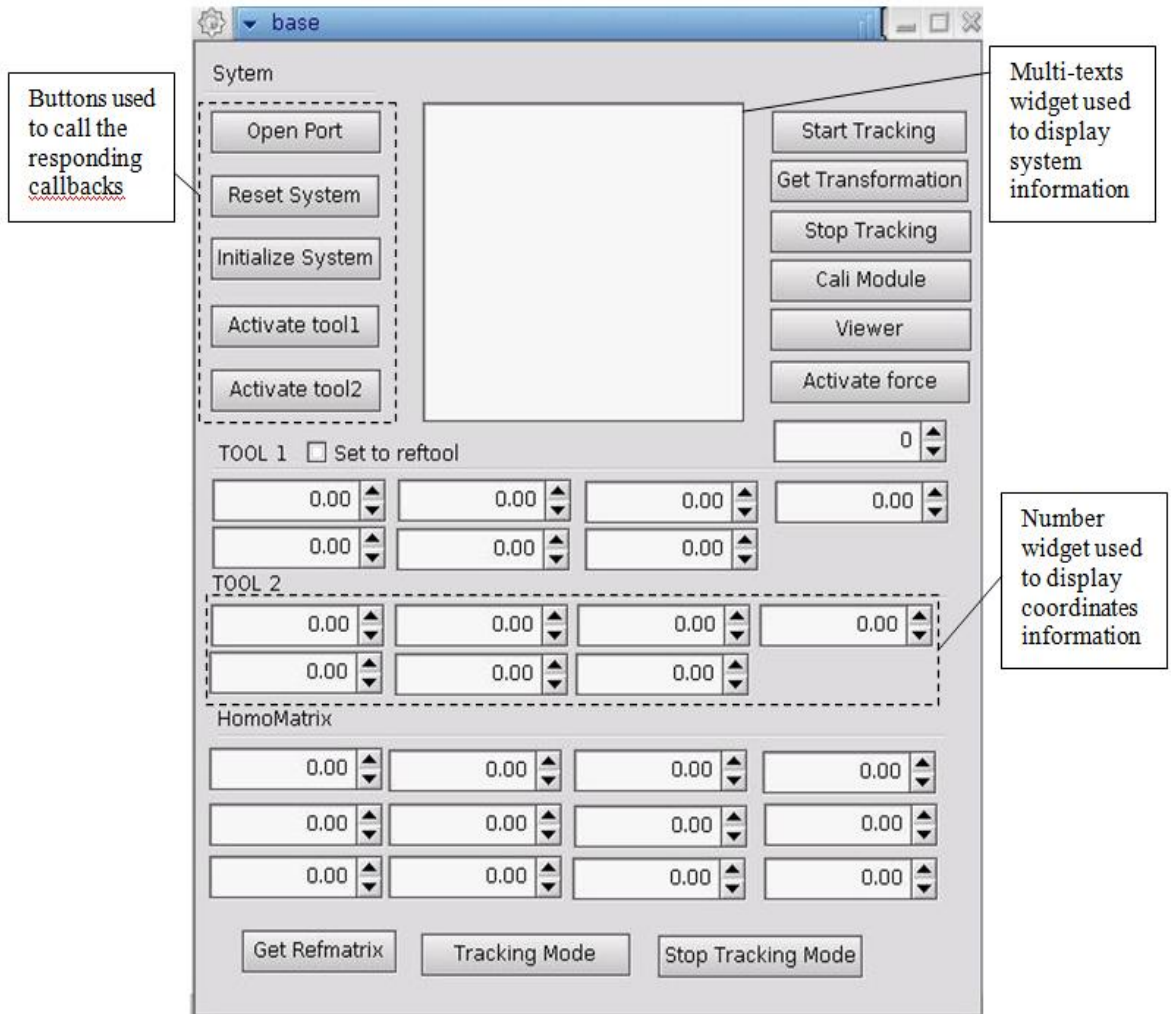


Figure A2.6 Design of the user interface.

