

**M.Tech (R)**

**Madhusmita  
Senapati**

**Design and Control of an Articulated Robotic Arm  
using Visual Inspection for Replacement Activities**



2016

# **Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Activities**

**Madhusmita Senapati**



Department of Mechanical Engineering  
**National Institute of Technology Rourkela**

# **Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Activities**

*Dissertation submitted in partial fulfillment  
of the requirements of the degree of*

***Master of Technology (Research)***

*in*

***Mechanical Engineering***

*By*

***Madhusmita Senapati***

(Roll Number: 613ME6008)

*based on research carried out*

*under the supervision of*

***Prof. J Srinivas***



July, 2016

Department of Mechanical Engineering  
**National Institute of Technology Rourkela**



Department of Mechanical Engineering  
**National Institute of Technology Rourkela**

---

July 15, 2016

## **Certificate of Examination**

Roll Number: *613ME6008*

Name: *Madhusmita Senapati*

Title of Dissertation: *Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Activities*

We the below signed, after checking the dissertation mentioned above and the official record book(s) of the student, hereby state our approval of the dissertation submitted in partial fulfilment of the requirements of the degree of *Master of Technology (Research)* in *Mechanical Engineering* at *National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness and originality of the work.

---

J Srinivas  
Principal Supervisor

---

D.R.K. Parhi  
Member, MSC

---

B. B. Biswal  
Member, MSC

---

Dipti Patra  
Member, MSC

---

External Examiner

---

S.K.Patel  
Chairman, MSC

---

S.S.Mohapatra  
Head of the Department



Department of Mechanical Engineering  
**National Institute of Technology Rourkela**

---

**Prof. Jonnalagadda Srinivas**

Associate Professor

July 15, 2016

## **Supervisor's Certificate**

This is to certify that the work presented in this dissertation entitled *Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Activities* by Miss. *Madhusmita Senapati*, Roll Number 613ME6008, is a record of original research carried out by her under my supervision and guidance in partial fulfilment of the requirements of the degree of *Master of Technology (Research) in Mechanical Engineering*. Neither this dissertation nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

---

J Srinivas  
Associate Professor



# Dedication

*Dedicated to Lord Jagannath  
Family and Friends*

*Madhusmita Senapati*

# Declaration of Originality

I, *Madhusmita Senapati*, Roll Number *613ME6008* hereby declare that this dissertation entitled *Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Activities* presents my original work carried out as a Master of Technology (Research) of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections “Reference” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation

July 15, 2016

NIT Rourkela

*Madhusmita Senapati*

Roll Number: 613ME6008

# Acknowledgment

I sincerely thank Prof. J Srinivas for all the time and effort he put in helping me in this work. I have selected the problem and known the developments in this area from time to time by constantly getting in touch with literature from Prof. Srinivas. I thank for his encouragement in making me to understand the complete details relating to manipulators.

I am thankful to our director Prof. Sunil Kumar Sarangi, and very much obliged to the Head of Department of Mechanical Engineering Prof. S.S. Mohapatra, NIT Rourkela for providing all the support and concern regarding my academic requirements.

I would like to thank to all my MSC members, Prof. S.K.Patel, Prof. D.R.K. Parhi, Prof. B.B Biswal, Industrial Design and Prof. Dipti Patra, Electrical Engineering, for their valuable suggestions and comments during this research work and other faculty members of the institute for their co-operation and help.

I owe my largest debt to my family. I take this opportunity to express my regards and obligation to my family members for encouraging me in all aspects for carrying out the research work.

Specially, I extend my deep sense of indebtedness and gratitude to all my colleagues Pranjal Bhuyan, K.V.Varalakshmi, Jakeer Hussain Shaik, Puneet Kumar, Prabhu L, Subhransu Padhee, Rajasekhara Reddy, Aditya Andra, Alok Sir and many other M.Tech friends who helped me to complete the project directly and indirectly. I am thankful to all the teaching & non-teaching staff of Mechanical Engineering Department for their kind cooperation. At last, I would like to acknowledge the financial support provided by the BRFSST (Board of Research in Fusion Science and Technology), Gandhinagar for encouragement of this project

July 15, 2016  
NIT Rourkela

*Madhusmita Senapati*  
Roll Number:613ME6008

# **Abstract**

Design of robotic systems and their control for inspection and maintenance tasks is highly complex activity involving coordination of various sub-systems. In application like inspections in fusion reactor vessels and deep-mining works, a regular off-line maintenance is necessary in certain locations. Due to the hostile environments inside, robotic systems are to be deployed for such internal observations. In this regard, current work focuses on the methodology for maintenance of the first wall blanket modules in a fusion reactor vessel using a manipulator system. A design is proposed for wall tile inspections in an ideal environment in which vacuum and temperature conditions are not accounted and wall surface curvature is not accounted initially. The entire design has four important modules: (i) mathematical modelling (ii) control system design (iii) machine vision and image processing, (iv) hardware development and testing.

A five- axis articulated manipulator equipped with a vision camera in eye-to-hand configuration is designed for performing the pick and place operations of the defected tiles in a systematic manner. Kinematic and dynamics analysis of the system are first carried-out and a scaled prototype is fabricated for testing various operating issues. Forward kinematics of manipulator allows in estimation of robot workspace and in knowing the singular regions during operation, while the inverse kinematics of the manipulator would be needed for real time manipulator control task. Dynamics of manipulator is required for design of model-based controllers. Interactive programs are developed in Matlab for kinematics and dynamics and three-dimensional manipulator assembly configuration is modelled in SolidWorks software. Motion analysis is conducted in ADAMS software in order to compare the results obtained from the classical kinematics. Two types of model-based control schemes (namely Computed Torque Control and Proportional Derivative-Sliding Mode Control approach) with and without external disturbances are implemented to study trajectory tracking performance of the arm with different input trajectories. A disturbance observer model is employed in minimizing the tracking errors during the action of external disturbances such as joint friction and payload.

In order to experimentally understand the inspection and replacement activities, a test set-up is developed using vision camera and microcontroller platform to guide the robot joint servos so as to perform defected object replacement activity. Presence of crack and the coordinate of the region are indicated with the use of image-processing operations.

Using a high resolution Basler camera mounted at fixed distance from the tile surface, the surface images are acquired and image processing module identifies the crack details using edge detection algorithms. Necessary motion of the end-effector will be provided based on the pose calculations using coordinate transformations. Both visual inspection and joint guidance are combined in a single application and the results are presented with a test case of tile replacement activity. The results are presented sequentially using a test surface with uniform rectangular tiles.

***Keywords: Articulated robot; Kinematics; Dynamics, Trajectory tracking control; Image-processing; Fabrication and testing***

# Contents

<b>Certificate of Examination.....</b>	<b>ii</b>
<b>Supervisor's Certificate.....</b>	<b>iii</b>
<b>Dedication.....</b>	<b>iv</b>
<b>Declaration of Originality .....</b>	<b>v</b>
<b>Acknowledgment .....</b>	<b>vi</b>
<b>Abstract .....</b>	<b>vii</b>
<b>List of Figures .....</b>	<b>ix</b>
<b>List of Tables.....</b>	<b>xv</b>
<b>Chapter 1.....</b>	<b>1</b>
<b>1.1    Inspection tasks in nuclear vessels.....</b>	<b>1</b>
<b>1.2    Statement of Problem .....</b>	<b>3</b>
<b>1.3    Scope and Objectives of the work.....</b>	<b>4</b>
<b>1.4    Outline of the thesis .....</b>	<b>4</b>
<b>Chapter 2.....</b>	<b>6</b>
<b>2.1    Robotic manipulators in industries .....</b>	<b>6</b>
<b>2.2    Manipulator Configurations and Modeling .....</b>	<b>8</b>
<b>2.3    Remote Control Architectures.....</b>	<b>11</b>
<b>2.4    Use of Vision Sensor in Robots and Image Processing.....</b>	<b>12</b>
2.4.1.    Crack detection using Image Processing .....	14
<b>2.5    Virtual Reality and mock-up models .....</b>	<b>15</b>
<b>2.6    Theoretical/ practical control tasks.....</b>	<b>15</b>
<b>2.7    Conclusion .....</b>	<b>17</b>
<b>Chapter 3.....</b>	<b>18</b>
<b>3.1    Kinematics .....</b>	<b>18</b>
3.1.1    Forward Kinematics .....	18
3.1.2    Inverse Kinematics .....	23
3.1.3    Workspace Envelope .....	27
3.1.4    Jacobian Analysis .....	28
<b>3.2    Dynamics.....</b>	<b>30</b>
<b>3.3    Numerical studies.....</b>	<b>31</b>
3.3.1    Inverse Kinematics .....	34

3.3.2	Inverse Dynamics .....	36
3.3.3	Forward Dynamics .....	42
<b>3.4</b>	<b>Conclusion .....</b>	<b>43</b>
<b>Chapter 4.....</b>	<b>.....</b>	<b>44</b>
<b>4.1</b>	<b>Introduction.....</b>	<b>44</b>
<b>4.2</b>	<b>Computed Torque Controller (CTC).....</b>	<b>44</b>
<b>4.3</b>	<b>PD-Sliding Mode Controller (SMC).....</b>	<b>47</b>
<b>4.4</b>	<b>Design of Disturbance Observer .....</b>	<b>48</b>
<b>4.5</b>	<b>Numerical studies for selected Trajectories .....</b>	<b>51</b>
4.5.1	First case .....	51
4.5.2	Second case.....	62
<b>4.6</b>	<b>Conclusion .....</b>	<b>72</b>
<b>Chapter 5.....</b>	<b>.....</b>	<b>73</b>
<b>5.1</b>	<b>Introduction.....</b>	<b>73</b>
<b>5.2</b>	<b>Types of Visual Servo Control .....</b>	<b>74</b>
<b>5.3</b>	<b>Image acquisition .....</b>	<b>78</b>
<b>5.4</b>	<b>Image Processing.....</b>	<b>79</b>
5.4.1	Histogram Equalization .....	81
5.4.2	Noise Removal and Image Sharpening .....	82
5.4.3	Image Thresholding .....	82
5.4.4	Morphological Operators.....	82
5.4.5	Edge detection techniques .....	83
5.4.6	Template Matching .....	83
<b>5.5</b>	<b>Case study of simulated wall tiles .....</b>	<b>83</b>
5.5.1	Features of the camera .....	84
5.5.2	Image processing with a group of tiles having random cracks.....	86
<b>5.6</b>	<b>Conclusion .....</b>	<b>96</b>
<b>Chapter 6.....</b>	<b>.....</b>	<b>97</b>
<b>6.1</b>	<b>Concept Design of the articulated robotic manipulator .....</b>	<b>97</b>
6.1.1	Base .....	98
6.1.2	Lower Arm .....	98
6.1.3	Upper Arm.....	99
6.1.4	End-Effector Casing .....	100

6.1.5	End-effector .....	100
<b>6.2</b>	<b>Fabrication of the prototype .....</b>	<b>102</b>
6.2.1	Machining of the Robotic Arm.....	102
<b>6.3</b>	<b>Electronics Interface .....</b>	<b>104</b>
6.3.1	Microcontroller Platform .....	105
6.3.2	Servo Motors .....	106
<b>6.4</b>	<b>Driving Tests.....</b>	<b>108</b>
<b>6.5</b>	<b>Conclusion .....</b>	<b>112</b>
<b>Chapter 7</b>	<b>.....</b>	<b>114</b>
7.1	Summary.....	114
7.2	Future scope .....	116
<b>Bibliography</b>	<b>.....</b>	<b>117</b>
<b>Appendix A</b>	<b>.....</b>	<b>122</b>
<b>Appendix B</b>	<b>.....</b>	<b>127</b>
<b>Appendix C</b>	<b>.....</b>	<b>133</b>
<b>Vitae</b>	<b>.....</b>	<b>137</b>



## List of Figures

Figure 1.1 Concept of remote guidance of inspection arm.....	3
Figure 2.1 Articulated arm nomenclature [21] .....	8
Figure 2.2 Sterling series FARO arm with 1.5 m long with 6-DOF [22].....	9
Figure 2.3 Five-axis SCORBOT-ER VII [18].....	10
Figure 2.4 Five-axis articulated mobile manipulator with Human-machine Interface [24] .....	10
Figure 2.5 A 13-axis Remote Handling Robot [25] .....	11
Figure 2.6 Inspection Robot [8].....	11
Figure 3.1 Link Coordinate frame of the present manipulator .....	20
Figure 3.2 Inverse kinematics from geometric approach .....	25
Figure 3.3 Workspace Computations .....	32
Figure 3.4 Manipulability index within the work envelope .....	33
Figure 3.5 Variation of dexterity index within the workspace .....	34
Figure 3.6 An input trajectory considered .....	34
Figure 3.7 Inverse kinematic solution using geometric method.....	36
Figure 3.8 Joint trajectory considered .....	38
Figure 3.9 Required joint torques .....	39
Figure 3.10 Motion simulation of model in ADAMS .....	40
Figure 3.11 Joint torques .....	42
Figure 3.12 Outputs of forward dynamics.....	43
Figure 4.1 Block –Diagram of Computed Torque Control .....	46
Figure 4.2 Block –Diagram of Disturbance Observer .....	49
Figure 4.3 Flow chart of trajectory tracking process.....	51
Figure 4.4 Joint Error plots without disturbance (case 1).....	53
Figure 4.5 Joint Control Torques variations without disturbance (case 1).....	54
Figure 4.6 Joint Error plots with disturbance case .....	55
Figure 4.7 Joint Control Torques variations with disturbance (case 1).....	56
Figure 4.8 Resultant control torques (N-m) with disturbance observer .....	57

Figure 4.9 Error at joints with disturbance observer .....	58
Figure 4.10 Desired trajectory vs. Actual trajectory .....	58
Figure 4.11 Control torques (N-m) along the trajectories .....	59
Figure 4.12 Joint Error.....	60
Figure 4.13 Joint errors along the trajectory.....	60
Figure 4.14 resultant joint control torques.....	61
Figure 4.15 Desired trajectory vs. Actual trajectory (CTC) .....	62
Figure 4.16 Joint Control Torques variations without disturbance (case 2).....	63
Figure 4.17 Joint errors along the trajectory at all individual joints.....	64
Figure 4.18 Trajectory tracking performance of CTC.....	64
Figure 4.19 Joint errors along the trajectory.....	65
Figure 4.20 Joint Control Torques variations with disturbance (case 2).....	65
Figure 4.21 Trajectory tracking performance of CTC with DO.....	66
Figure 4.22 Error at joints with disturbance observer (case 2).....	66
Figure 4.23 Joint Control Torques variations with disturbance Observer (case 2) .....	67
Figure 4.24 Joint Control Torques variations with disturbance (case 2).....	68
Figure 4.25 Error at joints with disturbance (case 2.....	69
Figure 4.26 Joint Control Torques.....	70
Figure 4.27 Joint Error plots with disturbance observer case.....	71
Figure 5.1 Simple block diagram of image based visual servo control.....	75
Figure 5.2 Simple block diagram of position visual servo control.....	77
Figure 5.3 Types of camera placements .....	79
Figure 5.4 Flowchart of the image processing steps .....	81
Figure 5.5 Developed experimental setup with camera placement .....	85
Figure 5.6 Flowchart of image processing steps for vision based fault identification .....	85
Figure 5.7 Captured image .....	86
Figure 5.8 Enhanced image .....	87
Figure 5.9 Segmented images.....	87
Figure 5.10 Crack detection.....	88
Figure 5.11 Vision Assistant toolbox screenshot for tiles without crack .....	89

Figure 5.12 The corresponding histogram for the grayscale image .....	89
Figure 5.13 Histogram Analysis .....	90
Figure 5.14 Gray scale Images of the tile 5 .....	91
Figure 5.15 Histogram analysis of tile 5 .....	92
Figure 5.16 Block diagram of image acquisition using LabVIEW .....	93
Figure 5.17 Front panel detecting the cracked regions .....	93
Figure 5.18 Schematic of tile placement on the wall .....	94
Figure 5.19 Important frames in eye-to-hand configurations .....	95
Figure 5.20 Location of camera with respect to robot base .....	96
Figure 6.1 Drawing view and solid model of the base .....	98
Figure 6.2 Drawing views and solid model of the lower arm .....	99
Figure 6.3 Drawing views and isometric view of the upper arm .....	99
Figure 6.4 Drawing views and solid model of the end-effector casing .....	100
Figure 6.5 Assembly model in SolidWorks .....	101
Figure 6.6 Fabricated Robot Arm .....	103
Figure 6.7 Arduino UNO schematic .....	106
Figure 6.8 Different components of servo motor .....	107
Figure 6.9 Arduino and servo motor connection configuration .....	107
Figure 6.10 Schematic diagram of the whole setup of the present work .....	108
Figure 6.11 Process sequence of tile replacement (Home position → Tile surface → Waste tile bin → new tile bin → Tile surface) .....	109
Figure 6.12 Screenshot of Arduino program .....	110
Figure 6.13 Arduino circuit connection .....	111

## List of Tables

Table 3.1 D-H Link parameters of present manipulator.....	20
Table 3.2 Dimensions and mass properties of Arm.....	38
Table 4.1 Simulation parameters used [61, 48] .....	52
Table 4.2 Maximum torque required for the trajectory 1 .....	72
Table 5.1 Camera specifications.....	84
Table 5.2 Computational results for crack detection technique .....	88
Table 5.3 Computational results for crack detection technique .....	94
Table 6.1 Manipulator summary .....	101
Table 6.2 Salient features of Arduino Uno.....	105
Table 6.3 Configurations of the servos.....	108

# Nomenclature

$T$	Transformation matrix	$k_p$	Position gain
$\alpha$	Link twist	$k_v$	Velocity gain
$\theta$	Joint angle	$H$	SMC- sliding gain vector
$a$	Link length	$\lambda$	Sliding surface slope constant
$d$	Link offset	$sign$	Sign function
$c$	cos	$\lambda$	Sliding surface slope constant
$s$	sin	$v$	Eigen vector
$p_x$	Position in x-axis	$D$	disturbance
$p_y$	Position in y-axis	$e$	Error
$p_z$	Position in z-axis	$t$	Time
$J$	Jacobian matrix	$F$	Force
$\tau$	Torque		
$M$	Mass & Inertia matrix		
$C$	Corialis vector		
$G$	Gravity vector		

# Chapter 1

## Introduction

Robotics, automation and remote handling technology plays a vital role in almost all facets of nuclear inspection task. The recent advancements in this fascinating area have been due to various necessities unique to nuclear industry such as starting from reducing the radiation expos during operation, technologies requirement to facilitate remote inspection at inaccessible areas of nuclear reactor or nuclear plants or to facilitate remote repair/refurbishments of operating plants. Remote handling/robotic tool design is essential in the areas of inspection and replacement tasks. Advancements in this technology, by way of mathematical modelling, control/automation, advance vision system and various sensors coupled with experimental works are facilitating applications in the in-vessel inspection or repair or maintenance of nuclear power plants.

With the tenacious need for increased quality, productivity and automation, the world is tuning more and more towards various autonomous and semi-autonomous robots which finds a wide array of application in various fields such as inspection, surveillances, quality check, fault detection, surgical, rehabilitation, agriculture, planetary space exploration etc. A common attribute of such described applications is that robot needs to operate in inhuman, unstructured environment where human intervention is risky. Motion control, trajectories planning for robots in unstructured environments face significant challenges due to various uncertainties in internal as well as external environment. So a complete study and analysis of mathematical modelling, control, various sensor systems such as vision is essentially needed.

### 1.1 Inspection tasks in nuclear vessels

High temperature and low vacuum condition often prevail in fusion reactor vessels. Hazardous environment continuously need monitoring from various perspectives including, gas deposition of transparent surfaces, tile breakage and so on. Manipulator linkages can be deployed in such conditions to minimize the risk and time of inspection. Computerized human-machine interfaces are used in many plants to control the manipulators and robots. For the last decades, advanced 3D computer models have been

used to visualize and monitor live operations. Before execution, these system are being used to simulate operations in virtual reality leading to real world mock-up operations.

Fusion reactors are increasingly becoming popular in recent times, due to their several advantages. Fusion of light elements like deuterium and tritium is achieved by a device called ‘tokamak’ designed based on the concept of magnetic confinement of the plasma .Under the fusion research program, India is working on tokamak called *Aditya*, a machine for research on plasma physics. This is a part of the worldwide project known as international thermonuclear experimental reactor (ITER).

The components close to the hot plasma require replacement due to erosion and damage. As the surface has to face 14.1 MeV neutron energy generated due to fusion of deuterium and tritium, the materials must have radiation resistance. The ITER vacuum vessel is covered by blanket modules, which are designed to be replaceable during normal scheduled maintenance. Due to the Deuterium-Tritium pulses, the levels of contaminations and radioactivity continuously changes inside the vessel. Hence, these replacement operations are carried –out by means of remote handling (RH) procedures using remote maintenance system. For example, the ITER divertor system has several removable cassettes weighing between 9 and 11 tons which require remote maintenance and replacement on a scheduled basis with the use of special transporters, robotic arms and tooling. These RH operations are like unlocking, removal and transportation of the expended diverter system from the tokamak building to the hot-cell building and replacing by spare units stored in the hot-cell. In addition, sometimes the failed cassettes are also to be replaced during ITER operation. The design of such a RH system is one of the major challenges in ITER remote maintenance system. Often, a vehicle manipulator system is employed to handle several such tasks. Various non-contact sensing system like, virtual reality simulators and robot vision may be used in these vehicles performing inspection and replacement operations. In a robot vision system, cameras are used like human eyes and can give feedback signal in the form of computer images. Now-a-days, radiation tolerant video camera tubes are also available leading to this approach as one of the commonly used non-contact sensing approach. This chapter briefly presents the existing systems and the implementation issues. Because of release of high energy neutron in tokamak device, the materials employed for construction of the first wall, the divertor and the blanket segments should have high structural and thermal characteristics. Several first wall materials currently in-use are: Boron carbide, graphite, carbon fiber composites, Beryllium, Tungsten and Molybdenum. Today Beryllium has been replaced by carbon as a

plasma facing material in main chamber. However, due to expose of poloidal faces to high heat fluxes, the Beryllium may melt into the chamber polluting the plasma.

Therefore, the tile inspection and replacement tasks are routine procedure during maintenance. The inspection tasks can be carried-out using high speed, Infra Red (IR) CCD cameras moving inside the wall chamber along with a deployed transporter. Often, the toroidal/cylindrical vessel may require two or more such robotic transporters to cover the entire circumference. Keeping a laser–range finder along with eye-in–hand camera set-up, permits the monitoring task more easily. The operator sits at a remote location and provides guidance to the transporter by joy sticks with the help of the received computer images.

## 1.2 Statement of Problem

Based on the available inspection procedures in high temperature vacuum vessels, there is a requirement to design a low-cost robotic platform which is to be operated remotely through the vision camera data. This should facilitate in avoiding human involvement in such hazardous polluted environments. Design of such a robotic platform requires several consideration including trajectory tracking and force control during replacement events. In such tasks, redundant robots are advantageous; however their control issues do not permit their usage everywhere.

In this regard, design and development of a serial articulated multi-link mechanism is a necessary task. As in conventional robots, the linkage needs joint control to guide the end effector. At the same time, it requires feedback signals of current state to know the motion states. Further, it also needs a kind of remote control of joint motions. All these tasks are to be tested using a test bed idealizing a sector of a hot-vessel. Figure 1.1 shows the concept of the problem.

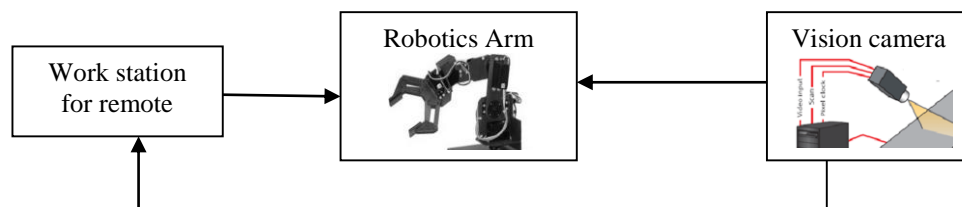


Figure 1.1 Concept of remote guidance of inspection arm

The observer at remote location directs the robotic arm for replacement activities using the processed image data received from a camera. The coordinate positions of the object at



replacement site are predicted and the end effector moves the faulty object and replaces with a new one.

### **1.3 Scope and Objectives of the work**

Most of the available literature relating to the maintenance activities of vacuum vessels focuses on the human-in-loop strategy. The guidance of the robotic vehicle for capturing of images and replacement of faulty objects in fact requires proper joint control methodology based on inverse kinematics and dynamics. Further, the conventional 2-finger gripper may not perform the desired job for unscrewing and removing a faulty object. So, it requires a design of end-effector which can hold the tools for replacement at a particular payload. Remote handling task is another issue to be focused in such critical environments. In such cases, a hybrid control of force and motion is required.

The main objectives of the present work are:

- i. Design and fabrication of articulated arm with wrist and gripper for holding the tiles of required size and weight.
- ii. Kinematics studies to know the workspace and singularity states.
- iii. Dynamic simulations of robotic arm to understand virtual behavior.
- iv. Mechatronic system design for motor guidance and control of end-effector paths over ideal surfaces.
- v. Practical studies with vision camera to identify the abnormal regions on the surface.
- vi. Application of image processing tools to locate the object corners.
- vii. Design a remote control system to drive the robotic vehicle smoothly.