A 2.2 Building an application using PhAb

The following steps are used to build applications with PhAB:

Step 1: Create modules. The primary building blocks to construct an application user interface in PhAB are called modules, with buttons placed on the module, as shown in Figure A2.6). One or more modules can be employed in an application, with a different role assigned to each of them. PhAB provides several module types:

- window--normally used for the application's major activities. A typical application has one main window that opens when the application starts up;

- dialog--allows the application to exchange information with the user;
- menu--presents commands to the user;
- picture--can be used in different ways. For example, you can use a picture to provide a convenient database of widgets or to change the contents of an existing module;

Step 2: Add widgets. Once a module is created, widgets are placed in it, by clicking on the appropriate icon in PhAB's widget palette, then clicking where to put on the module. PhAB provides several types of widgets including: display or edit values (e.g. labels, text, and multiline text); present choices (e.g. lists, comboboxes, and groups); display graphics (e.g. bitmaps, images, lines, rectangles, ellipses, and polygons) ; display scrolling areas (e.g. scrollbars and scrolling containers) initiate actions (e.g. buttons that contain text or images). A widget can also be customized by setting its attributes and resource.

Step 3: Attach callbacks. Callbacks are used to define how the application works. Every Photon widget supports several callback types. A code functions can be attached to a widget, and the widget invokes the code function whenever the conditions of the callback are met.

Step 4: Generate code. Once the modules, widgets and callbacks are completed, it is ready to generate and compile code to turn the application design into an executable application.

Step 5: Run your application. After the codes are generated, compiled, and linked to the application, it can be executed .

Step 6: Repeat any previous step. It is possible to change the interface, attach callbacks, and regenerate the code.

A 2.3 VAL3 programming

A 2.3.1 Variable types in VAL3 programming

VAL3 support the following simple types:

- **bool** type: for Boolean values (true/false)
- **num** type: for numeric values
- **string** type: for character strings
- **dio** type: for digital inputs/outputs
- **aio** type: for numeric inputs/outputs (analog or digital)
- **sio** type: for serial ports inputs/outputs and Ethernet sockets

VAL3 language also supports the following structured types:

- **trsf** type: for Cartesian geometrical transformations
- **frame** type: for Cartesian geometrical frames
- **tool** type: for robot mounted tools
- **point** type: for the Cartesian positions of a tool
- **joint** type: for robot revolute positions
- **config** type: for robot configurations
- **mdesc** type: for robot movement parameters

A 2.3.2 Programming with VAL3

The VAL3 language comprises seven key components: applications, programs, libraries, data types, constants variables (global data, local data and parameters) and tasks. An application is a self-contained software package in VAL3 for controlling the robot and I/O associated with a controller. It is similar to the concept of ‘project’ in visual C++, and the first step is to create an application to control a robot arm. An application usually comprises following elements: a set of programs, a set of global data, a set of libraries and a set of tasks. The global data is the variables used by all programs; and libraries are the external instructions and variables used in the application.

A program is a sequence of VAL instructions to be executed, which is the fundamental element in an application. The programs are used to group sequences of instructions that can be executed to control the robot and inputs/outputs of the controller. A program consists of the following elements:

- The sequence of instructions: the VAL3 instructions to be executed.
- A set of local variables: the internal program data.
- A set of parameters: the data supplied to the program when it is called.

A task is a program that is running. An application can have several tasks running such as: an arm movement task, a user interface task, communication tasks and so on. A task is defined as follows:

- a name: a task identifier that is unique in the library or application.
- a priority: a task sequencing parameter.
- a program: a task entry (and exit) point.
- a status: running or stopped.

Tasks can run asynchronously or synchronously. The *taskCreat()* instruction is used to create asynchronous tasks and *taskCreatSync()* for synchronous tasks. Synchronous tasks are required to be scheduled at regular periods of time for data acquisition or device control. They are executed in the sequencing cycle by interrupting the current asynchronous task between two VAL3 lines. When the synchronous tasks have finished, the asynchronous task resumes. The sequencing of the VAL3 synchronous tasks obeys the following rules:

1. Each synchronous task is sequenced exactly once per period of time specified at the task creation.
2. At each sequence, the system executes up to 3000 VAL3 instruction lines. It shifts to the next task when an instruction line cannot be completed immediately. In practice, a synchronous task is often explicitly ended by using the "*delay(0)*" instruction to force the sequencing of the next task.

3. The synchronous tasks with same period are sequenced in the order in which they were created.

A VAL3 variable or constant type is a characteristic that allows the system to control the applications and programs that can use it. The VAL3 language divides the variables and constants in to two major types: simple types and structured type. All the VAL3 constants and variables have a type. This enables the system to run an initial check when editing a program and hence detect certain programming errors immediately. More details of the types are given in Appendix 2.2. Once the definitions of variable types, programs and tasks of the application are clarified, the application can be built using the following steps:

- Build an application with the application manage on MCP. It can be seen from Figure A2.7, four elements (libraries, global data, programs and parameters) are included in the application. The *io* library is compulsory for controlling the inputs/outputs of the controller. The parameters are global parameters and include the unit of speed, unit of length, IP address of the Ethernet port used in the application.

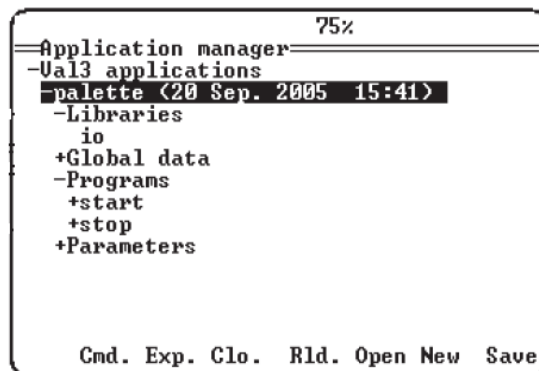


Figure A2.7 Application manager.

- Define the global data in the global data element such as the maximum speed of the robot arm, a reference point in world space etc.
- Create the new programs in the *programs* element according to specific application. Local variables and parameters can be defined in a program and every program can be called from another program. In addition, a program

can be used as a task if the *taskCreate()* instruction is used to define the program.

- Load the application onto the controller.

A 2.4 Automatic ‘hot-spot’ finder module

This section was developed by Prof. Stuart Baker, and is included here for completeness. The software module for automatic ‘hot-spot’ finder comprises three main sections: 3D points collection, ellipsoid fitting and Bayesian inference search algorithm (Figure A2.8).

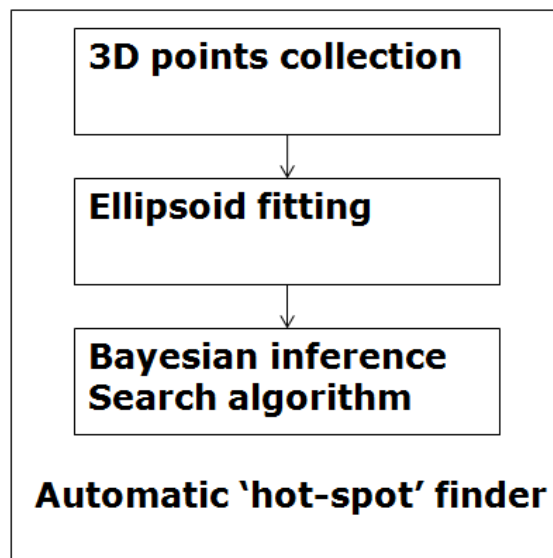


Figure A2.8 Components of the automatic ‘hot-spot’ finder

The control user interface is illustrated as (Figure A2.9), which runs as a sequence. First of all, the 3D points from the Polaris are collected with respect to the subject’s head coordinate system and the points prepared for use in the next step. Once the 3D points collection is completed, ellipsoid fitting using the ‘Down-hill’ simplex method is carried out to map the head surface. Finally, the Bayesian inference search algorithm is applied to find the ‘hot-spot’.