### **1.4 Outline of the thesis**

All the process involved in the designing, analyzing, fabricating, sensing and controlling of an articulated robotic manipulator has been categorized into following chapters.

Chapter 2 presents the review of literature pertaining to robotic manipulator design, kinematic and dynamic modelling, control activities, image processing and various experimental procedures relating to in-vessel inspection and replacement activities.

Chapter 3 deals with the mathematical modelling of the proposed 5-axis articulated arm. This chapter studies forward and inverse kinematics as well as the dynamic model of the manipulator using conventional procedures. Workspace analysis, manipulability and dexterity of the manipulator are studied.

Chapter 4 discusses theoretical control schemes which modulate the robot's behavior to achieve the desired joint space trajectories. Two commonly used control schemes namely computed torque control and proportional-derivative sliding mode control are described with reference to the present 5-axis arm. The effectiveness of the controllers during disturbance torques is also studied and a disturbance observer is designed to minimize the errors.

Chapter 5 describes the use of vision system for the 2-D surface monitoring and image processing issues employed in the present work. Approaches for inspection tasks for crack or fault detection on the wall surfaces are presented using edge-detection algorithms. Various steps such as imaging geometry, image acquisition, image processing, crack recognition using edge detection technique, interpretation etc. are described.

Chapter 6 deals with fabrication and testing of the proposed robotic arm for tile replacement activities. Various stages in fabrication process involved in making the robot and materials used have been stated. Using vision camera information, the cracked or defected object location is identified and the end-effector of manipulator is directed to reach the edges of the tile/object for pick and replacement activity. The details of electronic circuit for controlling the joint servos motors using Arduino controller are presented. Experimental tests conducted using vision camera to identify some simulated cracks and subsequent servo-guidance effectiveness for pick and place operation are illustrated with wooden tile board.

Finally, conclusions and future scope of the work are provided in Chapter-7 and some appendices on existing edge detection methods and basic computer programs used in the work are given at the end.

## Chapter 2

# Literature Review

With advancements in technology, there has also been vast renovation in the field of robotics. Robotic mechanisms are used in inspection and control tasks in various engineering sciences including agriculture, bio-technology, chemical, defense, electronics, food-processing, fusion sciences and so on. Exhaustive literature is available in order to obtain some broad understanding of various aspect and factors concerning robotic manipulators Literature can be further divided under various heading as articulated robotic arm, kinematic and dynamic analysis of robotic systems, control system designs, visual inspection tasks as well as end application of robotic arms in various domestic, commercial as well as industrial task. The final focus is on visual inspection of surface defects and replacement activities.

## 2.1 Robotic manipulators in industries

Robotics has evolved as an wonderful technology and been increasingly popular in last few decades and has been extremely servicable in various environment. The end application of robots includes manufacturing, structural inspection, space research, defance, food inspection, health and medical application, agri disaster relief and all those workplace where human intervention is harzdous. So academic research in robotics has became very promising and enough literat has been reviewed as such Bogue [1] reviewed various application of serial robot (STAUBLI) in labarotaory automation as well as industry such as in phrmacutical industry medicine sorting, testing and packing, Toxicity testing of some drugs where human interaction is dangerous, DNA analysis in forensic lab, various environmental monitoring etc.

Henten et al.[2] presented the results of an inverse kinematics algorithm which has been used in a efficient model of a cucumber-harvesting robot. It is a P6R architect manipulator and inverse kinematic is used to acquire 3-D information from a real greenhouse. It checks if the cucumber is hanging within the reachable workspace to aid a collision free harvest and motion control of the manipulator.

Wang et al.[3] presented motion planning of a 12 degrees of freedom (DOF) remote handling robot used for inspecting the working state of the ITER-like vessel and

maintaining key device components and also the forward and inverse kinematics and work space and post space calculation of this manipulator are considered.

Regarding the inspection tasks with RH of ITER vessels, several works highlighted their design of articulated robots. An 8.2m long robot made-up of 5 modules with 11 actuated joints was proposed by Perrot et al. [4] Also, Gargiulo et al [5] developed an AIA (articulated inspection arm) aiming at in-vessel RH inspection task using a long reach along with a payload carrier. It has eight DOF and 8m long. It allowed a visual inspection of complex plasma facing components including limiters, neutralizers, RF antenna etc. It was planned to be integrated with Tore Supra Tokamak of France. Mutka et al. [6] described the control and 3-D visualization of a manipulator for welding inspection of nuclear reactor vessels. They defined graphical user interface providing all necessary tools needed for planning robot scan trajectories, using a virtual 3-D model and executed on a remotely operated robot. Houry et al. [7] developed a multipurpose carrier prototype (articulated inspection arm) for operating embedded diagnostic or tools into tokamak vacuum vessel. This arm with vision diagnostics is deployed inside the vacuum-vessel to monitor the state of plasma facing components. It has interchangeable diagnostic tools plugged on its front head. Houry et al. [8] explained the design of articulated inspection arm with an embedded camera and interchangeable tools at its head. Ribeiro et al. [9] illustrated an integrated view and results related to the blanket remote handling system, divertor remote handling system, transfer cask system as well as in-vessel viewing system and multipurpose deployment system. Peng et al. [10] presented inspection approach for tokamak vessel using a long reach multi-articulated manipulator and processing tool having modular design for the subsystem. It can reach the first wall area and can be folded in the horizontal port during plasma discharge periods. Kitamura et al. [11] developed a remote control system for maintenance of in-cell type fuel fabrication equipment, which display a 3-D information of the workspace from data obtained by laser range finder and conventional cameras. A manipulator was remotely operated and monitored using mock-up equipment. Peng et al. [12] in other work, presented the struct of a serial link robot with 8 degrees of freedom with a 3-axes wrist carrying camera. Kinematics and dynamic modeling was illustrated with the use of ADAMS software. Valcarcel et al. [13] explained the inspection of Beryllium limiter tiles (1278°C) of JET tokamak.

Pagala et al. [14] presented the inspection tasks like remote manipulation inside the hot cell using a modular robot system. Snoj et al. [15] described an approach of measurement using a calibrated  $\text{Cr}^{252}$  neutron source deployed by in-vessel remote handling boom and

mascot manipulator for JET vacuum vessel. Monochrome CCD cameras were used as image sensors.

Few very recent literature has also been studied and analysed, with some as Wang et al. [16] presented motion planning of a 12 DOFs remote handling robot used for inspecting the working state of the ITER-like vessel and maintaining key device components and also the forward and inverse work space and post space of this manipulator are considered. Chen et al. [17] has proposed improve remote maintenance algorithm for complete automatic full coverage scanning of the complex tokamak cavity. Two different trajectory planning methods they proposed are RS (rough scanning) and FS (fine scanning). Villedieu et al. [18] presents various upgrades made on the mechanics, the sensors, the electronics, the control station and the integration adaptation for the prototype of the Articulated Inspection Arm in vacuum tokamak.

## 2.2 Manipulator Configurations and Modeling

As shown in Figure 2.1, there are three revolute joints at the first three axes. The first revolute joint swings robot back and forth about vertical base, while second joint itches the arm up and down about horizontal shoulder axis and third joint pitches the forearm up and down about horizontal elbow axis.

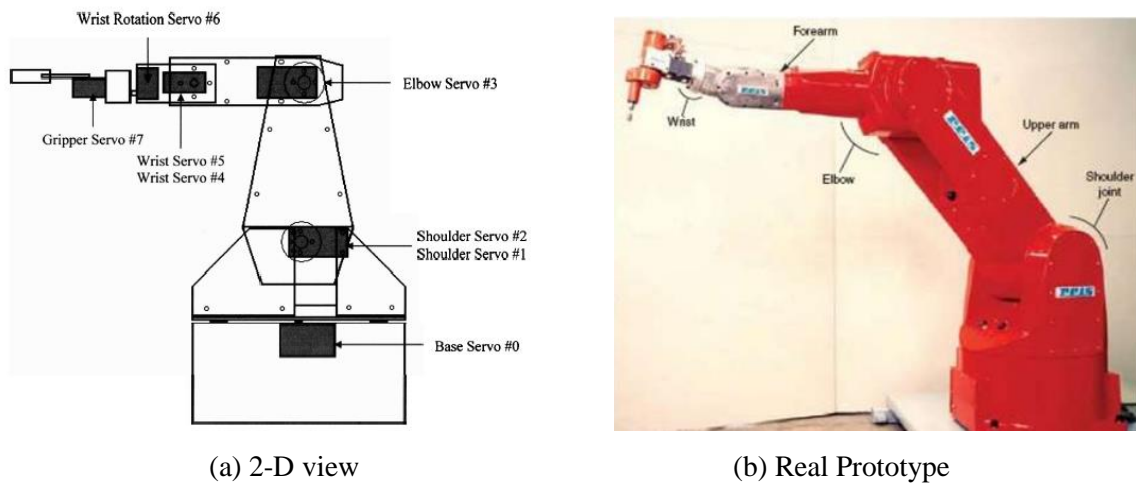


Figure 2.1 Articulated arm nomenclature [21]

Articulated robot configurations, permit the motions as in a human arm such as pick and place, assembling, straight-line tracking etc. Different articulated manipulator configurations were used in various applications.

Manzoor et al. [19] presented a robotic platform that has extensive potential in teaching robotics, control, electronics, image-processing and computer vision. They have

also verified the efficiency of their proposed platform by implementing various experiments of object sorting with accordance to various colour, shape and hardness, object grasping, trajectory generation, path planning, and also controller design.

Perrot et al. [20] reported a thorough research and development activities in advance articulated robotic manipulator (AIA) for inspection and intervention in various hazardous environment such as Tokamak fusion reactor. They have also presented the modelling (kinematic, architecture, control, vision camera), simulation and experimental results of field test.

Kaltsoukalas et al. [21] presented an analysis of an intelligent search algorithm for COMAU Smart5 Six, 6 DOF, Articulated Industrial Manipulator which defines the path to reach out the desired orientation and position of end-effector. They have performed various kinematic and dynamic analysis so as to perform various industrial activities smoothly.

Santolaria et al. [22] proposed a data capturing technique for identification of kinematic model parameters, using nominal data reached by a ball bar gauge based on a new approach including terms regarding measurement accuracy and repeatability. Figure 2.2 shows the arm used in their work.

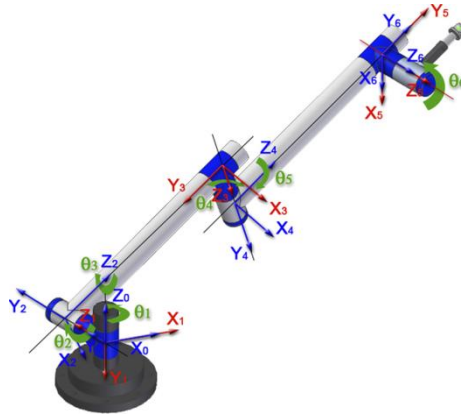


Figure 2.2 Sterling series FARO arm with 1.5 m long with 6-DOF [22]

Tung et al. [23] developed an image-guided 5-axis SCORBOT-ER VII coupled with two CCD camera robot for various crack detection, crack position and repair task inside fusion vessel. The paper also explains the kinematics and dynamics equation of motion and control architecture for better understanding. Image has been acquired from both the camera, compared and then for crack detection Sobel method has been used.

Villedieu et al. [18] developed a prototype of an Articulated Inspection Arm for vacuum tokamak inspection. The proposed prototype has 11 DOF translation and rotation

joints having pitch and yaw motion. The payload capacity is 10 kg. Details regarding the issues of mechanics, the sensor, the electronics, the control system have also been illustrated. Figure 2.3 shows the picture of the robot used in their work.

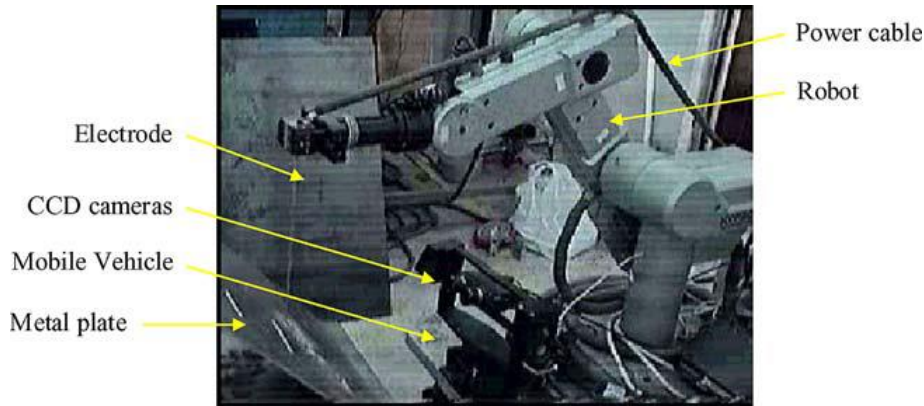


Figure 2.3 Five-axis SCORBOT-ER VII [18]

Soares et.al [24] came out with a multi-purpose rescue vehicle having an articulated arm with a mobile platform equipped with two different manipulators and different sets of end-effectors with a human-machine interface. They developed a human-machine interface which facilitates the controlled locomotion, receives information from all sensors. The three RGB cameras help to get the distance between end-effectors and the object, distance to the obstacles inside the vessel. The whole 3D model is designed using CATIA and simulated in a virtual reality environment. Figure 2.4 shows the robot arm employed in their work.

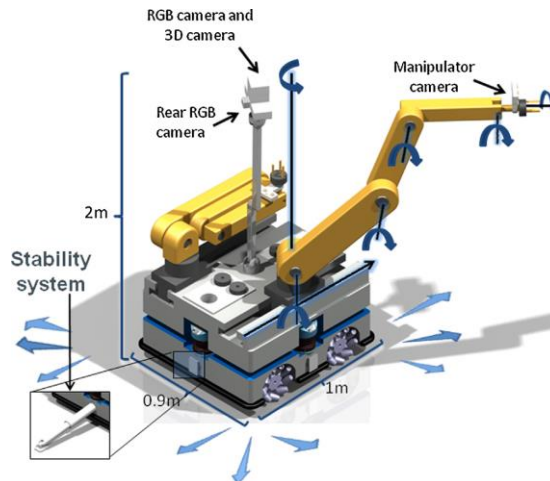


Figure 2.4 Five-axis articulated mobile manipulator with Human-machine Interface [24]

Chen and Cao [25] introduce a novel remote handling robot (RHR) for the maintenance of ITER\_D shaped vessel. The modular design has been divided into three parts: an omnidirectional transfer vehicle, a planar articulated arm, and a teleoperated manipulator with a total of 13 DOF and a payload capacity of 15 kg. With thorough kinematic analysis,

payload calculation, path planning, workspace simulation dynamic simulations good mobility and better performance has been proved. Figure 2.5 shows the robotic arm employed in this work.

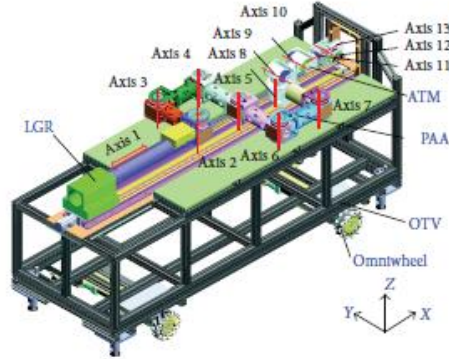
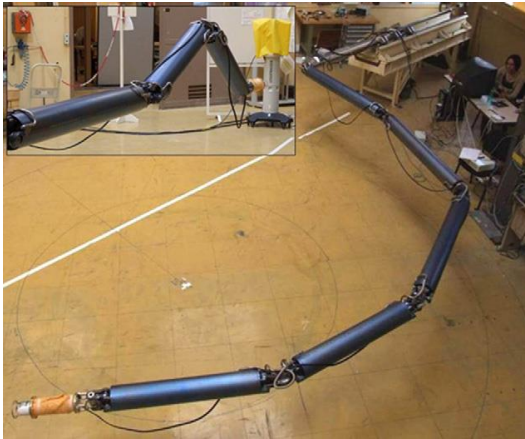
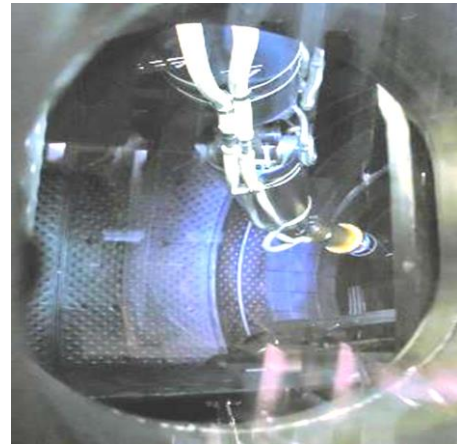


Figure 2.5 A 13-axis Remote Handling Robot [25]

Houry.et.al [8] developed a 8DOF multipurpose carrier prototype Articulated Arm Inspection for vision diagnostics and inspection of plasma facing components under high vacuum and temperature. The articulated inspection arm is 8 m long with rotational joints combination of pitch and yaw motions and a CCD camera at the front end. They have performed real time experiment during intervention between plasma sessions inside Tore Supra. Figure 2.6 shows the articulated robotic arm employed in their work.



(a) 8-axis AIA with vision system



(b) vision diagnostics deployment in Tore Supra

Figure 2.6 Inspection Robot [8]

## 2.3 Remote Control Architectures

Deploying the vehicle into the vessel and observing the wall surface using high speed camera system has become a common approach in several works. The dark surface of the



in-vessel is observed by some light source using a camera in either wireless or wired fashion through a remote computer or TV screen. Raimondi presented a review of remote handling device development for in-vessel maintenance tasks. Several other works stressed the remote control maintenance of the plasma facing components. Perrot et al. [20] presented the feasibility analysis of articulated inspection arm as a part of remote handling activities of ITER. Robbins et al. [26] illustrated the virtual reality system for visualization and tracking operations in remote handling. The system establishes a complete database for locating plant items and remote handling equipment during remote handling operation program. After carbon wall to ITER-like wall (Be/W/C) transition in 2010-11, neutron yield calibration was possible by using a remote handling boom and mascot manipulator inside JET vacuum vessel. Snoj et al. [15] developed a geometrical computational model of JET remote handling system and a script which helps in translating remote handling movement data to transformation of individual geometrical parts of the remote handling model. Vale et al. [27] described cask and plug remote handling system in ITER which transports heavy and highly activated in-vessel components between tokamak building and hot cell building. More recently, Maruyama et al. [28] emphasized the use of vision system for ITER-blanket RH activities. Dhanpal et al. [29] has given impetus to the design of remote devices for in-service inspection of vessels and pipes. They fabricated a mock-up 3-Dof scanner with a motion along z-axis. The camera is also mounted at the end-effector with a scan resolution of 380 lines for visual inspection of inner surface. Further multi-body dynamics simulation has been performed and the required torque required has been obtained.

## **2.4 Use of Vision Sensor in Robots and Image Processing**

Apart from kinematics and dynamics analysis of robot, many literatures are available where robotic arm is used for vision based control and vision based quality inspection system in various engineering discipline. Vision based 3-D position control and vision based tracking control of robotic arm are some of the example of the importance of vision in robotic system. In vision based tracking control a visual feedback is considered which takes in to account the uncertainties and disturbance of robot model as well as unknown camera parameter. For installation quality inspection in construction industry, vision assisted robotic arm is used. There are different such application in construction industry

such as bridge inspection, automated crack sealing, concrete surface grinding, sewer pipe inspection and tunnel inspection. [30-35]

Vision based sensing in robotic manipulators is an old concept. Camera placement and active vision based servo control have become regular activities. Korayem and Heldari [36] in 2009 presented a robot control with vision-based sensing of 6R robot. Direct and inverse kinematic equations, in addition to dynamic equations of robot were used in simulations. Image processing simulations and object recognition and pose estimation of end-effector and target object in 3-D space were shown. New methods of visual control of robots including neural networks were proposed in some paper. Sharan and Onwubolu [37] presented the software developments for a vision-based pick and place robot to provide computational intelligence required for its operation. Pinto et al. described an eye-on-hand approach, which executes a predefined path to produce grayscale images of the workstation. Based on feature recognition an object was identified by laser rangefinder. More recently, Bellandi et al. [38] applied the concept of combined 2-D and 3-D vision system in pick-and-place operations of 6-DOF robot with better accuracy. Here, the system was represented by using 2-D geometric template matching in order to classify 3-D objects. Nele et al. [39] proposed an image acquisition system for real time weld seam tracking using a vision camera and Lab VIEW interfacing. This sort of procedure is required in present work also. Larouche and Zhu [40] presented a framework for autonomous capture operation of a non-cooperative mobile target in 3-D workspace using robot using visual servoing with eye-in-hand set-up. Experimental work on 6-DOF manipulator was illustrated to estimate robustness during noisy conditions while doing vision capturing. Huang et al. [30] presents a 3-D position control for a robot arm using a pair of charge-coupled device (CCD) cameras, and vision geometry is utilized to measure the practical 3-D position of the robot arm's tip. Hua et al. [31] proposed a novel visual feedback control model that considers not only the uncertainties and disturbances in the robot model but also the unknown camera parameters and present a visual feedback control. Further by using various visual feed-back control design and outputs from camera robot was able to track the desired trajectory.

Many other research articles on the vision-based guidance and control of robotic manipulators are collected in this line to understand the on-going research in this area.

### **2.4.1. Crack detection using Image Processing**

Mohammad et al. [41] has made a through survey on the pattern recognition and image processing algorithms which is further been used to predict tile surface defects. They have taken six different types of defects, all having distinct morphologies and text and detection techniques has been divided in three subgroup: statistical pattern recognition, text classification, feat vector extraction. Some methods for image pre-processing for example filtering, edge detection, morphology, wavelet transform, contourlet transform are found to be most effective.

Laofor and Peansupap [42] describes in there paper the visual inspection of tiling work with innovative system of defect detection and quantification. The inspectors used is able to quantify the value of the defect and can predict all possible defects and the proposed prototype potential has been verified and compared with human inspection thus increasing its reliability.

Victores et al. [35]proposed a light weight robotic tool with control station at ground-level of a wheeled vehicle and vision sensor capable of particular inspection and maintenance. They prepared a graphical Human Machine Interface to identify fissure and cracks on composite surface of the tunnel's infrastructure

Yu et al. [43]proposed a robotic system for crack detection on concrete structure .the robot has a Charged Couple Device (CCD) camera and maintain a constant distance from the wall ,acquire the image. Further processing task has been done using edge detection (sobel method) to extract all feats of the crack like its measurement and coordinates. This paper was really helpful in our project work.

Choudhary and Dey [44] presented a paper concerning the crack detection in concrete surface. They explained how to extract feat from digital image, image processing using popular edge detection technique. They implemented two kind of approach one is the image approach which classifies an image as a whole, and the object approach which classifies each component or object in an image into cracks and noise. There proposed method have been verified on 205 images.

Moon and Kim [34] presented an automated crack detection system and divided into two parts image processing and image classification. They describe various methods like filtering, subtraction method and morphological operator which was useful in our project. The number of pixel and the ratio of the major axis to minor axis for connected pixels area extraction was innovative in their work.

Ekvall et al. [45] presented a 6-DOF vision based robot for object recognition and pose estimation using color co-occurrence histograms and geometric model. The process of object manipulation such as object detections, servoing to the object alignment and grasping has been explained.

## **2.5 Virtual Reality and mock-up models**

Recent practice of using virtual reality tools in remote handling operations of robotic inspection tasks was explored in several papers. Some of the collected literatures is given below in chronological order. Robbins et al. [26] described the aims, structure and use of a virtual reality system to JET fusion machine. For robots in welding for example, the application of augmented reality for orienting end-effectors was found to be very advantageous. Sibois et al. [46] proposed a RH programmed training the engines to support ITER project, using mock-ups. He et al. [47] emphasized the use of virtual prototype for kinematic analysis and numerical simulations of rigid flexible manipulators. Here, analysis forward and inverse kinematic approaches were very nicely proposed. A 3-D model was developed for 5-axis robot in UG NX software and was imported into ADAMS environment. Then the model material was specified and the constraints are added to each joint. Also joint drives were added before simulation. Maruyama et al. [28] proposed a blanket remote handling using robot powered devices with a vision camera. Here focus is given to vision system motion.

## **2.6 Theoretical/ practical control tasks**

From the literature beneath, it is seen that abundant work has been done in the area of control of robotics arm as well as disturbance rejection control of various serial link mechanisms. However, a very few works accounted the effect of joint friction and payload combinedly. Present work proposes three various control schemes with a disturbance rejection torque supplied by a disturbance observer (DO).