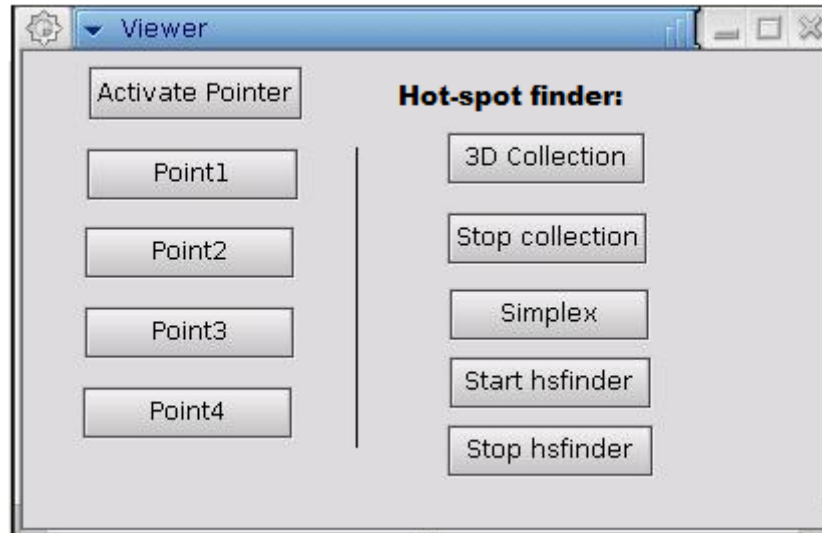


Figure A2.9 User interface of automatic 'hot-spot' finder

The tool used for collecting the 3-dimensional points is called a pointer, which consists of four markers and a metal tip. Once the system begins collecting data, the experimenter holds the pointer and uses the tip to pass across the subject's scalp until sufficient points are collected. All the points are random points distributed on the subject's head, and the data is saved in a single file (*.txt).



Figure A2.10 The pointer used in the test

A 2.5 Detailed information of the system and port status

System status has 4 hexadecimal characters (16 bits) and the Bit field is:	
bit 0	System communication synchronization error
bit 1	Too much external IR.
bit 2	System CRC error
bit 3	Recoverable system processing exception.
bits 4 and 5	Reserved
bit 6	Some port handle has become occupied
bit 7	Some port handle has become unoccupied
bit 8	Diagnostic pending
bit 9	Temperature (system is not within operating temperature range)
bits 10 to 15	Reserved

Port status has 8 hexadecimal characters (32 bits) and the Bit field is:	
bit 0	Occupied
bit 1	Switch 1 closed
bit 2	Switch 2 closed
bit 3	Switch 3 closed
bit 4	Initialized
bit 5	Enabled
bit 6	Out of volume
bit 7	Partially out of volume
bit 8	Algorithm limitation (processing requires more buffer than is available)
bit 9	IR interference (a large bright IR object)
bits 10 and 11	Reserved
bit 12	Processing exception (same as tool information bit 7 in reply option 0002)
bit 13	Reserved
bit 14	Fell behind while processing (same as tool information bit 3 in reply option 0002)
bit 15	Data buffer limitation (too much data; for example, too many markers)
bit 16 to 31	Reserved

A 2.6 Code for parsing the raw data

```
int nGetTransforms()
{
    Char *pszTransformInfo = NULL; //define the pointer
```

```

/* Parsing Routines */
pszTransformInfo = m_szLastReply;

nNoHandles = uASCIIToHex( pszTransformInfo, 2 ); /* Get number of handles */
pszTransformInfo+=2;

for ( int i = 0; i < nNoHandles; i++ )
{
    nHandle = uASCIIToHex( pszTransformInfo, 2 );
    pszTransformInfo+=2;

    m_dtHandleInformation[nHandle].Xfrms.ulFlags = TRANSFORM_VALID;
/*get rotation and translational parts*/
bExtractValue( pszTransformInfo, 6, 10000.,&m_dtHandleInformation[nHandle].Xfrms.rotation.q0 )

bExtractValue( pszTransformInfo + 6, 6,
10000.,&m_dtHandleInformation[nHandle].Xfrms.rotation.qx )
bExtractValue( pszTransformInfo + 12,
6,10000.,&m_dtHandleInformation[nHandle].Xfrms.rotation.qy )
bExtractValue( pszTransformInfo + 18,
6,10000.,&m_dtHandleInformation[nHandle].Xfrms.rotation.qz )
bExtractValue( pszTransformInfo + 24, 7,
100.,&m_dtHandleInformation[nHandle].Xfrms.translation.x )
bExtractValue( pszTransformInfo + 31, 7,
100.,&m_dtHandleInformation[nHandle].Xfrms.translation.y )
bExtractValue( pszTransformInfo + 38, 7,
100.,&m_dtHandleInformation[nHandle].Xfrms.translation.z )
bExtractValue( pszTransformInfo + 45, 6,
10000.,&m_dtHandleInformation[nHandle].Xfrms.fError ) )

pszTransformInfo+=51;

/*get frame number...*/
m_dtHandleInformation[nHandle].Xfrms.ulFrameNumber = uASCIIToHex( pszTransformInfo, 8 );
pszTransformInfo+=8;
pszTransformInfo++; /*for the carriage return*/
} /* for */

} /* nGetTransforms */