Mohammadi et al. [48] proposed control scheme and disturbance observer that assure global asymptotic position and disturbance tracking without any restrictions on the number of degrees of freedom (DOFs) and manipulator configurations. Chen et al. [49] came up with a new non-linear DO that overcome the disadvantage of existing DO by selection various design parameter and also it provides global exponential stability. Nikoobin et al. [50] proposed a non-linear DO using Lyapunov's direct method. Using

that nonlinear DO, the accurate dynamic model is not required to achieve high precision motion control because it makes the system robust against various internal disturbances. Liu and Peng. [51] proposed a state observer then corrects the disturbance estimation in a two-step design first, a Lyapunov-based feedback estimation law is used then estimation is then improved by using a feedforward correction term. Ginoya et al. [52] unifies the previously proposed linear and nonlinear disturbance observers in a general framework, which further does not required the exact dynamics of serial manipulator. They Presents a new design of multiple-surface sliding mode control for nonlinear uncertain systems with mismatched uncertainties and disturbances. Along with disturbance estimation it calculate the derivative of the virtual. Song et al. [53] presented a new approach combing CTC and Fuzzy Control (FC) for trajectory tracking problems of robotic manipulators with structured uncertainty and/or unstructured uncertainty. Another powerful nonlinear controller is a Sliding mode that is popularly used for the highly nonlinear system. The main reason to opt for this controller is its satisfactory control performance in wide range and solves two most challenging areas in control which is, stability and robustness. Piltan et al. [54] presented a review of the sliding mode control for the robotic manipulator and also explain the inclusion of fuzzy logic system to reduce or eliminate the disadvantages of classical sliding mode control. Rahmdel et al. [55] presented a comparison of both computed torque controller and sliding mode controller with disturbances for highly nonlinear dynamic PUMA robot manipulator in MATLAB/SIMULINK. Young et al. [56] presents a guide to sliding mode control for practicing control engineers which offers an accurate calculation to chattering phenomenon, catalogs implementable sliding mode control design solutions, and provides a frame of reference for future sliding mode control research. Ouyang et al. [57] proposed a new model free control law called PD with sliding mode control law or PD-SMC. In his proposed PD–SMC, PD control issued to stabilize the controlled system, while SMC is used to compensate the disturbance and uncertainty and reduce tracking errors dramatically. Piltan et al. [58] focused on the non-classical method like Fuzzy Logic in robust classical method like CTC and sliding mode control and also presented computed torque like controller with tunable gain. Kolhe et al. [59] worked in context of uncertainty and disturbance estimation (UDE) and designed a feed-back linearization controller-for trajectory tracking. Novelty presented is that, no accurate robot model is required about uncertainty and two-link manipulator result is verified with existing well know controllers.

Van and Kang [60] examined a novel adaptive quasi – continuous 2<sup>nd</sup> order sliding mode controller for uncertain robot manipulator (PUMA 560).The implemented scheme was found to be chatter-free, good accuracy in position tracking and quick stability and convergence.

## **2.7 Conclusion**

**This chapter highlighted the important research activities on robotic manipulator working in hostile environments in terms of their applications, configurations, kinematic and dynamic modelling, remote controlling abilities, vision sensing and image processing. Typical application of various controls schemes were also reviewed. The use of articulated configuration simplifies the objects handling with proper programming of joint motors. The review of past works indicated several valuable suggestions regarding the design and control of robots. Several authors employed the articulated arms for inspection tasks in real time applications. Modelling of manipulators has been attempted by several researchers for specific tasks. Advanced application of remote control and vision sensing has been found be the recent research areas in controlling the robots accurately. Theoretical control using various schemes including CTC, FC, and PD-SMC are being employed in the latest studies. Further virtual reality and mock-up studies are very much important before going for final prototyping. In this regard the software tools the ADAMS and other motion simulator are found to be very much useful.**

It is observed that robotic manipulators are used in important activities such as leakage detection, damage identification etc. Various manipulators have been employed and a wide scope for their use in pick and place and replacement activities was explored. It is observed from the literature review that there are still several open areas such as designing a control system of manipulator with an online visual data based on the status of the object surface, dynamic path planning and new concepts in pick and place activities.**A following issues are found to be the open areas in further study.**

- 1. A comparisons of kinematic studies of selected robotic vehicle from the rigid body motion analysis with that of flexible link considerations.**
- 2. A generalized dynamic model of the robot accounting unmodelled disturbances like joint friction, crack and damage on the links.**
- 3. A development of a model based controller for handling the disturbance using observer.**
- 4. The use of vision sensing for object condition monitoring and linking its outputs with the manipulator control for necessary actions.**
- 5. Considerations of the surrounding factor including temperature, pressure during the design and control of the manipulator.**

## Chapter 3

# Robot Modelling

Robot/manipulator is a vehicle or mechanism for implementing the proposed task. This chapter describes basic kinematics and dynamics of the manipulator system considered in the present work. Mathematical models for forward, inverse kinematics, workspace and singularity analysis of the manipulator are presented. **Manipulator considered for the tile replacement activities is a conventional articulated 5-axis robotic arm, because the application requires an operation similar to pick and place where the end-effector orientations plays a limited role and control task become easier. Further the dynamics of manipulator can also be simplified by treating the wrist of the manipulator to be in rigid posture.** It has a 3-axis arm and 2 degree of freedom wrist. At the end of wrist, an end-effector (2-finger gripper) is connected. Each link is actuated by its joint servomotor as per requirement. This configurations is selected due to its well-known kinematics and dynamics. Further, the manipulability can be enhanced by mounting it over a mobile base.

### 3.1 Kinematics

Kinematics is the study dealing with motion of system without accounting forces and inertia. It defines the position, velocity, acceleration and higher derivatives of the variables. The kinematic studies of robot manipulator are divided into two types: the first one is called forward kinematics and the second one is known as inverse kinematics. Forward kinematics determines the position and orientation (pose) of end-effector when all the joint angles are provided. On the other hand, inverse kinematics calculates the solutions of each joint variable corresponding to a specified end-effector pose in Cartesian space. Hence, forward kinematics is defined as transformation from joint space to Cartesian space where as inverse deals with transformation from Cartesian space to joint space. In industrial serial robots, inverse kinematics is a multi-solution problem.

#### 3.1.1 Forward Kinematics

Given a set of rigid bodies connected by joints, the post of this kinematic model is specified by the orientation of the joints. Consider a robot having  $n$  links numbered from

zero, starting from the base of the robot to the end-effector and the base is taken as link 0, all the joints are numbered from 1 to  $n$ . Denavit and Hartenberg (DH) in year 1955 proposed a systematic notation for assigning right-handed orthonormal coordinate frames, one to each link in an open kinematic chain of links. After assigning a reference ( $X_i, Y_i, Z_i$ ), a set of DH parameters describing the spatial relationships between a joint axis and its two neighbor joint axes are defined. Once these link-attached coordinate frames are assigned, transformation between adjacent coordinate frames can then be represented by a single standard  $4 \times 4$  homogenous coordinate transformation matrix.

By considering,

$d_i$  as the distance between  $Z_{i-1}$  and  $Z_i$  in the direction of  $X_{i-1}$ ,

$\alpha_i$  as the angle between  $Z_{i-1}$  and  $Z_i$  in the direction of  $X_{i-1}$ ,

$a_i$  (link length) as the distance between  $X_{i-1}$  and  $X_i$  in the direction of  $Z_i$

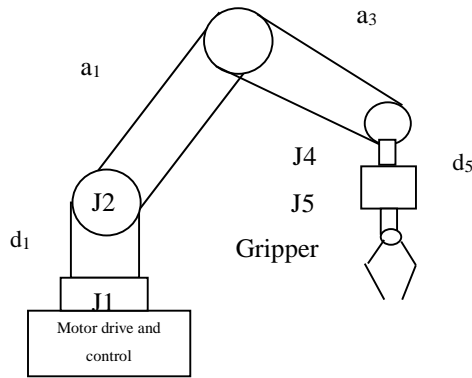
$\theta_i$  as the angle between  $X_{i-1}$  and  $X_i$  in the direction of  $Z_i$ ,

The resultant transformation between any two joints is given by the following link transformation matrix:

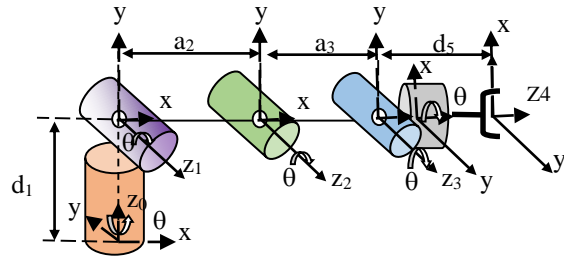
$${}^i_{i+1}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

All joint axes are coded from the first joint (connected to base) to the last joint (connected to gripper). The corresponding sets of DH parameters are obtained in a recursive way. The coordinate's frames for the present manipulator are assigned at each joint as shown in Figure 3.1





(a) Schematic Diagram



(b) Joint Frames

Figure 3.1 Link Coordinate frame of the present manipulator

Joint-1 represents the shoulder and its axis of motion is  $z_1$ . This joint is assigned with a rotational angle  $\theta_1$  (angular motion) around  $z_1$  axis in  $x_1y_1$  plane. Similarly joint 2 is the elbow and its axis is perpendicular to joint 1. It has a rotational angular motion of  $\theta_2$  around  $z_2$  axis. Joint 3 facilitates the tool pitching motion denoted as  $\theta_3$  and joint 4 facilitates tool roll motion along  $z_4$  axis which is perpendicular to joint 3. Joint 5 is at a vertical offset with joint 4 and provides an angular motion of  $\theta_5$  which is also identical to gripper rotation. The gripper sliding motion while picking is not considered as a degree of freedom of the manipulator.

All the parameters for the above manipulator are listed in Table 3.1 below, where  $\theta_i$  is angular rotation about  $z$  axis,  $a_i$  is the link length,  $d_i$  is the link offset along  $z$  axis and  $\alpha_i$  is link twist along the rotation about  $x$ -axis.

Table 3.1 D-H Link parameters of present manipulator

Joint	$\theta_i$	$a_i(\text{mm})$	$d_i(\text{mm})$	$\alpha_i$	Limits
1	$\theta_1$	0	$d_1$	$-90^\circ$	$140^\circ$ - $+140^\circ$
2	$\theta_2$	$a_2$	0	0	$90^\circ$ - $+90^\circ$
3	$\theta_3$	$a_3$	0	0	$90^\circ$ - $+90^\circ$
4	$\theta_4-90$	0	0	$-90^\circ$	$90^\circ$ - $+90^\circ$
5	$\theta_5$	0	$d_5$	0	$90^\circ$ - $+90^\circ$

On substituting the DH parameters, individual transformation matrices  ${}^0_1T, {}^1_2T, \dots$  are obtained as follows .

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & d_1 \\ 0 & 0 & 1 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^3_4T = \begin{bmatrix} s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c_4 & s_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^4_5T = \begin{bmatrix} c_5 & -s_5 & 0 & d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Overall the transformation matrix relating the end-effector coordinate frames with base frame is given as.

$$\begin{aligned} {}^0_5T &= {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T \\ &= \begin{bmatrix} c_1c_{234}c_5 + s_1s_5 & -c_1s_{234}s_5 + s_1c_5 & c_1c_{234} & c_1(d_5c_{234} + a_3c_{23} + a_2c_2) \\ s_1c_5c_{234} - c_1s_5 & -s_1s_{234}s_4 - c_1c_5 & s_1c_{234} & s_1(d_5c_{234} + a_4c_{23} + a_2c_2) \\ -c_{234}c_5 & c_{234}s_5 & -s_{234} & d_1 - a_2s_2 - a_3s_{23} - d_5s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.7)$$

Here,  $c_1, s_1, c_{23}$  and  $c_{234}$  are respectively  $\cos\theta_1, \sin\theta_1, \cos(\theta_2 + \theta_3)$  and  $\cos(\theta_2 + \theta_3 + \theta_4)$ . This product matrix is  $4 \times 4$  matrix with the three elements of last column represent position of end-effector (denoted as  $p_x, p_y$  and  $p_z$ ), while the remaining 9 elements indicate the end-effector orientation vectors. In other words, the end-effector is represented with its position and orientation as follows

$${}^0_5T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

where  $\{n_x, n_y, n_z\}, \{o_x, o_y, o_z\}, \{a_x, a_y, a_z\}$  represent the orientation vectors which together constitute rotation matrix while,  $(p_x, p_y, p_z)$  is the position of the end-effector. In practice, orientation is dictated by wrist, while position is specified by arm. Most of the time, position vectors are of interest and their simplified expressions are given as follows:

$$p_x = c_1(d_5 c_{234} + a_3 c_{23} + a_2 c_2) \quad (3.9)$$

$$p_y = s_1(d_5 c_{234} + a_3 c_{23} + a_2 c_2) \quad (3.10)$$

$$p_z = d_1 - a_2 s_2 - a_3 s_{23} - d_5 s_{234} \quad (3.11)$$

Also the orientation terms are given as follows:

$$n_x = c_1 c_{234} c_5 + s_1 s_5 \quad (3.12)$$

$$n_y = s_1 c_5 c_{234} - c_1 s_5 \quad (3.13)$$

$$n_z = -c_{234} c_5 \quad (3.14)$$

$$o_x = -c_1 s_{234} s_5 + s_1 c_5 \quad (3.15)$$

$$o_y = -s_1 s_{234} s_5 - c_1 c_5 \quad (3.16)$$

$$o_z = c_{234} s_5 \quad (3.17)$$

$$a_x = c_1 c_{234} \quad (3.18)$$

$$a_y = s_1 c_{234} \quad (3.19)$$

$$a_z = s_{234} \quad (3.20)$$

Here,

$$c_1 = \cos \theta \text{ and } s_1 = \sin \theta \quad (3.21)$$

$$s_{12} = s_1 c_2 + c_1 s_2 \quad (3.22)$$

$$c_{12} = c_1 c_2 - s_1 s_2 \quad (3.23)$$

$$c_{234} = c_2 [c_3 c_4 - s_3 s_4] - s_2 [s_3 c_4 + c_3 s_4] \quad (3.24)$$

$$s_{234} = s_2 [c_3 c_4 - s_3 s_4] + c_2 [s_3 c_4 + c_3 s_4] \quad (3.25)$$

### 3.1.2 Inverse Kinematics

The inverse kinematics analysis describes the angles of the joints for desired orientation and position of end-effector in Cartesian space. In other words, we have the Cartesian coordinates (x, y and z) and we would like to get the joint angle. The solution is more difficult and complex as compared to that of forward kinematics because there is no single solution to it. Solving the inverse kinematics problems in general for the spatial manipulator is an interesting area of research. The solution exists if the desired position and orientation is within the manipulator workspace. Inverse kinematics is important in controlling robot in Cartesian coordinate frame.

In inverse kinematics, tool configurations is required, which is described by orientation (first 3 x 3) and positions ( $p_x$ ,  $p_y$  and  $p_z$ ) of the end-effector with respect to the base frame. Here, as we are dealing with five axis manipulator to control its position & orientation, its inverse kinematics is required. Inverse kinematics results in multiple solutions each representing an assembly mode corresponding to the same end-effector configuration. The

problem can be solved by different methods: (i) analytic method (ii) algebraic method (iii) numerical method (iv) methods based on soft-computing tools like neural networks. Analytical method is used to solve the problem in the following stages:

Considering the following transformation matrix

$$({}^0_5T) = ({}^0_1T(\theta_1))({}^1_2T(\theta_2))({}^2_3T(\theta_3))({}^3_4T(\theta_4))({}^4_5T(\theta_5)) \quad (3.26)$$

Where  $\theta_i$  is the joint variable (revolute joints) for joint i. And in order to find the inverse kinematic solution, it has to be solved for  $\theta_i$  as a function of the known element of  ${}^{base}_{tool-tip}T$ . To solve for set of angles pre-multiply both the sides with  ${}^{i-1}_iT$  one by one. For example to find  $\theta_1$  as a function of the known elements, link transformation inverses are pre-multiplied as follows

$$\begin{aligned} [{}^0_1T(\theta_1)]^{-1}({}^0_5T) &= [{}^0_1T(\theta_1)]^{-1}({}^0_1T(\theta_1))({}^1_2T(\theta_2))({}^2_3T(\theta_3))({}^3_4T(\theta_4))({}^4_5T(\theta_5)) \\ &= ({}^1_2T(\theta_2))({}^2_3T(\theta_3))({}^3_4T(\theta_4))({}^4_5T(\theta_5)) \end{aligned} \quad (3.27)$$

Since  $[{}^0_1T(\theta_1)]^{-1}({}^0_1T(\theta_1)) = I$  is identity matrix. Terms on both the sides are equated and the unknown joint angle  $\theta_1$  is computed.

2. Likewise, the other joint variables are obtained from the following equations

$$[{}^0_1T(\theta_1){}^1_2T(\theta_2)]^{-1}{}^0_5T = {}^2_3T(\theta_3){}^3_4T(\theta_4){}^4_5T(\theta_5) \quad (3.28)$$

$$[{}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3)]^{-1}{}^0_5T = {}^3_4T(\theta_4){}^4_5T(\theta_5) \quad (3.29)$$

$$[{}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3){}^3_4T(\theta_4)]^{-1}{}^0_5T = {}^4_5T(\theta_5) \quad (3.30)$$

Alternatively a lower dimensional approach may also be used as follows

$$({}^2_3T)({}^3_4T) = ({}^1_2T)^{-1}({}^0_1T)^{-1}({}^0_5T)({}^4_5T)^{-1} \quad (3.31)$$

When, the end-effector pose ( $p_x$ ,  $p_y$ ,  $p_z$ ) along with normal, orientation and approach vector are given, the classical inverse solution for five axis articulated arm is given as follows:

$$\theta_1 = atan2(p_y, p_x) \quad (3.32)$$

To find the joint angle  $\theta_2$  the joints from 2 to 4 are considered as a 3-DOF planar manipulator moving in X-Z plane. If  $(p_x', p_z')$  is end-effector coordinates and  $(p_{xc}', p_{zc}')$  is wrist center coordinates such that

25

$$p_z = p_z - d_1, \quad (3.34)$$

$$p_{xc} = p'_x - d_5 \cos \theta_p^d, \quad (3.35)$$

$$p_{zc} = p'_z - d_5 \sin \theta_p^d \quad (3.36)$$

$$\theta_3 = \cos^{-1} \left[ \frac{p'^2_{xc} + p'^2_{zc} - a_2^2 - a_3^2}{2a_2a_3} \right] \quad (3.37)$$

$$\theta_2 = \text{atan2}(p'_z, p'_x) - \text{atan2}(a_3 \sin \theta_3, a_2 + a_3 \cos \theta_3) \quad (3.38)$$

$$\theta_4 = \theta_p^d - \theta_2 - \theta_3 \quad (3.39)$$

$$\theta_5 = \theta_w^d - \theta_1 \quad (3.40)$$

Here,  $p_x, p_y, p_z, \theta_w^d$  are specified as input data along with  $\theta_p^d = -90^\circ$  in present case.

### Numerical solution techniques

In order to achieve effective joint angles solving inverse kinematic solution is a must and this section utilizes numerical techniques to solve it. This approach is rather straightforward in that it utilizes the well –known and simple Newton's root finding technique. Further some minor complication are as such that we have five degree of freedom manipulator, and the desired configuration is specified as a homogeneous transformation which contains 12 element ,but the total number of degree of freedom is five. For a given set of initial values for Newton's method, the algorithm converges to only one solution when perhaps multiple solutions may exist.

### Newton's Method

Newton's method is finds a solution to the following system of equations

$$f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = n_x$$

$$f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = o_x$$

$$f_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = a_x$$

$$f_4(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = p_x$$

.....

$$f_{12}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = p_z$$

These can be written as

$$f(\theta) = a \text{ or } f(\theta) - a = 0 \quad (3.41)$$

Here,  $\theta$  are the vectors in  $R^5$  space. The method solves this equation for  $n^{\text{th}}$  iteration as follows:

$$\theta_{n+1} = \theta_n - (J(\theta_n))^{-1} f(\theta_n) \quad (3.42)$$

Where

$$J(\theta_n) = \left[ \frac{\partial f_i(\theta)}{\partial \theta_j} \right] = \begin{bmatrix} \frac{\partial f_1(\theta)}{\partial \theta_1} & \frac{\partial f_1(\theta)}{\partial \theta_2} & \dots & \frac{\partial f_1(\theta)}{\partial \theta_n} \\ \frac{\partial f_2(\theta)}{\partial \theta_1} & \frac{\partial f_2(\theta)}{\partial \theta_2} & \dots & \frac{\partial f_2(\theta)}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\theta)}{\partial \theta_1} & \frac{\partial f_n(\theta)}{\partial \theta_2} & \dots & \frac{\partial f_n(\theta)}{\partial \theta_n} \end{bmatrix} \quad (3.43)$$

This matrix is calculated either by differentiating the expressions for  $f_1, f_2 \dots f_{12}$  or by using numerical differentiation method such as forward or backward difference. The iterations are started with initial values of vector  $\{\theta\}$ . The convergence of the solution is verified by checking the norm of vector  $\{\Delta\theta\} = \{\theta_{n+1} - \theta_n\}$  with respect to the error tolerance specified.

### 3.1.3 Workspace Envelope

Workspace is the region or volume of space described by the reach of the end-effector, when all the manipulator joints execute all possible motions. Various factors affect the workspace such as the geometry of the manipulator, type of joints as well as its mechanical limits. For a solution to be real, the specified goal point must lie within the



limit of workspace. Further, the workspace has been divided into two categories i.e., dexterous workspace and reachable workspace. Dexterous workspace is that volume of space that the robot end-effector can reach with all orientation. Simply saying at every point in the dexterous workspace, the end-effector can be arbitrarily oriented. The reachable workspace on the other hand is that volume of space that robot can reach in at least one orientation. Thus, we can conclude that dexterous workspace is a subset of the reachable workspace. The reachable workspace is obtained as a 3-D plot between x, y and z position coordinates of the end-effector obtained from forward kinematics solution by varying the joint angles  $\theta_1$  to  $\theta_5$  from minimum to maximum values.

### 3.1.4 Jacobian Analysis

The differential motion of manipulator is also important when the manipulator has to follow a trajectory with different velocities. The relationship between end-effector velocity  $\dot{X}$  and the joint velocity vectors  $\dot{\theta}$  is given as Jacobian matrix [J]. It is defined as

$$\dot{X} = [J]\dot{\theta} \quad (3.44)$$

It can be used to identify the singularities within the workspace, understand the performance of the manipulator while tracking trajectories such as manipulability and stiffness. Singularity state refers to a situation where manipulator either loses or gains additional degrees of freedom. When manipulator loses the degrees of freedom which a common situation within the workspace, it is called Type1 or displacement singularity. It can be identified by using the determinant of Jacobian matrix. When determinant is equal to or close to zero; the point in the workspace or Cartesian space belongs to singular. There are other indices derived from Jacobian matrix, one such index is called manipulability index defined as follows:

$$(\theta_1, \theta_2 \dots \theta_5) = \det\left(\sqrt{[J]^T [J]}\right) \quad (3.45)$$

The kinematic manipulability index was introduced to evaluate the kinematic transmissibility of a robot manipulator as an index of the relationship between the linear and angular velocities at each joint and at the end-effector of the manipulator.

Also, singularities are represented with singularity index

In present case, the Jacobian matrix is defined in a simplified form as follows

$$[J] = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \frac{\partial p_x}{\partial \theta_3} & \frac{\partial p_x}{\partial \theta_4} & \frac{\partial p_x}{\partial \theta_5} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \frac{\partial p_y}{\partial \theta_3} & \frac{\partial p_y}{\partial \theta_4} & \frac{\partial p_y}{\partial \theta_5} \\ \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} & \frac{\partial p_z}{\partial \theta_3} & \frac{\partial p_z}{\partial \theta_4} & \frac{\partial p_z}{\partial \theta_5} \end{bmatrix} \quad (3.46)$$

Using the expressions for  $p_x$ ,  $p_y$ ,  $p_z$  obtained from forward kinematics various terms in the matrix [J] can be derived as follows.

$$J(1,1) = -s_1(a_3c_{23} + a_2c_2 + d_5c_{234}) \quad (3.47)$$

$$J(1,2) = -c_1(a_3s_{23} + a_2s_2 + d_5s_{234}) \quad (3.48)$$

$$J(1,3) = -c_1(a_3s_{23} + d_5s_{234}) \quad (3.49)$$

$$J(1,4) = c_1(-d_5s_{234}) \quad (3.50)$$

$$J(1,5) = 0 \quad (3.51)$$

$$J(2,1) = c_1(a_3c_{23} + a_2c_2 + d_5c_{234}) \quad (3.52)$$

$$J(2,2) = -s_1(a_3s_{23} + a_2s_2 + d_5s_{234}) \quad (3.53)$$

$$J(2,3) = -s_1(a_3s_{23} + d_5s_{234}) \quad (3.54)$$

$$J(2,4) = s_1(-d_5s_{234}) \quad (3.55)$$

$$J(2,5) = 0 \quad (3.56)$$

$$J(3,1) = 0 \quad (3.57)$$

$$J(3,2) = -d_5 c_{234} - a_2 c_2 - a_3 c_{23} \quad (3.58)$$

$$J(3,3) = -d_5 c_{234} - a_3 c_{23} \quad (3.59)$$

$$J(3,4) = -d_5 c_{234} \quad (3.60)$$

$$J(3,4) = 0 \quad (3.61)$$

Singularities are often computed from the determinant of Jacobian matrix. When the determinant of the Jacobian is equal to zero, it means that the manipulator approaches singularities. However, the actual value of the determinant cannot be used as a practical measurement of the degree of ill-conditioning. For this purpose, it is convenient to use the condition number of the Jacobian. It is defined as:

$$(J) = \|J^{-1}\| \|J\| \quad (3.62)$$

The inverse of the condition number  $1/\kappa$ , having a value in between 0 and 1 is often used.

## 3.2 Dynamics

Deriving the dynamic equation for the manipulator is quite an essential task for number of factors such as analysis of manipulator structure, designing of control architectures, study of motion simulation and trajectory tracking. Dynamics analysis is further helpful for designing the mechanical prototype and computation of various forces and torque required for executions of motion. A variety of methods are available to compute manipulator dynamics such as Lagrangian formulation, D'Alembert Principle, Kane equation, recursive Newton-Euler, Euler-Lagrange method. The objective in this section is to derive the equations of motion using the Euler-Lagrange formulation as it is easy to implement in the form of computer code. Manipulator dynamics is basically the study of motion such as displacement of end-effector, joint velocity and acceleration when certain joint forces are acting. There are two kinds of problems in dynamics. First one is forward dynamics which computes the motion of manipulator from the given actuator forces and torques. Inverse dynamics is a process of computing the joint actuation forces or torques from the given position of end-effector, velocity and acceleration. We use forward dynamics as a model of the manipulator, while inverse dynamics is used to validate dynamic model or

sometimes in control. Forward dynamic equations are solved by time-integration methods such as Runge-Kutta or Euler methods. For an n-link manipulator, the dynamic equations are generalized as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau \quad (3.63)$$

where

$\theta \in \mathbb{R}^n$  represents joint position vector,

$\dot{\theta} \in \mathbb{R}^n$  represents joint velocities,

$\ddot{\theta} \in \mathbb{R}^n$  represents joint accelerations,

$\tau \in \mathbb{R}^n$  is the joint torque vector, which is often function of time.

$M(\theta) \in \mathbb{R}^{n \times n}$  is symmetric positively definite inertia matrix,

$C(\theta, \dot{\theta}) \in \mathbb{R}^n$  is the vector of Coriolis and centrifugal forces,

$G(\theta) \in \mathbb{R}^n$  is vector representing gravity.

Detailed expressions for all the terms are given in Appendix-A.

### 3.3 Numerical studies

The proposed 5 degree of freedom robotic arm is simulated in computer for understanding various issues. The link lengths and off-sets used in present analysis are given below;

$$d_1=50 \text{ mm}, a_2=140 \text{ mm}, a_3=120 \text{ mm}, d_5=100 \text{ mm}.$$

Care is taken to maintain the same dimensions in the fabricated prototype. Due to limitation of servos at the joints, the following joint limits are considered:

$$\theta_1 \in [-140^\circ, +140^\circ]$$

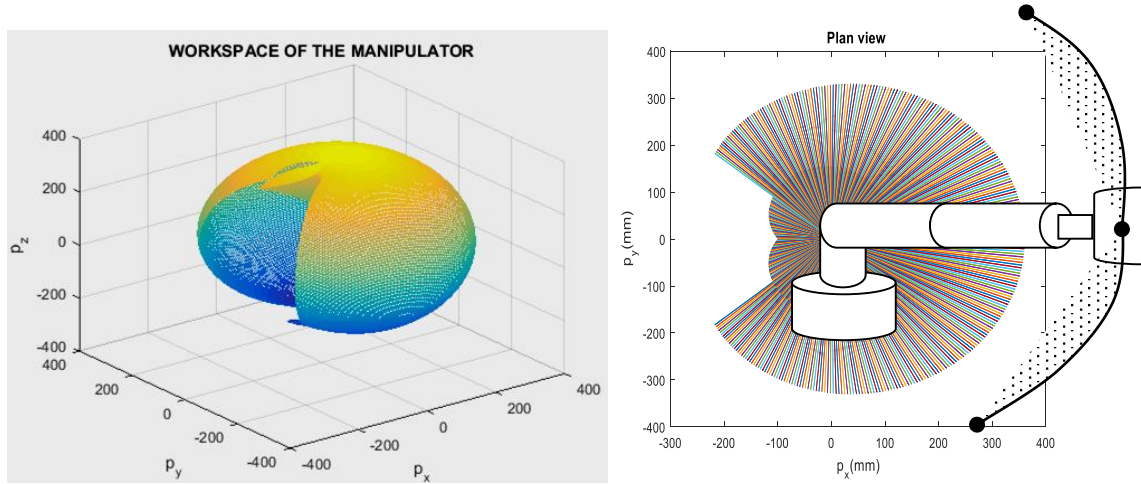
$$\theta_2 \in [-90^\circ, +90^\circ]$$

$$\theta_3 \in [-90^\circ, +90^\circ]$$

$$\theta_4 \in [-90^0 - +90^0]$$

$$\theta_5 \in [-90^0 - +90^0]$$

Initially the reachable workspace of the linkage is studied. Fig.3.3 shows the hemispherical workspace of present articulated manipulator.

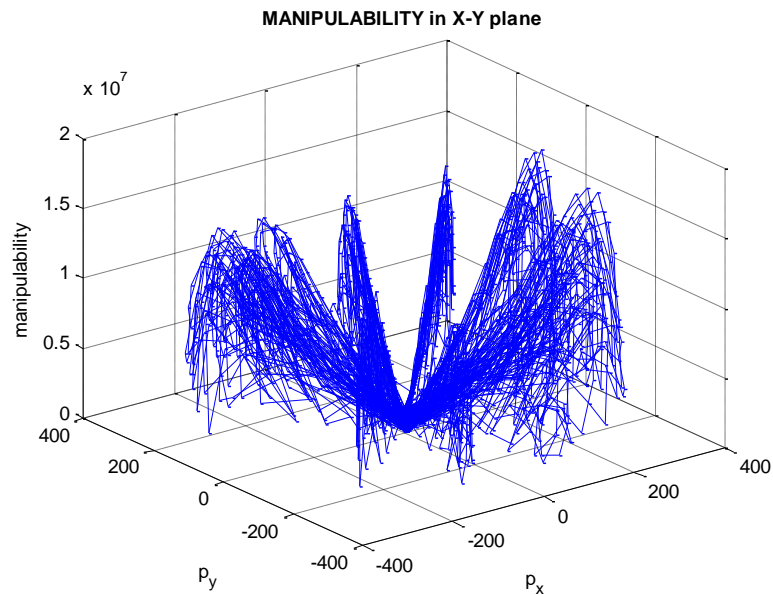


(a) Reachable Workspace

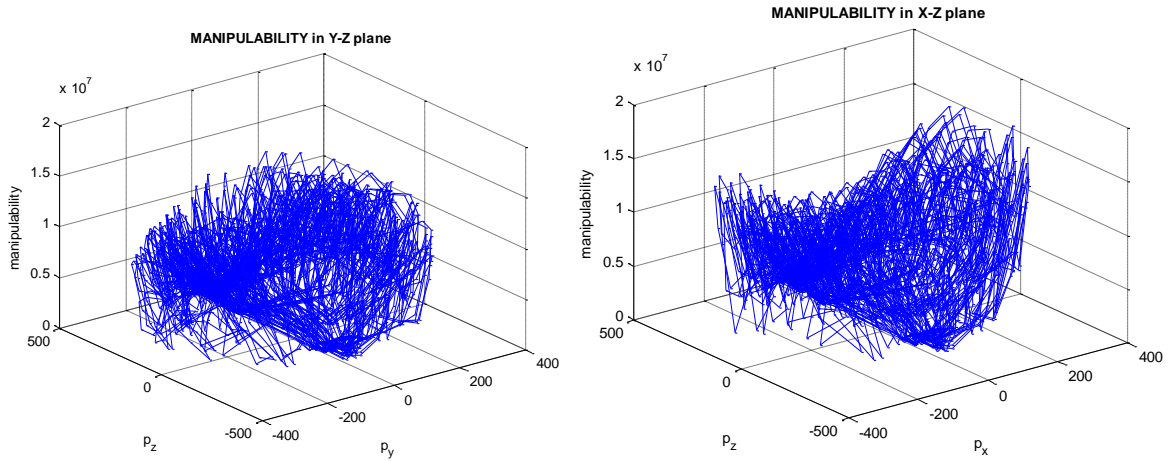
(b) Two-dimensional configuration

Figure 3.3 Workspace Computations

It is obvious that the manipulator has a workspace of radius equal to  $a_2+a_3+d_5=380$  mm. As Jacobian matrix varies from point to point in the workspace, the manipulability index and dexterity index also vary. Figure 3.4 shows the manipulability index.



(a) Manipulability index in X-Y plane

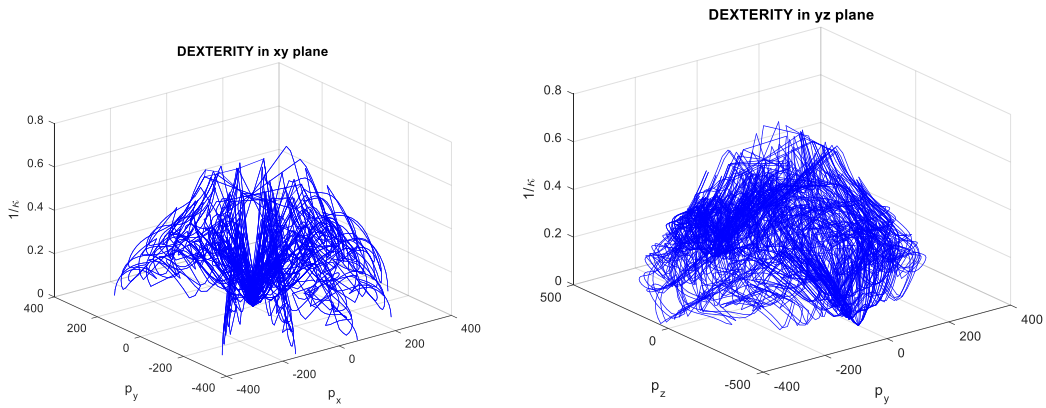


(b) Manipulability index in Y-Z plane

(c) Manipulability index in X-Z plane

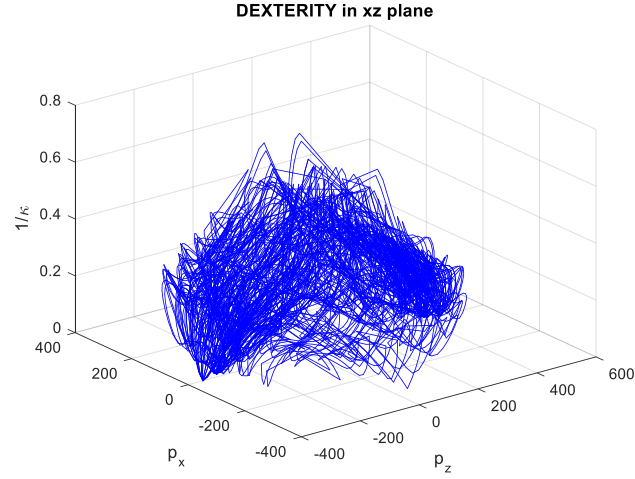
Figure 3.4 Manipulability index within the work envelope

Likewise, the dexterity index which is inverse of condition number of Jacobian matrix defines the ability of the manipulator to cross the singularities. Figure 3.5 show the dexterity index within the workspace in three planes. If this index is zero, it is a state of singularity. It can be seen that dexterity is always less than 0.8 and in some locations it is also zero indicating the singular regions within the workspace.



(a) Dexterity in x-y plane

(b) Dexterity in y-z plane



(c) Dexterity in x-z plane

Figure 3.5 Variation of dexterity index within the workspace

### 3.3.1 Inverse Kinematics

To illustrate inverse kinematics of present manipulator, a 3-D curved trajectory is considered. It is defined by the following parametric representation:

$$\left. \begin{aligned} p_x &= (6t - 3) \cos(2(6t - 3)) \\ p_y &= \sin(3(6t - 3)) \\ p_z &= \cos(3 - 6t) \end{aligned} \right\} t \in [0, 1];$$

$$\theta_w^d = \pi/6.$$

Figure 3.6 shows the Cartesian space trajectory as a 3-D plot.

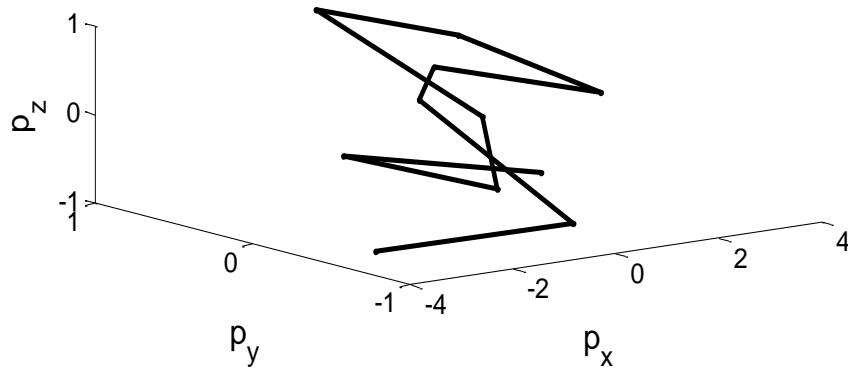
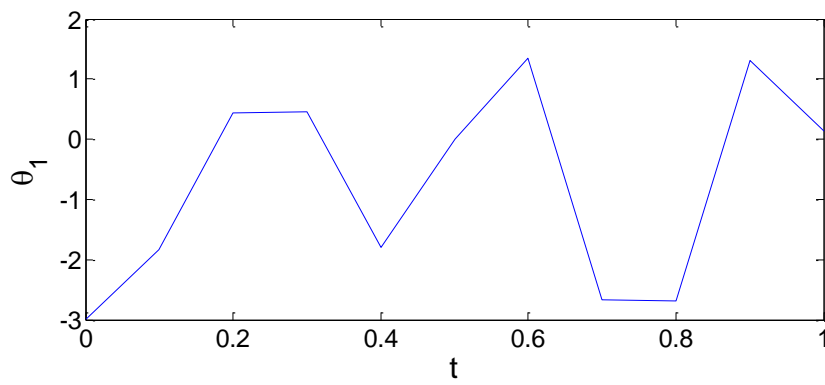
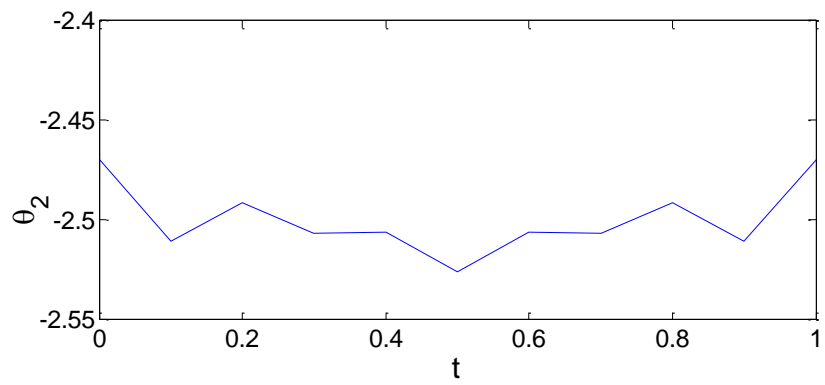


Figure 3.6 An input trajectory considered

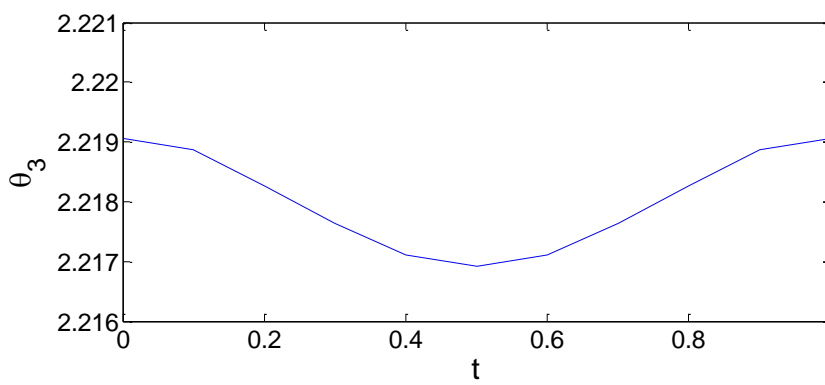
The corresponding inverse kinematic solution is shown in Figure 3.7



(a) theta 1

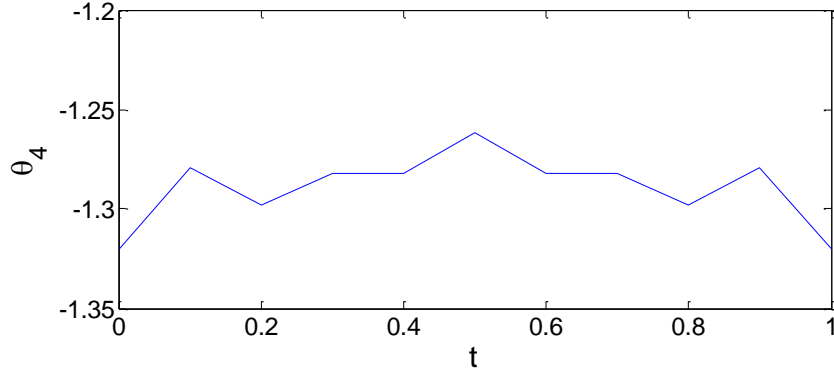


(b) theta 2

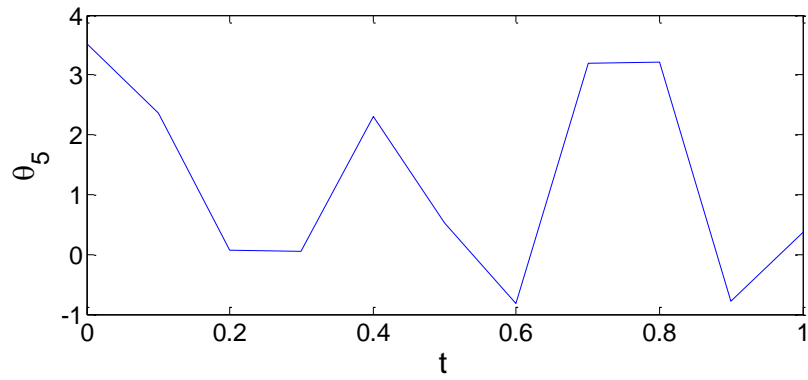


(c) theta 3





(d) theta 4



(e) theta 5

Figure 3.7 Inverse kinematic solution using geometric method

### 3.3.2 Inverse Dynamics

Next, for the considered manipulator arm (first 3 degrees of freedom), the dynamic equations are formulated and forward/inverse dynamic solutions are shown.

Inertial matrix derived from Euler-Lagrangian method is given below:

$$M(\theta) = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix}$$

Where,

$$m_{11} = (m_4 + m_3 + 0.25m_2)a_2^2 \cos^2 \theta_2 + (m_4 + 0.25m_3)a_3^2 \sin^2(\theta_2 + \theta_3)$$

$$+ (m_4 + m_3)a_2a_3 \cos\theta_2 \sin(\theta_2 + \theta_3) + 1$$

$$m_{22} = (m_4 + m_3 + 0.25m_2)a_2^2 + (m_4 + 0.25m_3)a_3^2 + (2m_4 + m_3)a_2a_3 \sin\theta_2 + 1$$

$$m_{23} = (m_4 + m_3 + 0.25m_2)a_3^2 + (m_4 + 0.5m_3)a_2a_3 \sin\theta_2 + 1$$

$$m_{32} = m_{23}$$

$$m_{33} = (m_4 + 0.25m_3)a_3^2 + 1$$

Here,  $m_1$ ,  $m_2$ ,  $m_3$  and  $m_4$  are respectively, masses of first, second, third link and end effector and load masses.

The terms in Coriolis and centrifugal vector is given as:

$$\begin{aligned} c_1 = & \dot{\theta}_1 \dot{\theta}_2 (m_4 + 0.25m_3)a_3^2 \sin(2(\theta_2 + \theta_3)) - (m_4 + m_3 + 0.25m_2)a_2^2 \sin 2\theta_2 \\ & + \dot{\theta}_1 \dot{\theta}_3 [(m_4 + 0.25m_3)a_3^2 \sin(2(\theta_2 + \theta_3)) + (m_4 + m_3)a_2a_3 \cos\theta_2 \cos(\theta_2 + \theta_3)] \\ & + (m_4 + m_3)a_2a_3 \cos(2(\theta_2 + \theta_3)) \end{aligned}$$

$$\begin{aligned} c_2 = & \dot{\theta}_2 \dot{\theta}_3 (-(2m_4 + m_3)a_2a_3 \cos\theta_3) + \theta_3^2 (-(m_4 + 0.5m_3)a_2a_3 \cos\theta_3) \\ & + (m_4 + 0.25m_3)a_3^2 \sin(2(\theta_2 + \theta_3)) + (m_4 + 0.5m_3)a_2a_3 \cos(2(\theta_2 + \theta_3)) \\ & + 0.5\dot{\theta}_1^2 ((m_4 + m_3 + 0.25m_2)a_2^2 \sin(2\theta_2)) \end{aligned}$$

$$\begin{aligned} c_3 = & 0.5\dot{\theta}_1^2 [(m_4 + 0.25m_3)a_3^2 \sin(2(\theta_2 + \theta_3)) + (m_4 + m_3)a_2a_3 \cos\theta_2 \cos(\theta_2 + \theta_3)] \\ & - (0.5\dot{\theta}_2^2 + \dot{\theta}_2 \dot{\theta}_3) (2m_4 + m_3)a_2a_3 \cos\theta_3 \end{aligned}$$

Terms in gravitational vector are given as follows:

$$g_1 = 0$$

$$g_2 = (m_4 + m_3 + 0.5m_2)ga_2 \cos\theta_2 + (m_4 + 0.5m_3)ga_3 \sin(\theta_2 + \theta_3)$$

$$g_3 = (m_4 + 0.5m_3)ga_3 \sin(\theta_2 + \theta_3)$$

Table 3.2 shows the dimensional parameters of the robotic manipulator for numerical simulation of dynamic analysis.[61]

Table 3.2 Dimensions and mass properties of Arm

Parameter	Value
$d_1$ (length of first link)	50 mm
$a_2$ (length of second link)	140 mm
$a_3$ (length of third link)	120 mm
$m_1$ (mass of first link)	0.2 kg
$m_2$ (mass of second link)	0.2 kg
$m_3$ (mass of third link)	0.15 kg
$m_4$ (end-effector + load mass)	0.5 kg

Based on the above calculated inertia matrix, Coriolis and gravity vectors, joint torques were evaluated to verify the computations using a set of pre-defined trajectories given below [60] :

$$\theta_1 = \cos\left(\frac{t}{5\pi}\right) - 1$$

$$\theta_2 = \cos\left(\frac{t}{5\pi} + \frac{\pi}{2}\right)$$

$$\theta_3 = \sin\left(\frac{t}{5\pi} + \frac{\pi}{2}\right) - 1, \quad t \in [0, 10]$$

Figure 3.8 shows these joint trajectories as a function of time.

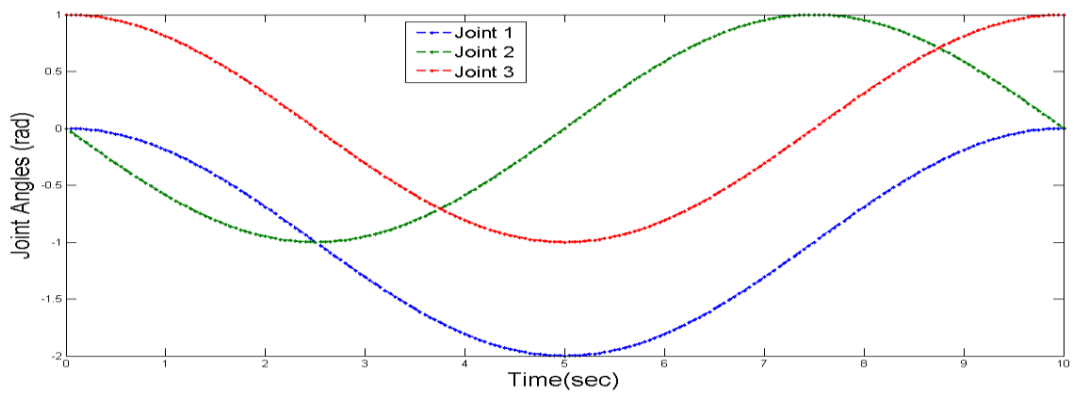


Figure 3.8 Joint trajectory considered

The required joint torques to execute these trajectories is shown in Figure 3.9. As seen from the figure, the joint torque 2 is highest throughout the trajectory.