```

A 2.7 'C' structure of the Analyze head file

```

/* ANALYZE™ Header File Format
*
* (c) Copyright, 1986-1995
* Biomedical Imaging Resource
* Mayo Foundation
*
* dbh.h
*
* databse sub-definitions
*/
struct header_key /* header key */
{ /* off + size */

```

```

int sizeof_hdr /* 0 + 4 */
char data_type[10]; /* 4 + 10 */
char db_name[18]; /* 14 + 18 */
int extents; /* 32 + 4 */
short int session_error; /* 36 + 2 */
char regular; /* 38 + 1 */
char hkey_un0; /* 39 + 1 */
}; /* total=40 bytes */

```

```

struct image_dimension
{ /* off + size */
short int dim[8]; /* 0 + 16 */
short int unused8; /* 16 + 2 */
short int unused9; /* 18 + 2 */
short int unused10; /* 20 + 2 */
short int unused11; /* 22 + 2 */
short int unused12; /* 24 + 2 */
short int unused13; /* 26 + 2 */
short int unused14; /* 28 + 2 */
short int datatype; /* 30 + 2 */
short int bitpix; /* 32 + 2 */
short int dim_un0; /* 34 + 2 */
float pixdim[8]; /* 36 + 32 */
/*

```

pixdim[] specifies the voxel dimensions:

pixdim[1] - voxel width

pixdim[2] - voxel height

pixdim[3] - interslice distance

...etc

*/

```

float vox_offset; /* 68 + 4 */
float funused1; /* 72 + 4 */
float funused2; /* 76 + 4 */
float funused3; /* 80 + 4 */
float cal_max; /* 84 + 4 */
float cal_min; /* 88 + 4 */
float compressed; /* 92 + 4 */
float verified; /* 96 + 4 */
int glmax, glmin; /* 100 + 8 */
}; /* total=108 bytes */

```

```

struct data_history
{ /* off + size */
char descrip[80]; /* 0 + 80 */
char aux_file[24]; /* 80 + 24 */
char orient; /* 104 + 1 */
char originator[10]; /* 105 + 10 */
char generated[10]; /* 115 + 10 */
char scannum[10]; /* 125 + 10 */
char patient_id[10]; /* 135 + 10 */
char exp_date[10]; /* 145 + 10 */
char exp_time[10]; /* 155 + 10 */
char hist_un0[3]; /* 165 + 3 */
int views /* 168 + 4 */
int vols_added; /* 172 + 4 */
int start_field; /* 176 + 4 */
int field_skip; /* 180 + 4 */
int omax, omin; /* 184 + 8 */
int smax, smin; /* 192 + 8 */

```

```

};

struct dsr
{
struct header_key hk; /* 0 + 40 */
struct image_dimension dime; /* 40 + 108 */
struct data_history hist; /* 148 + 200 */
}; /* total= 348 bytes */
/* Acceptable values for datatype */
#define DT_NONE 0
#define DT_UNKNOWN 0
#define DT_BINARY 1
#define DT_UNSIGNED_CHAR 2
#define DT_SIGNED_SHORT 4
#define DT_SIGNED_INT 8
#define DT_FLOAT 16
#define DT_COMPLEX 32
#define DT_DOUBLE 64
#define DT_RGB 128
#define DT_ALL 255
typedef struct
{
float real;
float imag;
} COMPLEX;

```