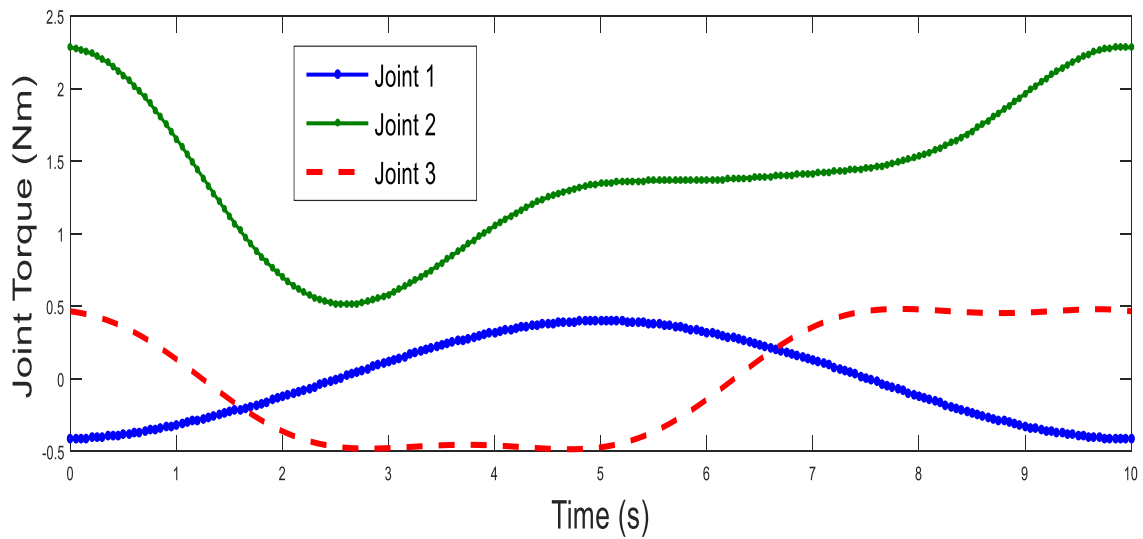
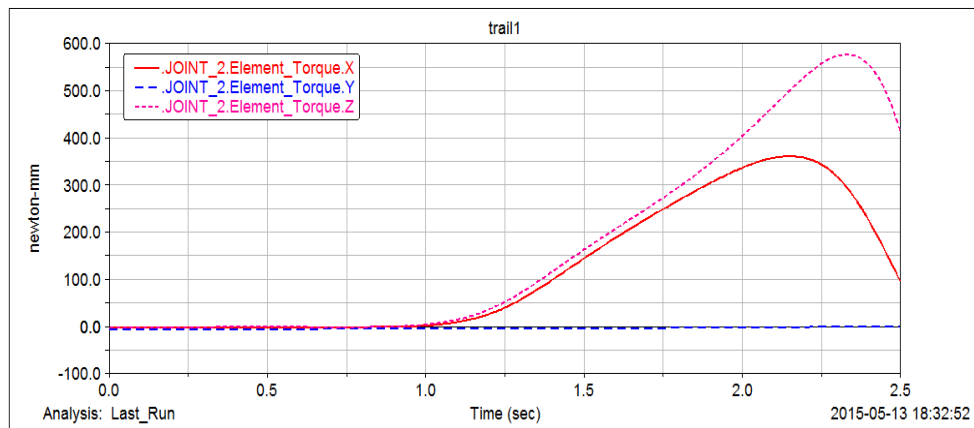


Figure 3.9 Required joint torques

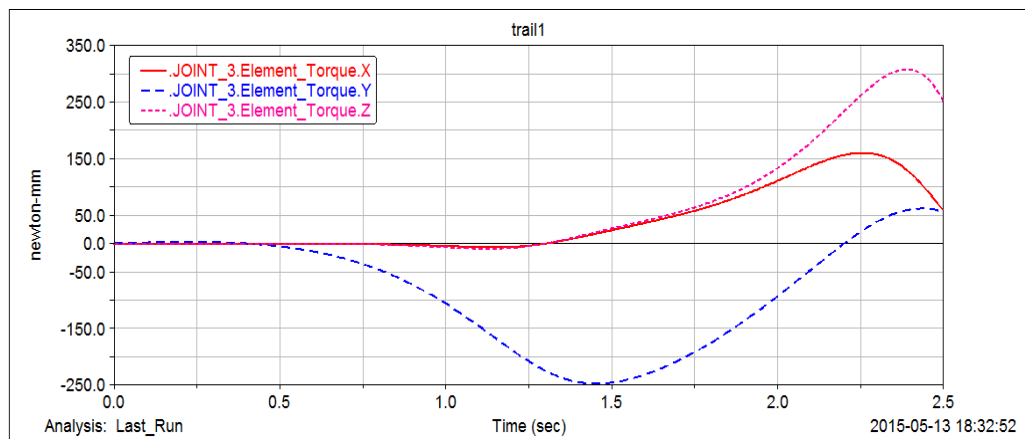
This illustrates the inverse dynamics from joint angles and its first and second derivatives. Another case study is conducted with a solid model of the manipulator with 2 degree of freedom wrist developed in SolidWorks and analyzed using ADAMS simulation tool. ADAMS (Automatic Dynamic Analysis of Mechanical Systems) simulation tool is a powerful modeling and simulating environment that allows building, simulating, refining, and ultimately optimizing any mechanical system. It also helps to understand the virtual prototyping process by improving the design of the mechanism. ADAMS has a basic module called VIEW where the model can be constructed or may be imported from SolidWorks software. The 2-D view of the model in ADAMS environment is shown in Figure 3.10 and the constraints are placed appropriately to simulate the motion of the mechanism and find out the joint torques to achieve a desired trajectory of end-effector.



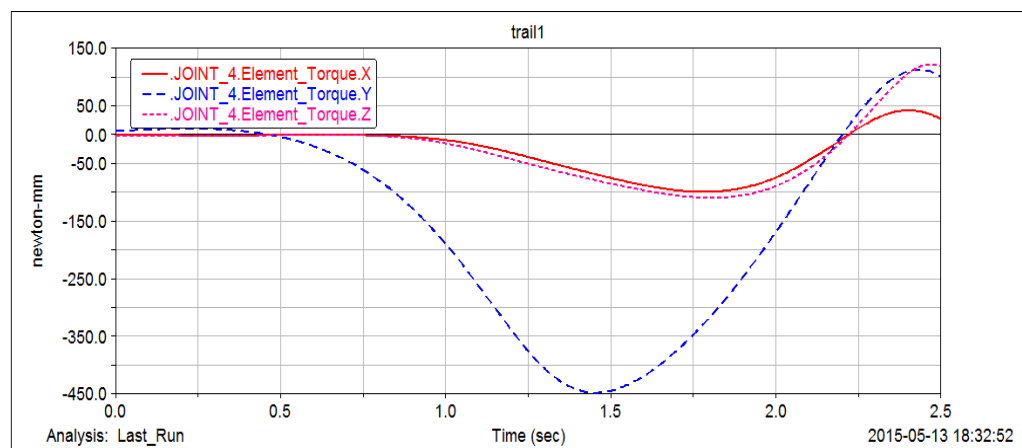
(a) Joint 1 Torque



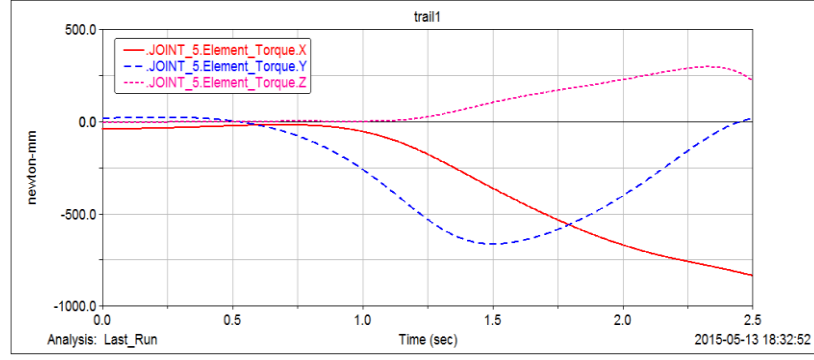
(b) Joint 2 Torque



(c) Joint 3 Torque



(d) Joint 4 Torque



(e) Joint 5 Torque

Figure 3.11 Joint torques

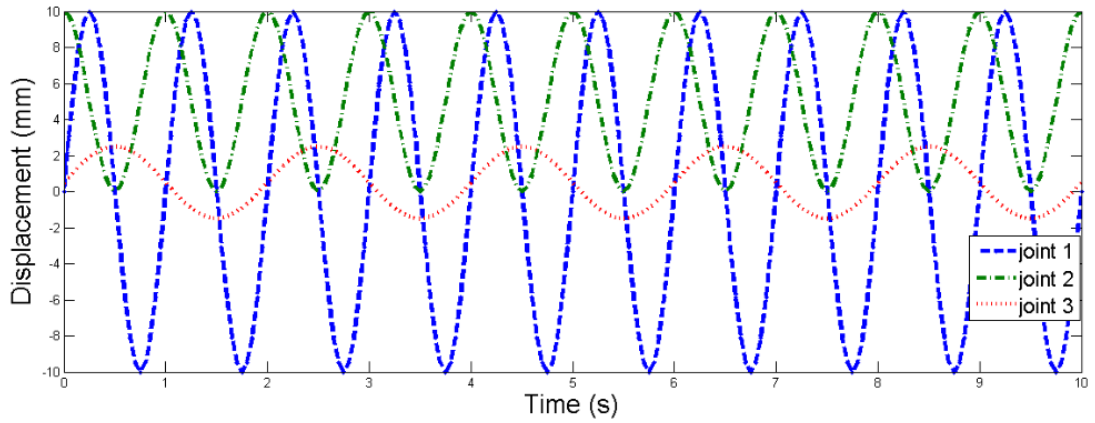
At each joint, a single torque component in fact corresponds to the motor torque.

### 3.3.3 Forward Dynamics

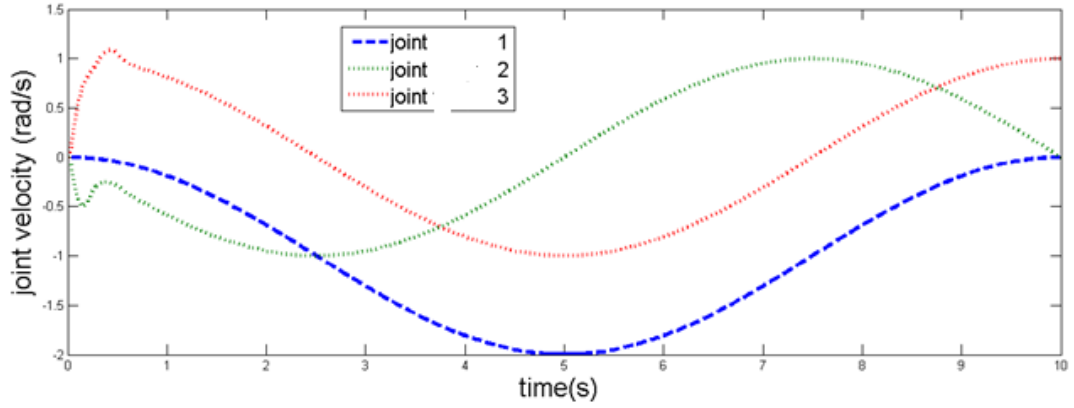
A forward dynamics takes joint torques as input and solves the differential equation using initial conditions. A program is developed in Matlab and it obtains the displacements and velocity at all the three joints of the arm. The input torque equations used are described below:

$$\begin{aligned}\tau_1 &= 10\sin(2\pi t) \\ \tau_2 &= 5 + 5\cos(2\pi t) \\ \tau_3 &= 0.5 + 2\sin(\pi t), t \in [0,10]\end{aligned}$$

Figure 3.12 shows the output displacements and velocities at the three joints obtained on solving the differential equations.



(a) The corresponding joint displacement



(b) Joint Velocities

Figure 3.12 Outputs of forward dynamics

The differential equations are solved by Runge-Kutta 4<sup>th</sup> order numerical method with a built in function ode45 available in Matlab.

### 3.4 Conclusion

This chapter presented some kinematic and dynamic characteristics of the proposed manipulator used in inspection task. It is observed that the manipulator has enough workspace to move in horizontal or vertical planes within the reach of end-effector. Also, it has limited singularity regions. Further, the dynamic torque capability is within the limits of the joint motors commonly available for driving the manipulator. All these basic modules are developed as separate functions so as to use them in controller design.



## Chapter 4

# Model-Based Control Studies

### 4.1 Introduction

After studying the kinematic and dynamics of manipulator, controllability of the manipulator is required to perform a desired trajectory tracking task. Use of nonlinear-control techniques is valid only when the system under study is modeled mathematically by nonlinear differential equations and dynamic equations are properly represented by it. We command the manipulator joints to follow prescribed position trajectories, but on other hand, the actuators are commanded in terms of torques, so the essence of the control system is to compute appropriate actuator commands that will realize this desired motion. The robot receives a vector of joint torques  $\tau$  from the control system. The manipulator sensors are allowed to read the vectors of joint position and joint velocities  $\dot{\theta}$ . It is also given a joint trajectory  $\theta_d$  which is to be tracked. For better convenience, it is assumed that  $\theta_d$  is specified for all time and it is at least twice differentiable. Unfortunately, imperfections in the dynamic model and the inevitable presence of disturbances make such a control system impractical for use in real applications. So, the disturbance observers are employed in control circuits which may produce compensation torque to the system in a proper way. This chapter presents two common control schemes: computed torque control (CTC) and PD-Sliding mode control (PD-SMC) applied to the present manipulator.

### 4.2 Computed Torque Controller (CTC)

CTC was one of the first model-based motion control approach for nonlinear system. The accurate dynamics of the model is very difficult to achieve and moreover is changes considerably when the payload is high. This results in degradation of trajectory tracking performance. CTC is a desirable controller in this case because of non-linear nature (function of joint position and velocity), the feedback loop helps in linearizing the robot dynamics. It cancels all the nonlinearities and applies exactly the torque needed to overcome the inertia of the actuators [64, 65].

$$\tau = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta}) + G(\theta) \quad (4.1)$$

And substituting this control law into the dynamic equation of the manipulator, we see that

$$M(\theta)\ddot{\theta} = M(\theta)\ddot{\theta}_d \quad (4.2)$$

And since  $M(\theta)$  is uniformly positive definite in  $\theta$ , we have,

$$\ddot{\theta} = \ddot{\theta}_d \quad (4.3)$$

Therefore, if the manipulator initial position and velocity counterparts the desired ones, the manipulator performs well as per the desired trajectory. Further the tracking properties of the control law can be improved by an addition of a state feedback. The linearity of equation suggests the following control law:

$$\tau = M(\theta)[\ddot{\theta}_d - k_v \dot{e} - k_p e] + C(\theta, \dot{\theta}) + G(\theta, \dot{\theta}) \quad (4.4)$$

Where  $e = \theta - \theta_d$  and  $\dot{e} = \dot{\theta} - \dot{\theta}_d$  are the instantaneous errors in joint position and velocity respectively. And  $k_p$  and  $k_v$  are the position and velocity gain matrices. The error dynamics can be written as:

$$M(\theta)(\ddot{e} + k_v \dot{e} + k_p e) = 0 \quad (4.5)$$

Since  $M(\theta)$  is always positive definite, we have

$$\ddot{e} + k_v \dot{e} + k_p e = 0 \quad (4.6)$$

Figure 4.1 shows the block diagram of CTC control scheme.

The computed torque control law consists of two components which can be written in equation as:

$$\tau = \underbrace{M(\theta)\ddot{\theta}_d + C\dot{\theta} + G}_{\tau_{ff}} + \underbrace{M(\theta)(-k_v \dot{e} - k_p e)}_{\tau_{fb}} \quad (4.7)$$

The term  $\tau_{ff}$  is the feed-forward components. It provides the amount of torque necessary to drive the system along the desired path. The term  $\tau_{fb}$  is the feedback component, which delivers corrected torque to reduce any error in the trajectory of the manipulator.

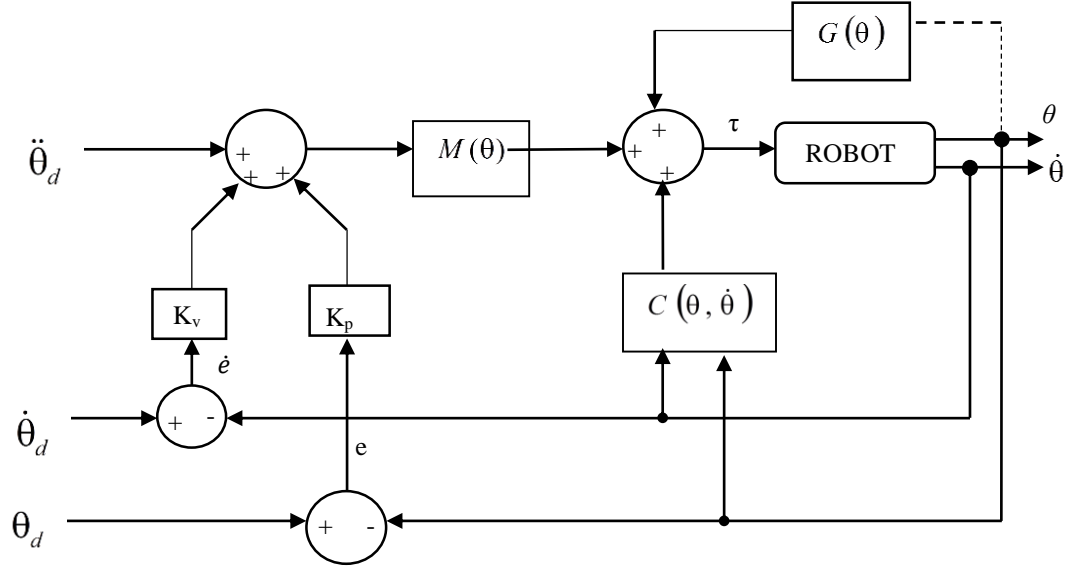


Figure 4.1 Block –Diagram of Computed Torque Control

Since the error equation is linear, it is easy to choose  $k_v$  and  $k_p$  so that the overall system is stable and  $e \rightarrow 0$  exponentially as  $t \rightarrow \infty$ . However, we can freely choose  $k_v$  and  $k_p$  such that an independent exponentially stable system is obtained.

### Stability of the Computed Torque Law

If  $k_p, k_v \in \mathcal{R}^{n \times n}$  are positive definite, symmetric diagonal matrices, then the control law applied to the system results in exponential trajectory tracking.

*Proof.* The error dynamics can be written as a first-order linear system:

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_p & -k_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (4.8)$$

It suffices to show that each of the eigen values of A has a negative real part. Let  $\lambda \in \mathcal{H}$  be an eigen values of A with corresponding eigenvector  $v = (v_1, v_2) \in \mathcal{H}^{2n}, v \neq 0$ . Then,

$$\lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_p & -k_v \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_2 \\ -k_p v_1 - k_v v_2 \end{bmatrix} \quad (4.9)$$

It follows that if  $\lambda = 0$  then  $v = 0$ , and hence  $\lambda = 0$  is not an eigenvalue of A. Further, if  $\lambda \neq 0$ , then  $v_2 = 0$  implies that  $v_1 = 0$ . Thus  $v_1, v_2 \neq 0$  and we may assume without loss of generality that  $\|v_1\| = 1$ . Using this, we write

$$\begin{aligned}\lambda^2 &= \mathbf{v}_1^* \lambda^2 \mathbf{v}_1 = \mathbf{v}_1^* \lambda \mathbf{v}_2 \\ &= \mathbf{v}_1^* (-k_p \mathbf{v}_1 - k_v \mathbf{v}_2) = -\mathbf{v}_1^* k_p \mathbf{v}_1 - \lambda_1 \mathbf{v}_1^* k_v \mathbf{v}_1,\end{aligned}$$

Where  $*$  denotes complex conjugate transpose. Since  $\alpha = \mathbf{v}_1^* k_p \mathbf{v}_1 > 0$  and  $\beta = \mathbf{v}_1^* k_v \mathbf{v}_1 > 0$ , we have

$$\lambda^2 + \alpha\lambda + \beta = 0, \quad \forall \quad \alpha, \beta > 0 \quad (4.10)$$

And hence the real part of  $\lambda$  is negative.

The advantage of CTC law is that it converts a nonlinear dynamical system into a linear system, and this further helps in the use of any number of linear control synthesis tools. This is basically a feedback linearization technique for robotic manipulator and CTC has a very good performance characteristics.

### 4.3 PD-Sliding Mode Controller (SMC)

Sliding mode control (SMC) is rapidly acquiring acceptance for its better robustness. It is insensitive to various parameter fluctuations and model disturbances. Additionally, the ease of implementation of numeric and computational algorithm of SMC in microcontroller made it's a cutting advantage over classical PID and CTC methods. Proposed PD-SMC control law is defined as follows:

$$\tau = k_p e + k_v \dot{e} + H \text{sign}(\dot{e} + \lambda e) \quad (4.11)$$

Where  $k_p, k_v$  the position and velocity gains,  $H$  is SMC gain vector,  $\lambda$  is the sliding surface slope constant and  $\text{sign}$  is the sign function.

To reduce chattering problem, the saturation function is used in control law instead of sign function and can be modifies as follows:

$$\tau = k_p e + k_v \dot{e} + H \text{sat}((\dot{e} + \lambda e), \phi) \quad (4.12)$$

Where  $\phi$  is a constant diagonal matrix that determines the boundary layer of the sliding surface. The saturation function is defined as follows:

$$\text{sat}((\dot{e} + \lambda e), \phi) = \begin{cases} \text{sign}(\dot{e} + \lambda e) & \text{if } |\dot{e} + \lambda e| > \phi \\ \frac{(\dot{e} + \lambda e)}{\phi} & \text{if } |\dot{e} + \lambda e| \leq \phi \end{cases} \quad (4.13)$$

The stability of the control law was proved by considering Lypunove function  $L$  to be positive definite and  $\dot{L}$  as negative definite [57]. The proposed PD-SMC is globally asymptotically stable according to Lyapunov method where its error and error derivatives tend to zero

## 4.4 Design of Disturbance Observer

While transmitting the motion from joints to the end-effector, there is a marked difference in the trajectory of the practical system when compared to computer models. This is due to the effect of several parameters of manipulator like joint friction, link elasticity and uncertainties in the parameters of mechanism. In order to account the effects of this external disturbances and unmodelled uncertainties, several estimator models were reported in literature. Use of disturbance rejection control with a disturbance observer (DO) for the robotic manipulators is still an important research area, where the stability and controllability are of main concern.

Few objectives behind the usage of DO in robotic manipulator are as follows:

- 1) Controlling joints independently is popular among industrial robot and there performance can be enhanced by the use of such DO to approximate and compensate the coupling torques from additional link as a disturbance (external).
- 2) Friction is unavoidable in any mechanical system and consistently affect the system performance, DO seems a better methodology for estimating and compensating friction model
- 3) DO's also works as torque sensor for force feedback and position control, thus eliminating the use of additional sensor thereby reducing the cost, refining reliability by simplifying system structure.
- 4) DO's deliver signals for monitoring and trajectory tracking which is advantageous for robot working in highly uncertain environment.

The primary objective of the use of DO's is to gather external, unknown parameter or uncertain disturbance torques without the use of an additional sensor. Disturbance observers can lead to fast, good trajectory tracking performance and smooth control actions without the use of large feedback gains. The model of the n-link manipulator with disturbance vector  $d$  can be expressed as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau + d \quad (4.14)$$

Here  $\theta \in \mathbb{R}^n$  is the joint position vector,  $\dot{\theta} \in \mathbb{R}^n$  is vector of joint,  $\ddot{\theta} \in \mathbb{R}^n$  is the vector of joint accelerations,  $M(\theta) \in \mathbb{R}^{n \times n}$  is the mass matrix,  $C(\theta, \dot{\theta}) \in \mathbb{R}^n$  is Coriolis and centrifugal force vector,  $G(\theta) \in \mathbb{R}^n$  is vector for gravitational forces,  $\tau \in \mathbb{R}^n$  is the vector for applied torques and  $d$  is vector representing external disturbances.

Figure 4.3 shows the block diagram of the proposed Disturbance Observer for SMC system.

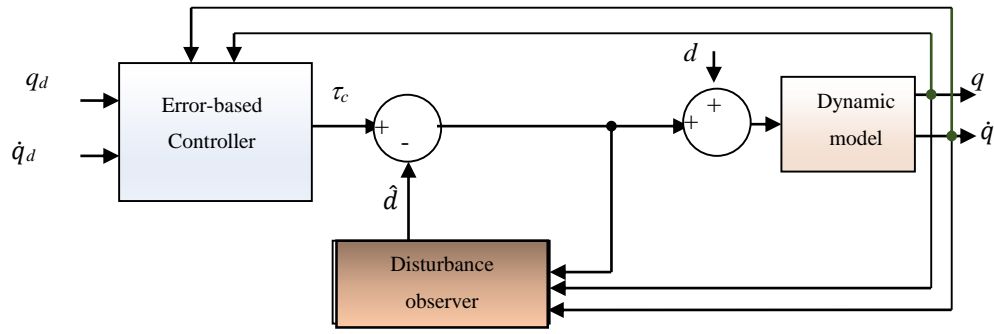


Figure 4.2 Block –Diagram of Disturbance Observer

In order to design a non-linear disturbance observer, a generic approach is followed using an auxiliary variable vector:

$$Z = \hat{d} - p(\theta, \dot{\theta}) \quad (4.15)$$

Where  $Z \in \mathbb{R}^n$ ,  $\hat{d} \in \mathbb{R}^n$  a vector representing estimated disturbance and  $p(\theta, \dot{\theta})$  has to be found out. A nonlinear function  $K(\theta, \dot{\theta})$  has been defined such that

$$K(\theta, \dot{\theta})M(\theta)\ddot{\theta} = \frac{dp(\theta, \dot{\theta})}{dt} \quad (4.16)$$

Let  $\dot{\hat{d}}$  is being defined by the following equation

$$\dot{\hat{d}} = -K(\theta, \dot{\theta})(\hat{d} - d) \quad (4.17)$$

Differentiating eq.(4.15) and substituting  $\dot{\hat{d}}$  into Eq. (4.17) and simplifying

$$\dot{Z} = -K(\theta, \dot{\theta})(Z + p(\theta, \dot{\theta})) + K(\theta, \dot{\theta})(M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta)) - \tau - K(\theta, \dot{\theta})M(\theta)\ddot{\theta}$$

$$-K(\theta, \dot{\theta})Z + K(\theta, \dot{\theta})(C(\theta, \dot{\theta})) + G(\theta) - \tau - p(\theta, \dot{\theta}) \quad (4.18)$$

And the error in the force estimation is given by the following equation

$$e = d - \hat{d} \quad (4.19)$$

Because no exact information about the disturbance is available, we assume that  $\dot{d} = 0$  and then differentiating the previous equation we get

$$\dot{e} = \dot{d} - \dot{\hat{d}} = -K(\theta, \dot{\theta})e \quad (4.20)$$

Thus the estimation  $\hat{d}$  converges to  $d$  if the function  $K(\theta, \dot{\theta})$  defined in eq (4.17) is asymptotically stable. So the  $p(\theta, \dot{\theta})$  should be selected in such a way that the function  $K(\theta, \dot{\theta})$  satisfy the stability conditions in Eq (4.17). Although it is difficult to select such a proper function, but in the case of robotics the preferred choice is  $p(\theta, \dot{\theta}) = c\dot{\theta}$ , where  $c$  is a any positive scalar which is necessary for convergence. Hence the choice becomes:

$$p(\theta, \dot{\theta}) = c\dot{\theta} \quad (4.21)$$

Using Eq (4.16), we have

$$K(\theta, \dot{\theta}) = cM^{-1}(\theta) \quad (4.22)$$

In the case of robotics manipulator, the mass matrix is positive-definite and also symmetric. Thus  $K(\theta, \dot{\theta}) = cM^{-1}(\theta)$  is also symmetric and positive –definite matrix.

Next, Lyapunov function is defined as follows:

$$\dot{L}(e) = \frac{1}{2} e^T e \quad (4.23)$$

Further diffentiating along the observer trajectory gives

$$\dot{L}(e) = -ce^T M^{-1}(\theta)e \quad (4.24)$$

Therefore,  $\frac{dL(e)}{dt} < 0, \forall e, t$ , because of positive definiteness of inverse of mass matrix.

Hence Eq (4.18) is stable exponentially and the convergence rate is directly proportional to  $c$ . In present work,  $c = 0.01$  is considered.

Computer programs are developed in Matlab to control the joint space trajectories with and without disturbances at the joints.

## 4.5 Numerical studies for selected Trajectories

The simulations are performed in the joint space for two cases of trajectories. A flowchart of the present approach is shown in Figure 4.3.

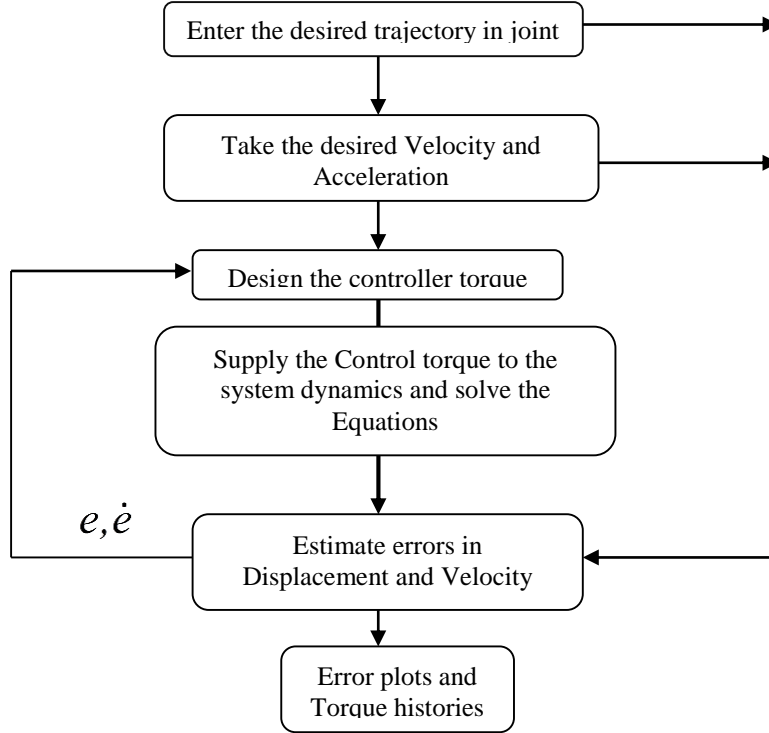


Figure 4.3 Flow chart of trajectory tracking process

Computer programs are developed for both CTC and PD-SMC in Matlab which takes displacement and velocity at every joint and computes joint errors using forward dynamics solution in every time step. In each time step differential equations are solved and in total it took on an average of 10 seconds to compute the errors in entire trajectory using a core-i5 processor with 1 GHz speed.

### 4.5.1 First case

The joint space trajectories considered for the three joints of the arm are as follows [48]

$$\begin{aligned}
 \theta_1 &= \frac{\pi}{5} \left( \frac{t}{T} - \frac{1}{2\pi} \sin \left( 2\pi \frac{t}{T} \right) \right) \\
 \theta_2 &= \frac{\pi}{8} \left( \frac{t}{T} - \frac{1}{2\pi} \sin \left( 2\pi \frac{t}{T} \right) \right), \\
 \theta_3 &= \frac{\pi}{12} \left( \frac{t}{T} - \frac{1}{2\pi} \sin \left( 2\pi \frac{t}{T} \right) \right)
 \end{aligned} \tag{4.25}$$

$0 \leq t \leq T$



Where  $T$  is the duration time of the manipulator and taken as 10 sec. Initially it is at rest. The friction at the joints is modeled as

$$\tau_{friction} = F_{ci} \operatorname{sgn}(\dot{q}_i) \left[ 1 - \exp\left(\frac{-\dot{q}_i^2}{v_{si}^2}\right) \right] + F_{si} \operatorname{sgn}(\dot{q}_i) \exp\left(\frac{-\dot{q}_i^2}{v_{si}^2}\right) + F_{vi} \dot{q}_i \quad (4.26)$$

Where  $F_{ci}, F_{si}, F_{vi}$  are the Coulomb, static and viscous friction coefficient respectively. The  $v_{si}$  is the Stribeck parameters. Table 4.1 shows the dynamic data of the manipulator along with the controller constants.

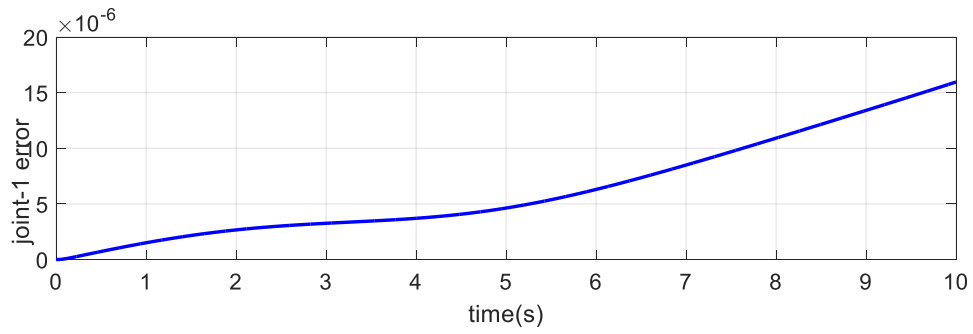
Table 4.1 Simulation parameters used [61, 48]

Parameters	Value	Parameter	Values
$m_1$	0.5 kg	$m_2$	0.5 kg
$m_3$	0.5 kg	$m_4$	2 kg
$F_{c1}$	0.49 Nm	$F_{c2}$	0.31 Nm
$F_{c3}$	0.1 Nm		
$F_{s1}$	3.5 Nm	$F_{s1}$	2.8 Nm
$F_{s3}$	1.65Nm		
$F_{v1}$	0.15 kg m/s	$F_{v2}$	0.12 kg m/s
$F_{v3}$	0.08 kg m/s		
$v_{s1}$	0.19 rad/s	$v_{s2}$	0.15 rad/s
$v_{s3}$	0.12 rad/s	$g$	9.8N/kg
$k_p$	450	$k_v$	$2\sqrt{k_p}$
$H_1$	150	$H_2$	100
$H_3$	75		
$\phi_1$	0.09	$\phi_2$	0.06
$\phi_3$	0.03	$\lambda$	100

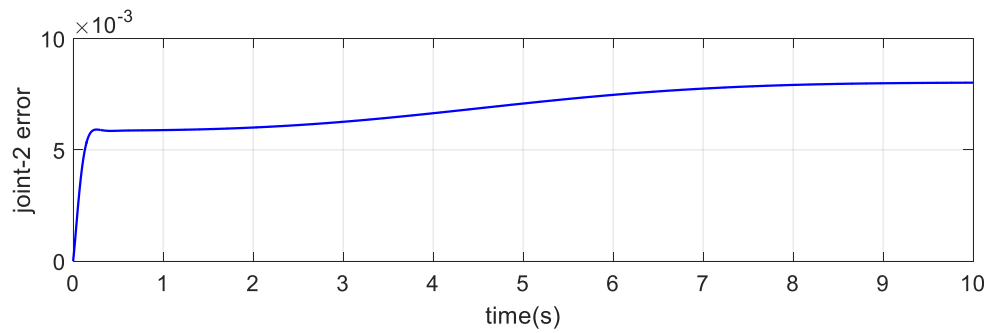
The numerical results of three link manipulator with and without disturbance and disturbance observer are shown sequentially for both CTC and PD-SMC.

#### (a) Computed Torque Control Scheme

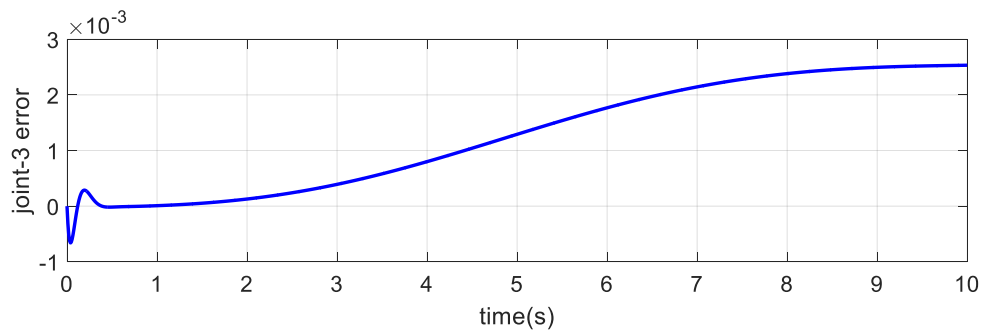
Figure 4.4 shows error plots and trajectory at one of the joint with no disturbances.



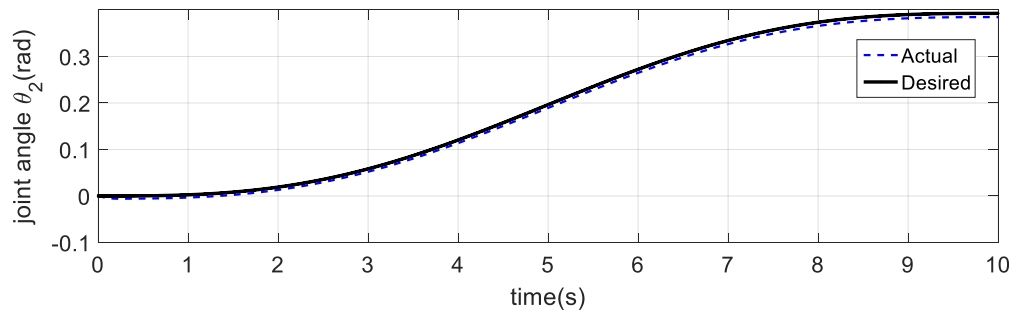
(a) Error at joint 1



(b) Error at joint 2



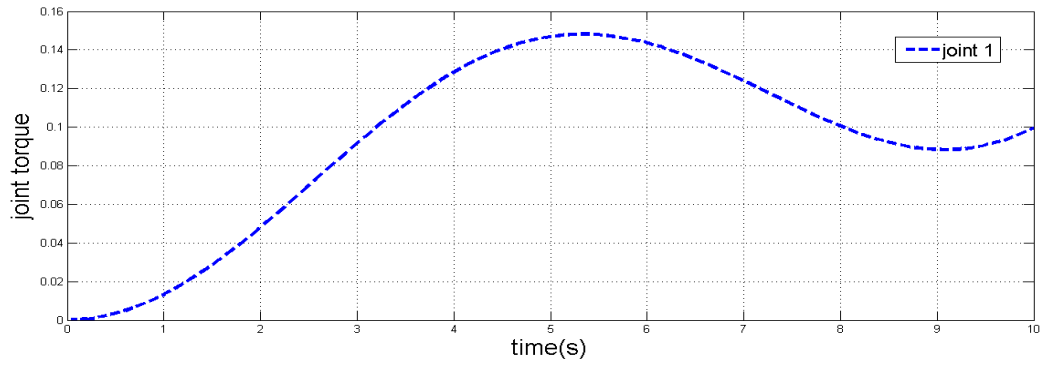
(c) Error at joint 3



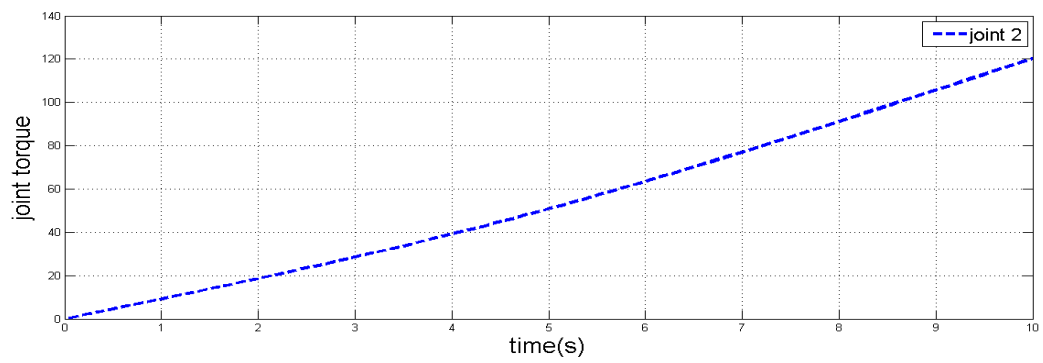
(d) Trajectory at Joint 2

Figure 4.4 Joint Error plots without disturbance (case 1)

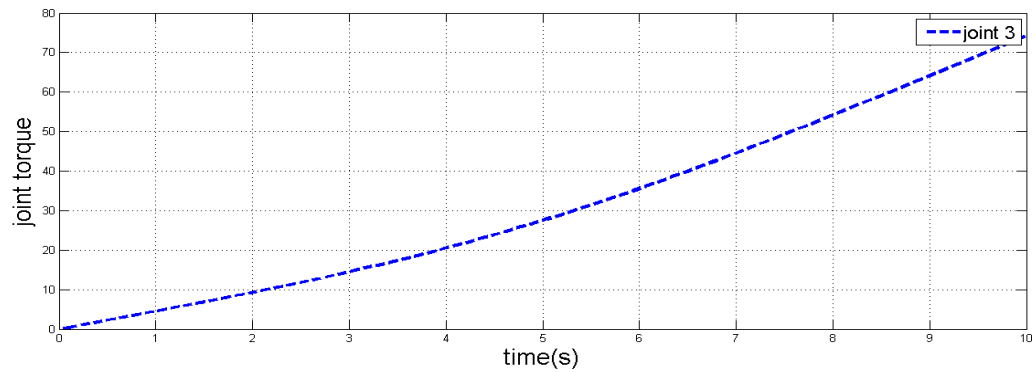
The corresponding joint torques is shown in the Figure 4.5.



(a)Control torque (N-m) at joint 1



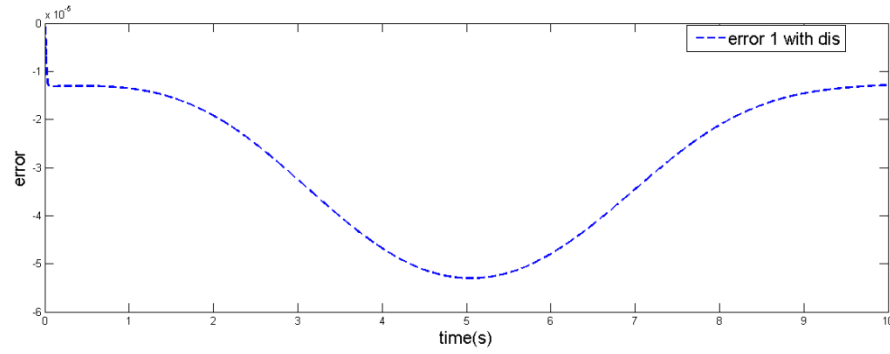
(b)Control torque (N-m) at joint 2



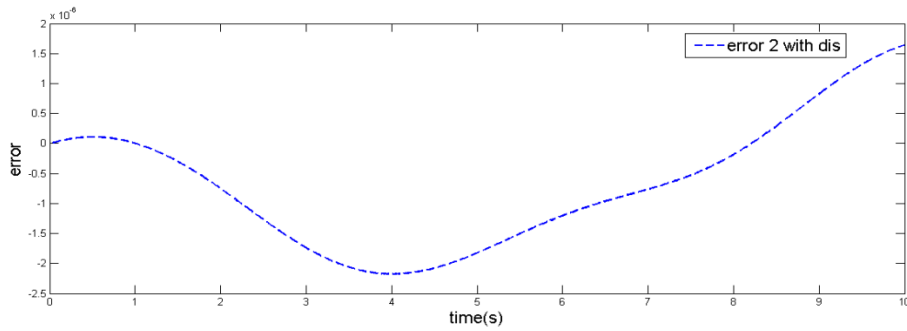
(c)Control torque (N-m) at joint 3

Figure 4.5 Joint Control Torques variations without disturbance (case 1)

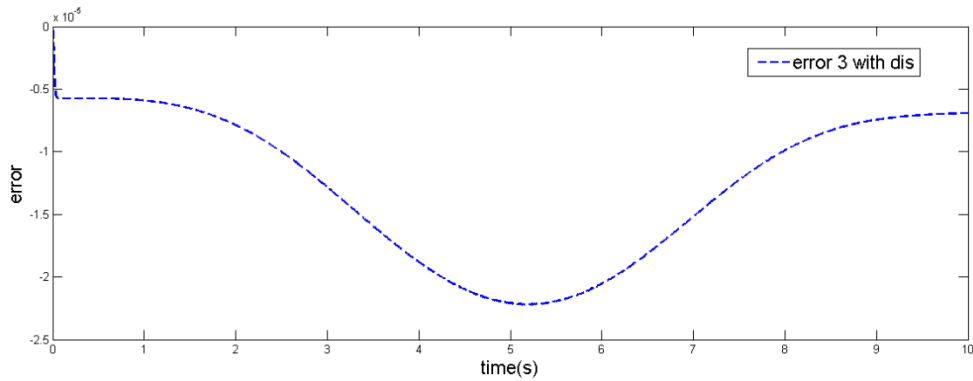
With external disturbance, Figure 4.6 shows the joint error variation along the trajectory. It is observed that the errors are relatively large due to external disturbance.



(a) Error at joint 1



(b) Error at joint 2

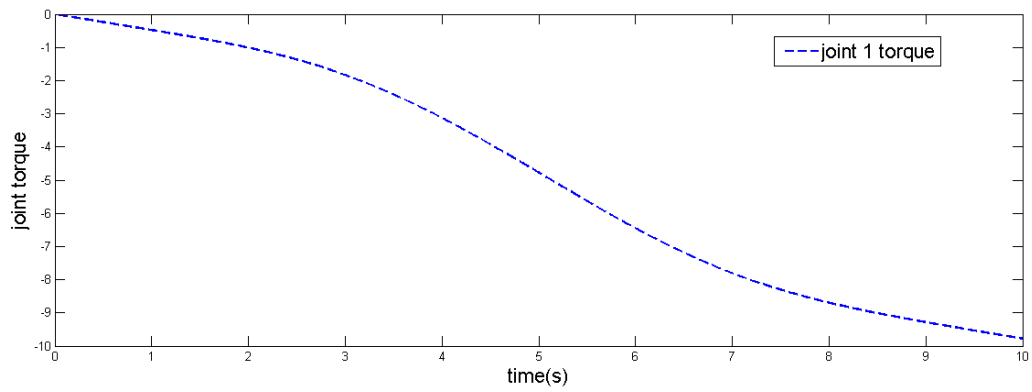


(c) Error at joint 3

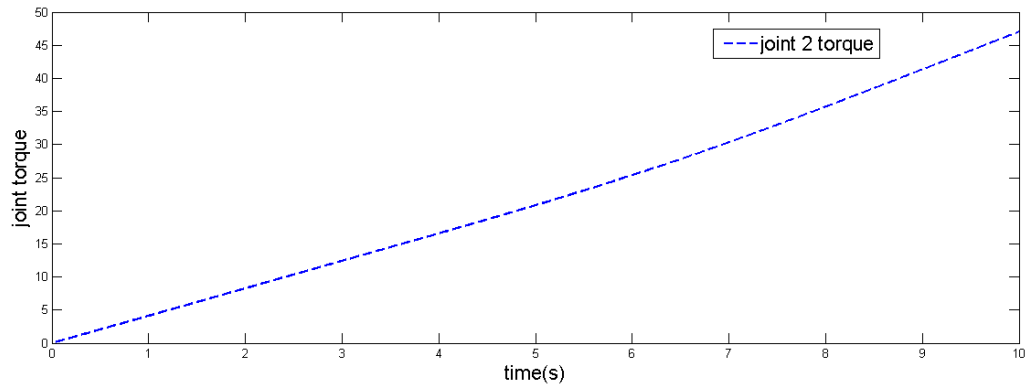
Figure 4.6 Joint Error plots with disturbance case

Figure 4.7 shows the control torques in each joint. Joint 1 which is found to be very low as compared to other joint torques, provided with the given trajectory with respect to time in seconds. The Figure 4.7 (b) shows the control torque requirement in the joint 2 which is found to be very high as compared to other joint torques, provided with the given trajectory with respect to time in seconds. The Figure 4.7 (c) below shows the control

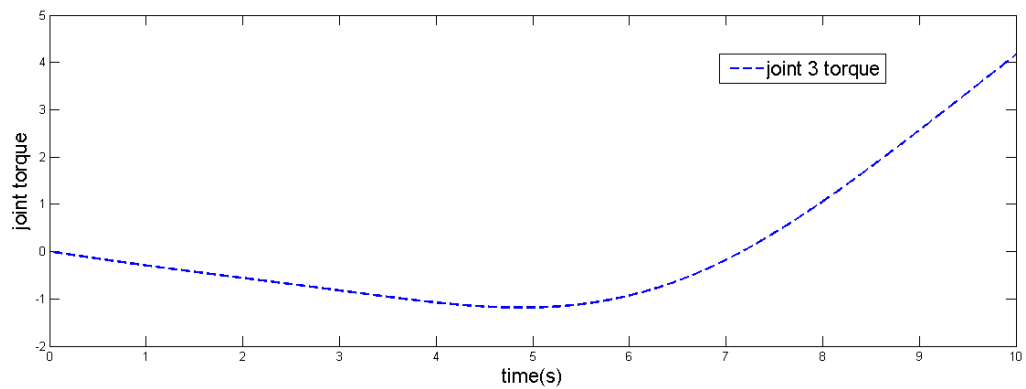
torque requirement in the joint 3 which is found to be average as compared to other joint torques, provided with the given trajectory.



(a) Control torque (N-m) at joint 1



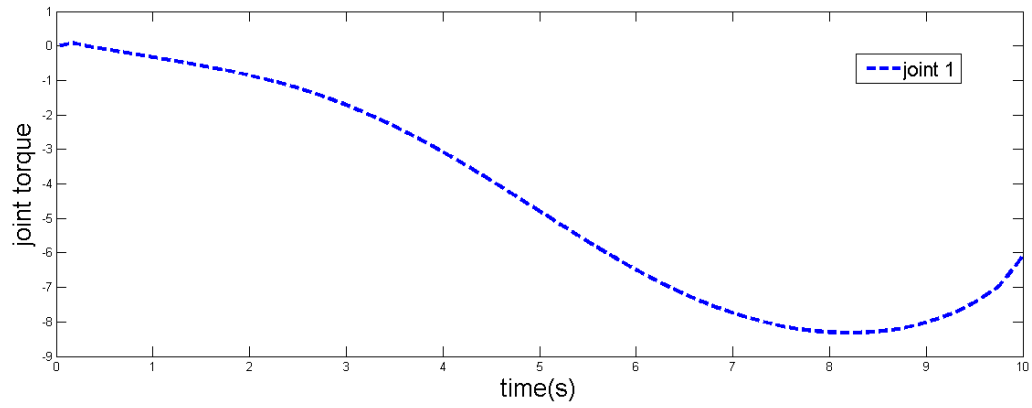
(b) Control torque (N-m) at joint 2



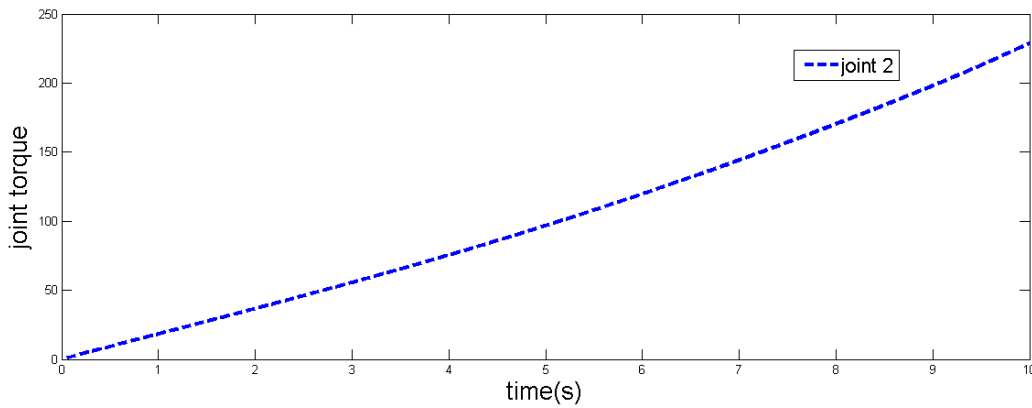
(c) Control torque (N-m) at joint 3

Figure 4.7 Joint Control Torques variations with disturbance (case 1)

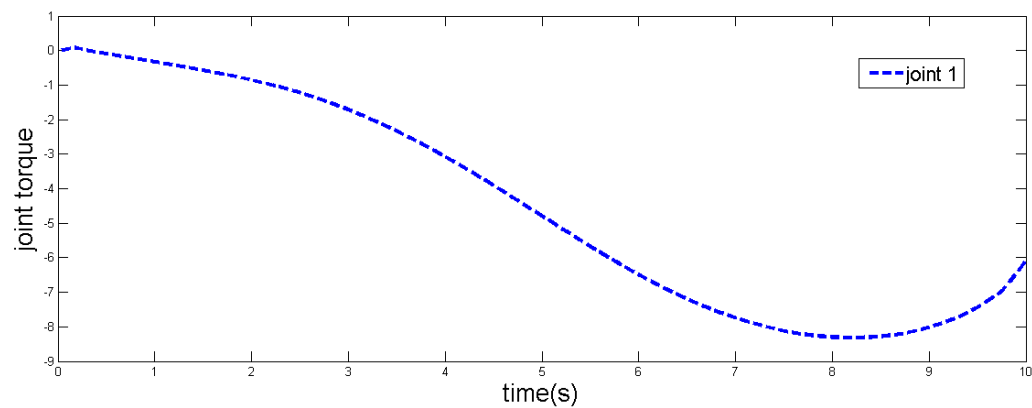
Now for the system with disturbance observer ( $c=0.001$ ), the corresponding joint torques are shown in Figure 4.8.



(a) Control torque (N-m) at joint 1



(b) Control torque (N-m) at joint 2



(c) Control torque (N-m) at joint 3

Figure 4.8 Resultant control torques (N-m) with disturbance observer

Here the torque at the joint 2 is relatively more and the magnitudes of the torques at all the joints have been reduced. Figure 4.9 shows the joint errors with disturbance observer.

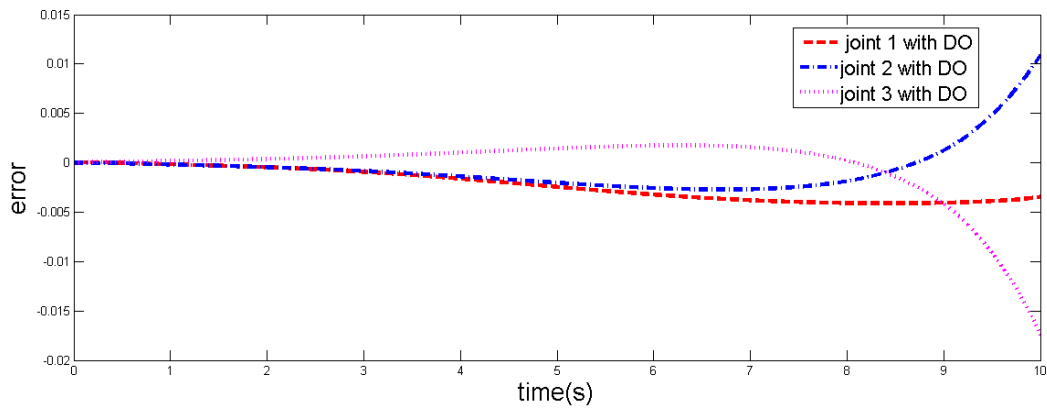


Figure 4.9 Error at joints with disturbance observer

### (b) PD-SMC

Without disturbance, PD-SMC is first applied to the system. In these cases, same gain values are used. Figure 4.10 depicts the joint trajectory and the corresponding output tracking performance.

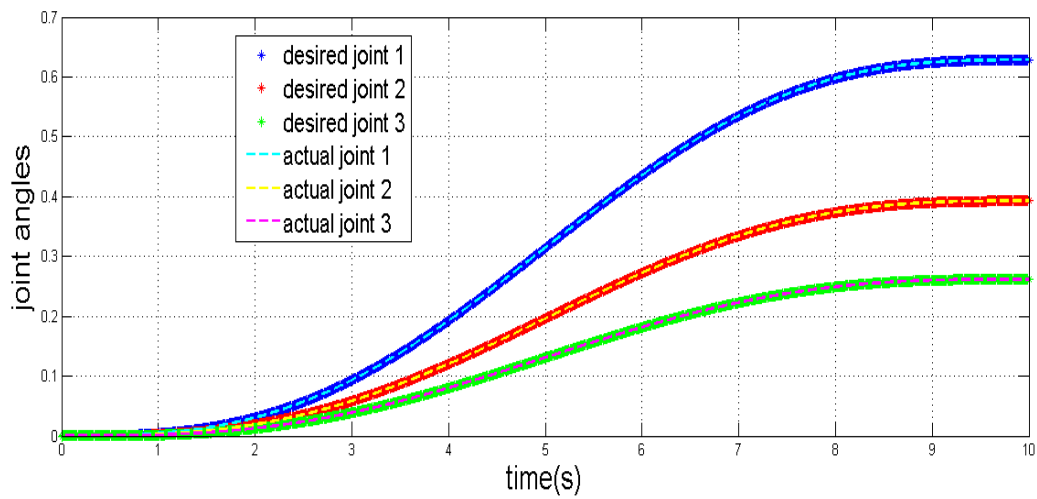
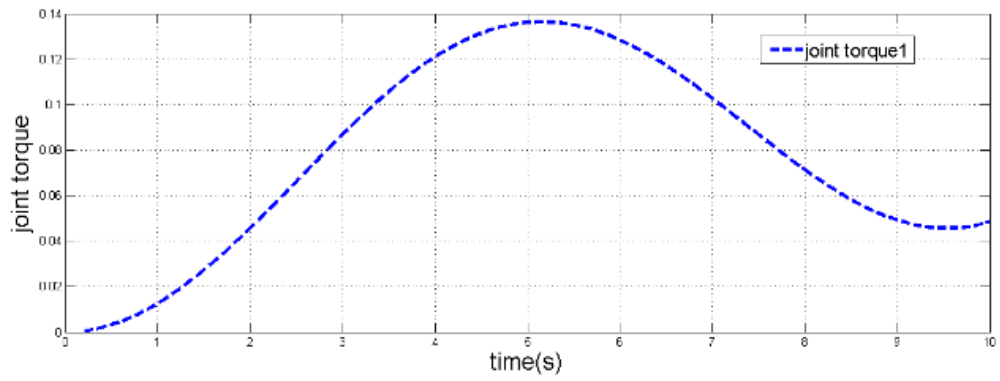
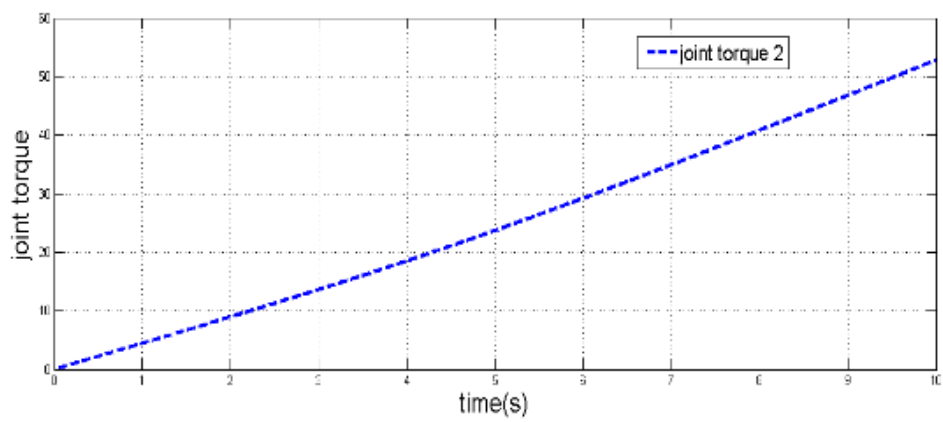


Figure 4.10 Desired trajectory vs. Actual trajectory

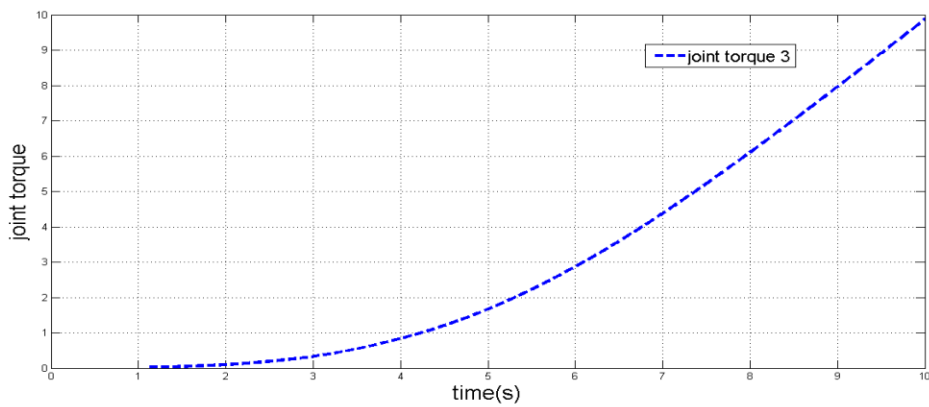
This Figure 4.11 below depicts the joint torque profile in (N-m) required to execute the desired trajectory. It is observed that the joint 2 requires highest torque throughout the trajectory.



(a) Joint 1



(b) Joint 2



(c) Joint 3

Figure 4.11 Control torques (N-m) along the trajectories



Figure 4.12 shows the error convergence with respect to time at all the three joints. In this case, because of no disturbance, all the errors in the respective joints are converging exactly to zero in a given period of time and the performance of PD-SMC seems to be better.

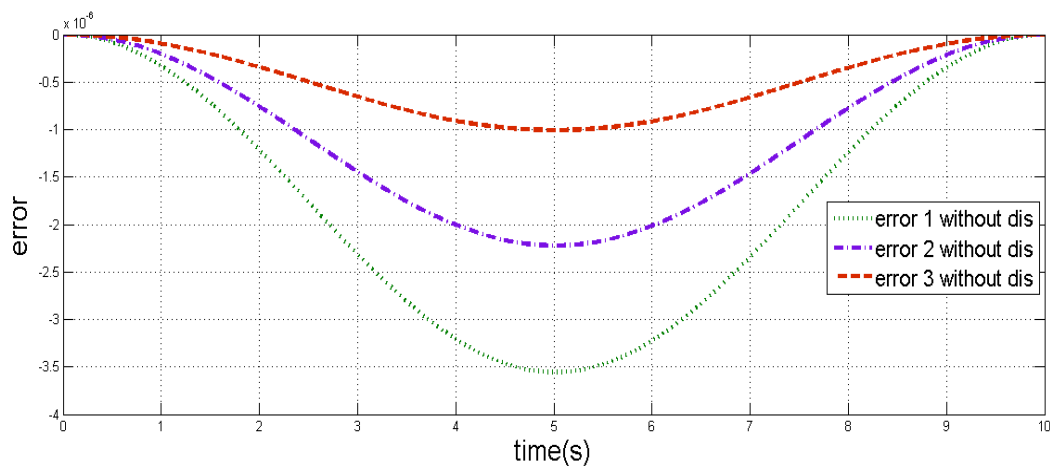


Figure 4.12 Joint Error

With disturbance observer, the performance of PD-SMC is presented in the form of joint errors and joints torques. Figure 4.13 shows the joint errors along the trajectory with disturbance observer.

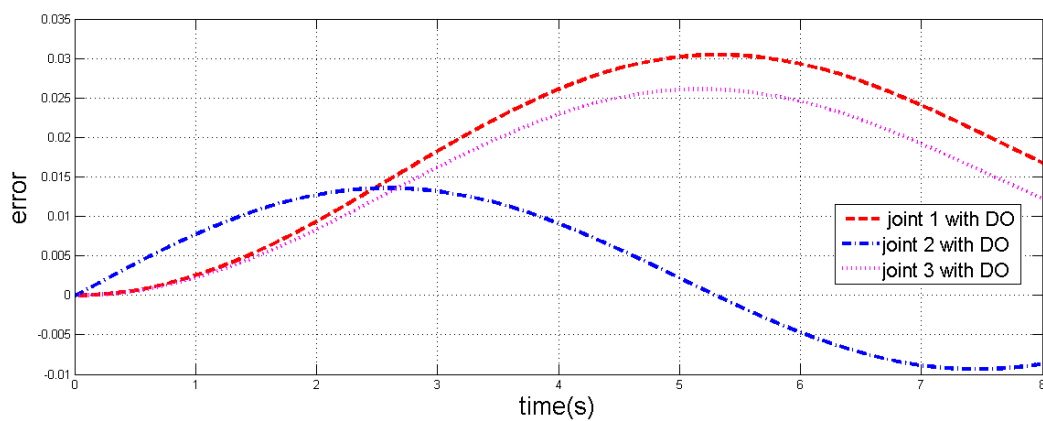
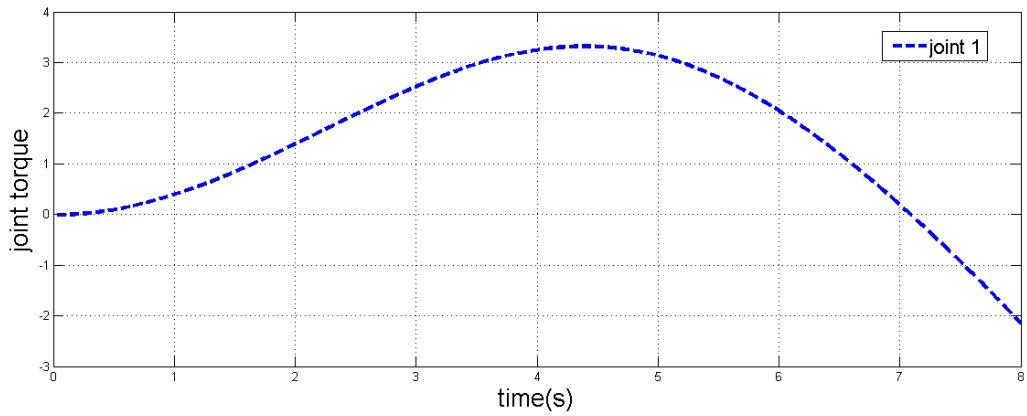
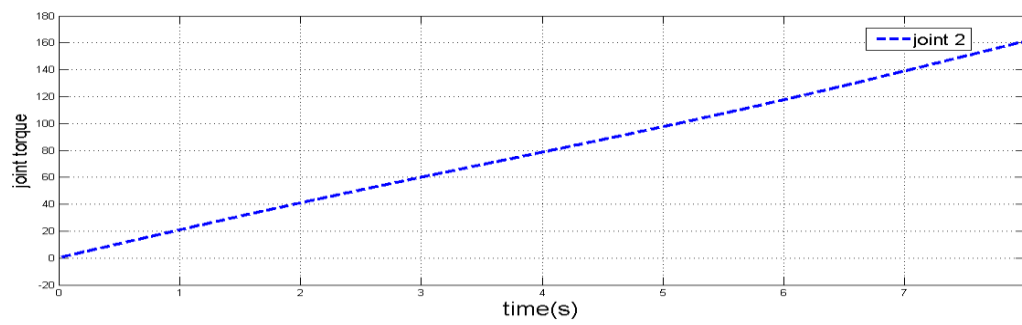


Figure 4.13 Joint errors along the trajectory

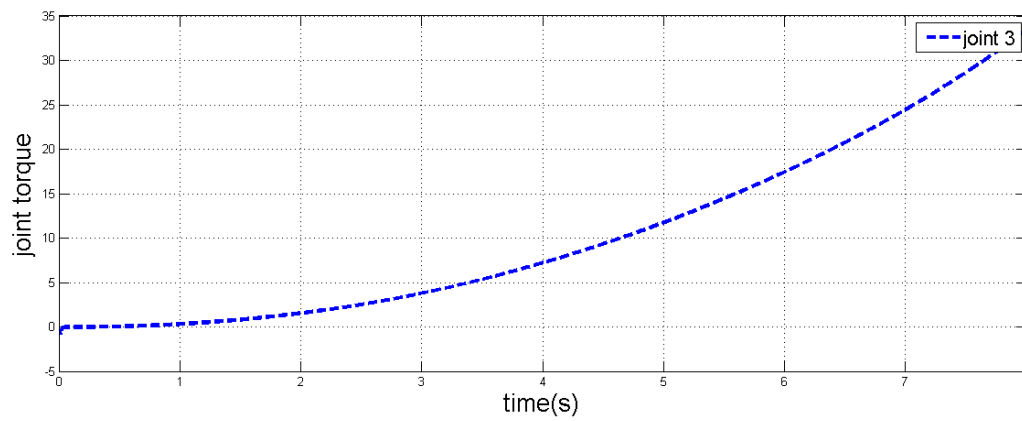
Figure 4.14 shows the corresponding net torque at each joint.



(a) Control torque (N-m) at joint 1



(b) Control torque (N-m) at joint 2



(c) Control torque (N-m) at joint 3

Figure 4.14 resultant joint control torques

It is observed that the joint control torques with observer have reduced drastically as compared to previous case.

### 4.5.2 Second case

The second set of joint space trajectories considered for the three joints of the robot are as follows [60]

$$\begin{aligned}\theta_1 &= \cos\left(\frac{t}{5\pi}\right) - 1 \\ \theta_2 &= \cos\left(\frac{t}{5\pi} + \frac{\pi}{2}\right), \quad 0 \leq t \leq T \\ \theta_3 &= \sin\left(\frac{t}{5\pi} + \frac{\pi}{2}\right)\end{aligned} \quad [4.27]$$

#### (a) CTC

Without disturbance, CTC is applied to the system for a second set of trajectories. In these cases, same gain values are used. Figure 4.15 depicts the trajectory tracking performance of the desired path and actual path followed by CTC controller obtained without disturbance torques being accounted.

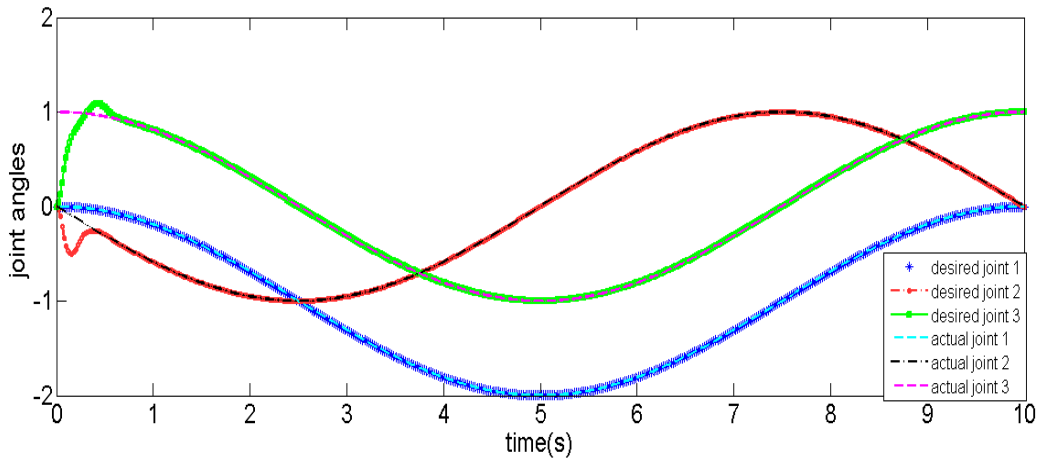
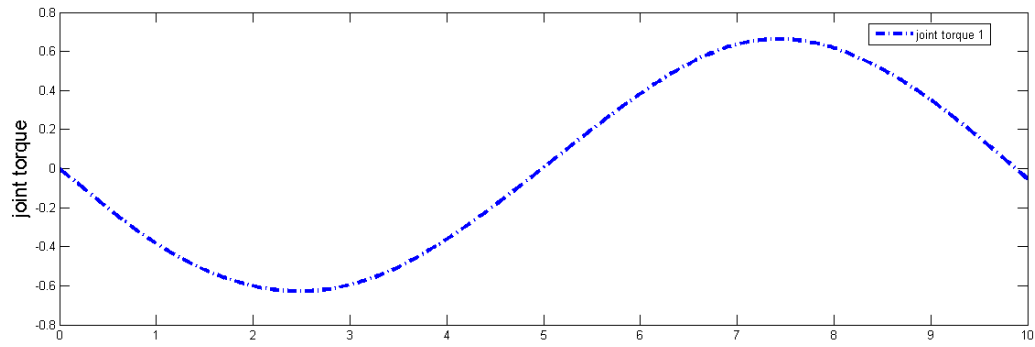
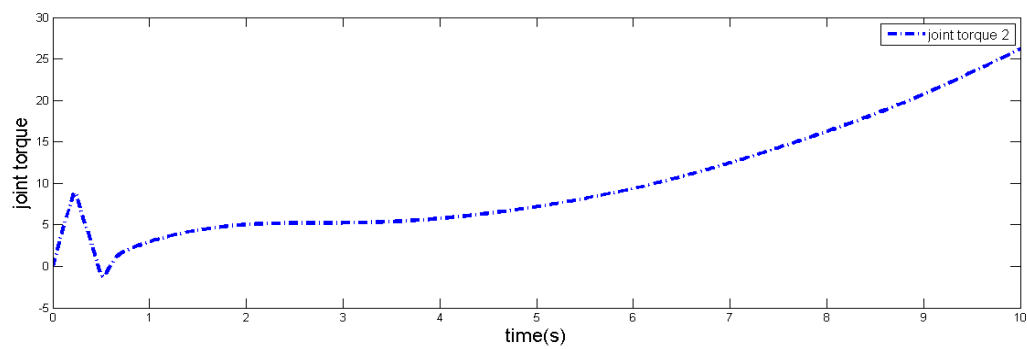


Figure 4.15 Desired trajectory vs. Actual trajectory (CTC)

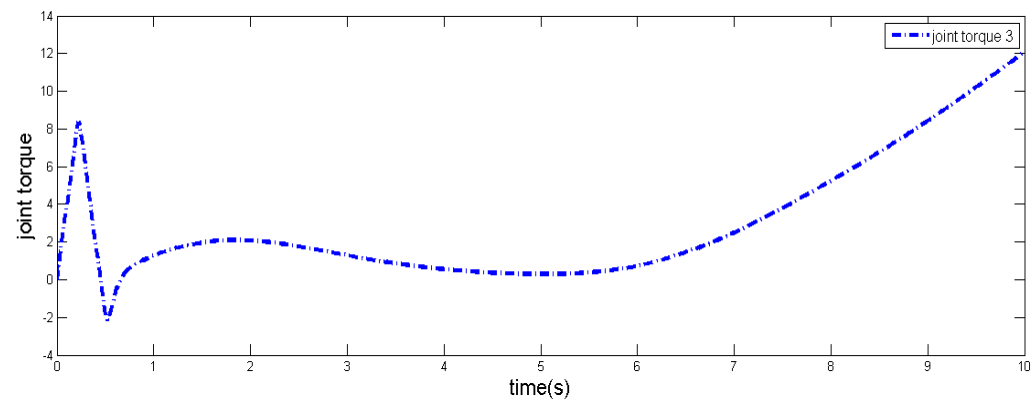
Figure 4.16 below shows the control torque requirement at the joint 1 which is found to be very low as compared to other joint torques, provided with the given trajectory. From the figure, it can be seen that the control torque requirement in the joint 2 is found to be very high as compared to other joint torques.



(a) Control torque (N-m) at joint 1



(b) Control torque (N-m) at joint 2



(c) Control torque (N-m) at joint 3

Figure 4.16 Joint Control Torques variations without disturbance (case 2)

Figure 4.17 below shows the error convergence with respect to time. In this case, because of no disturbance all the error in the respective joints is converging exactly to zero.

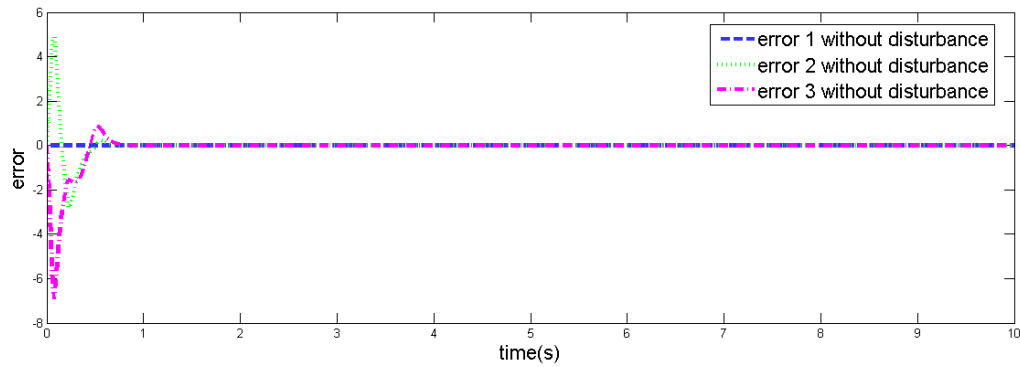


Figure 4.17 Joint errors along the trajectory at all individual joints

With external disturbance, CTC is applied to a second set of trajectories. In these cases, same gain values are used. Figure 4.18 depicts the trajectory tracking performance of the desired path given and actual path followed.

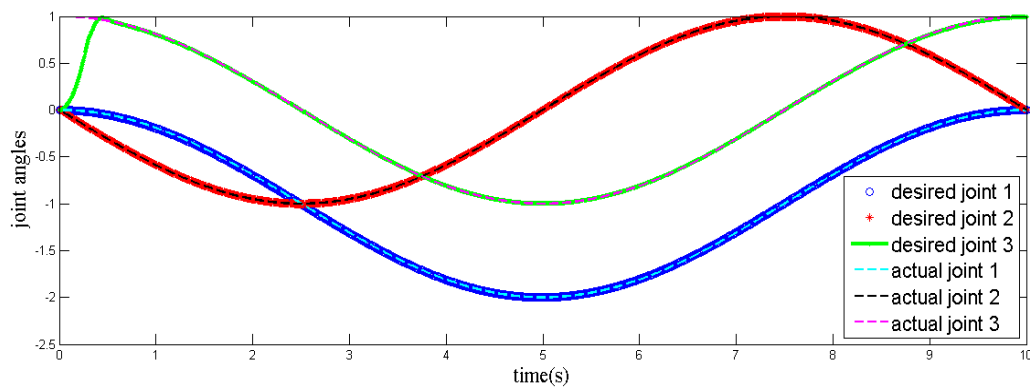


Figure 4.18 Trajectory tracking performance of CTC

CTC controller results with external disturbance being taken into consideration and Figure 4.19 shows the joint error plots.

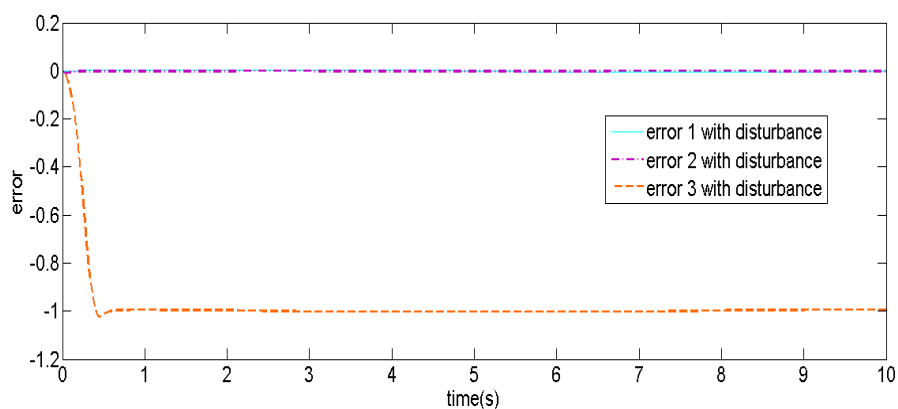
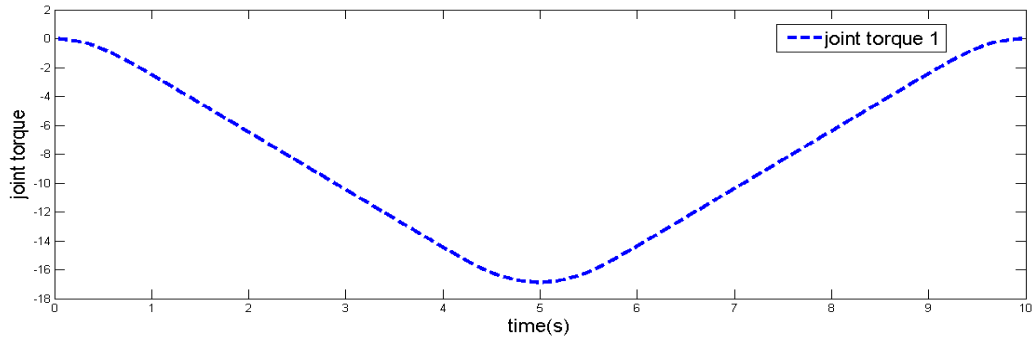
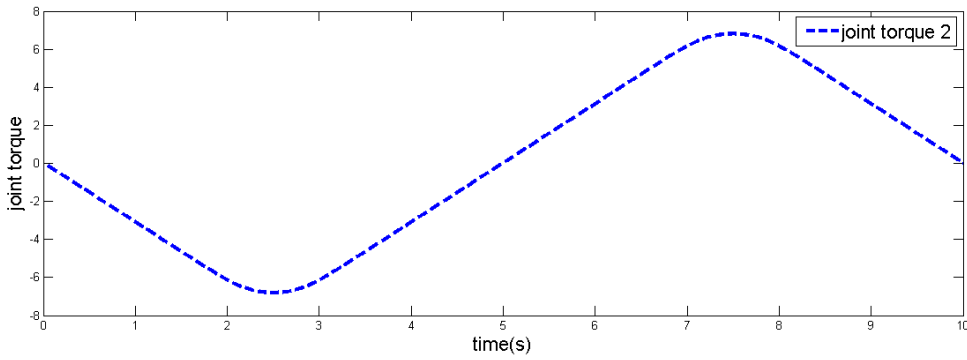


Figure 4.19 Joint errors along the trajectory

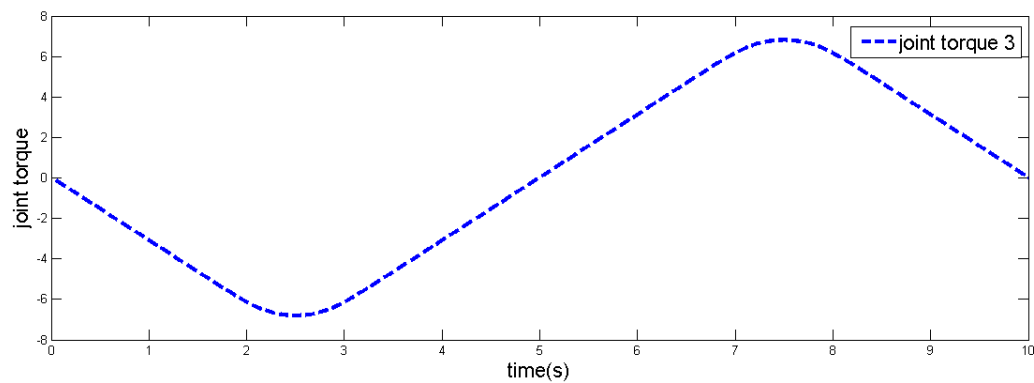
The Figure 4.20 shows the corresponding control torque requirement at each joint.



(a) Control torque (N-m) at joint 1



(b) Control torque (N-m) at joint 2



(c) Control torque (N-m) at joint 3

Figure 4.20 Joint Control Torques variations with disturbance (case 2)

Figure 4.21 shows the actual and desired trajectories with CTC plus disturbance observer with  $c=0.001$ . Objective here is to make the output of the observer equal to the disturbance

(joint friction torques) and from the figure we could see a better trajectory tracking performance is achieved because of the disturbance observer.

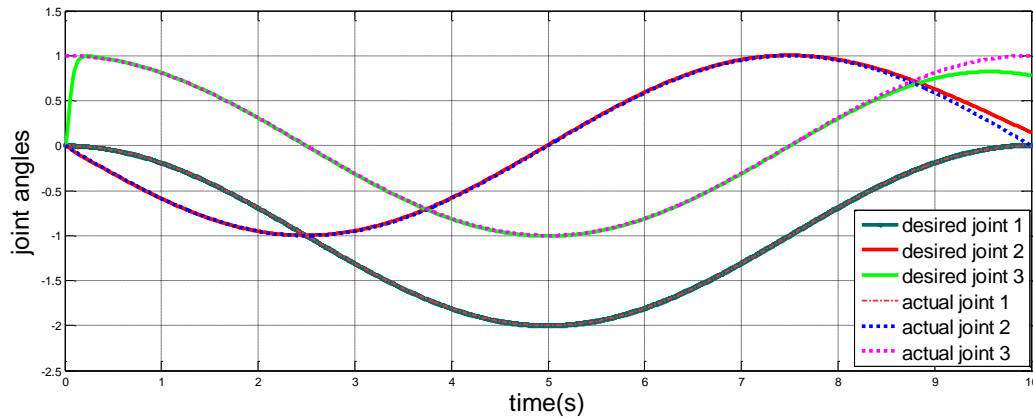


Figure 4.21 Trajectory tracking performance of CTC with DO

Figure 4.22 shows the corresponding error plots at three joints. Figure 4.23 below depicts the joint torque profiles required to execute the desired trajectory. It is observed that the joint 2 requires highest torque throughout the trajectory but the overall torque requirement has been reduced after the implementation of the observer.

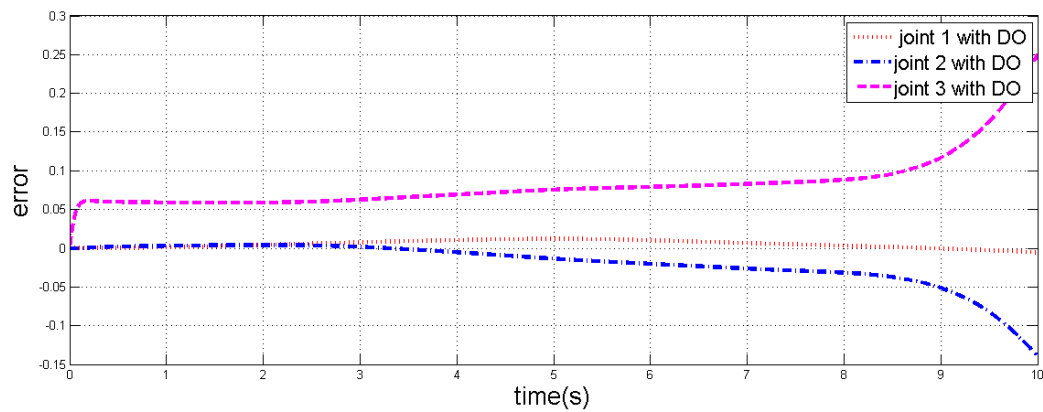
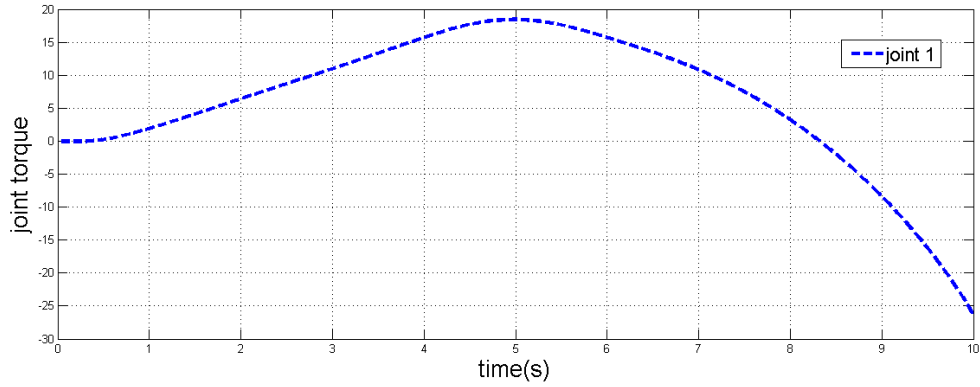
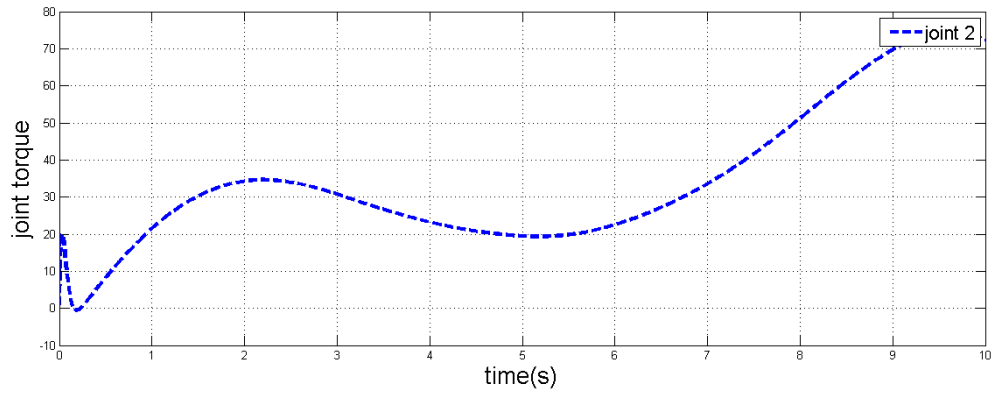


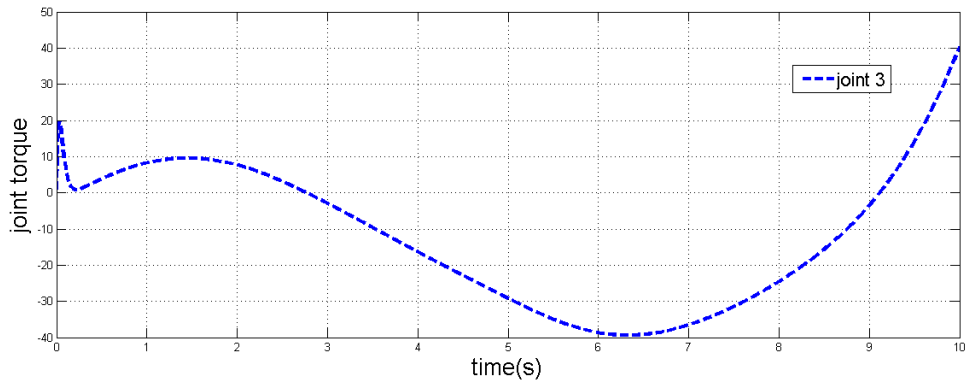
Figure 4.22 Error at joints with disturbance observer (case 2)



(a) Control torque (N-m) at joint 1



(b) Control torque (N-m) at joint 2



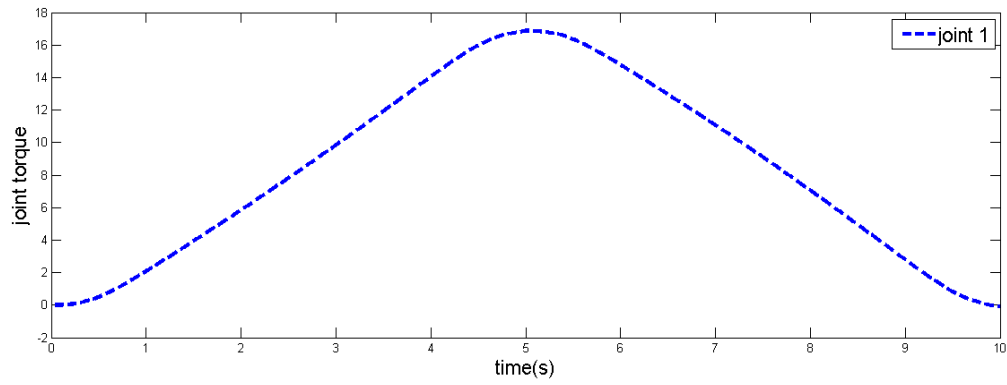
(c) Control torque (N-m) at joint 3

Figure 4.23 Joint Control Torques variations with disturbance Observer (case 2)

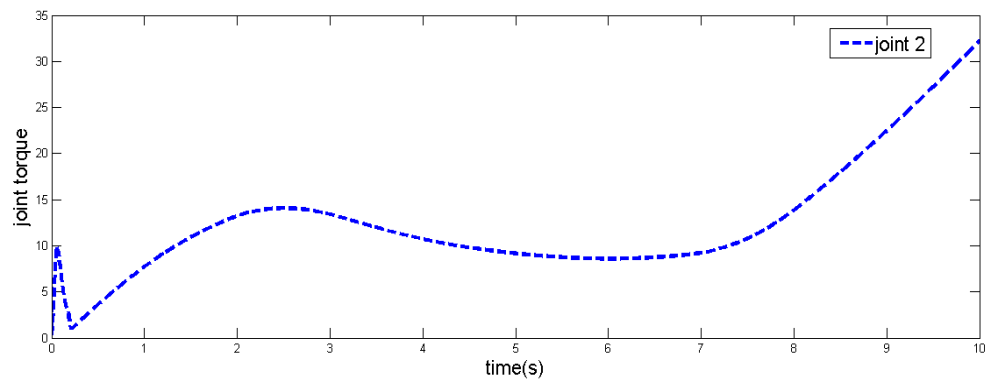
**(b) PD-SMC scheme**

With disturbance, PD-SMC is applied to the system for a second set of trajectories. In these cases, same gain values are used. Figure 4.24 depicts the torque requirement at the different joints with joint friction torques taken into consideration

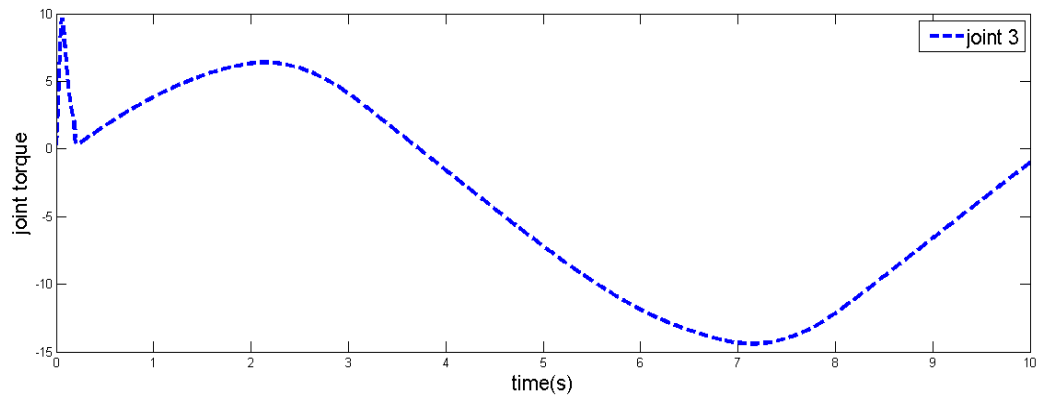




(a) Control torque (N-m) at joint



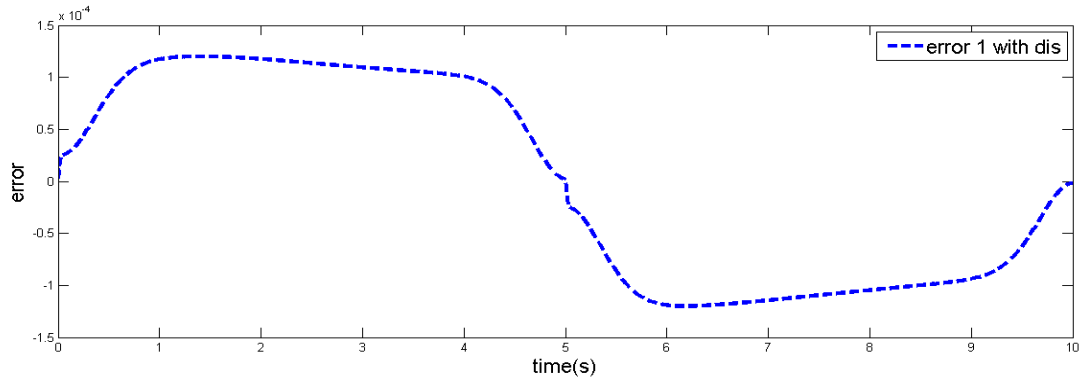
(b) Control torque (N-m) at joint 2



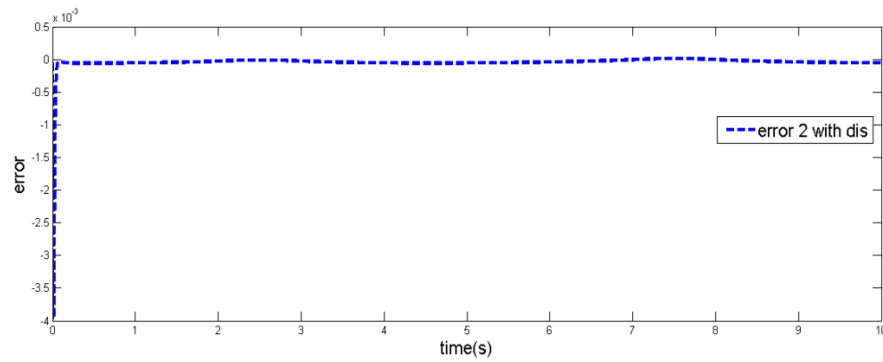
(c) Control torque (N-m) at joint 3

Figure 4.24 Joint Control Torques variations with disturbance (case 2)

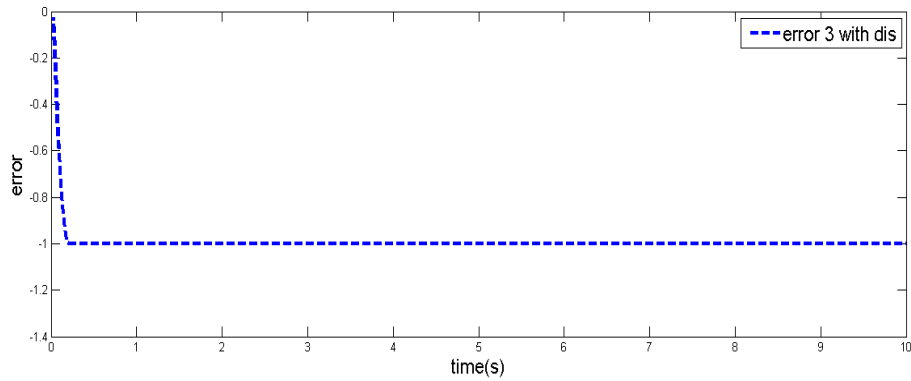
Figure 4.25 shows the error at the individual joints when disturbance has been taken into account



(a) Error at joint 1



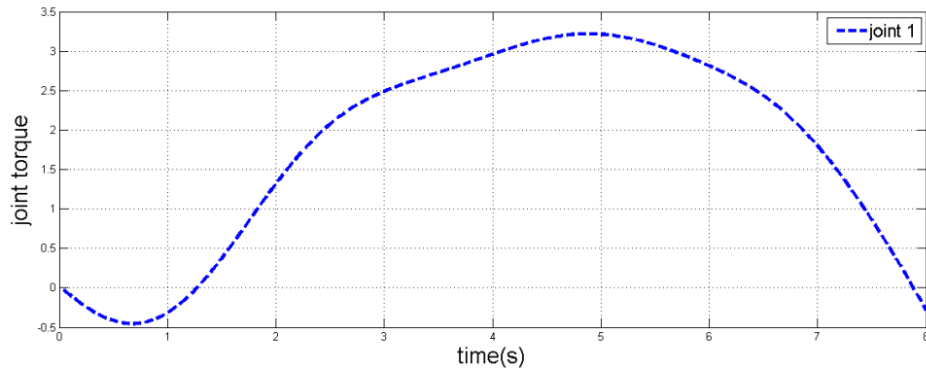
(b) Error at joint 2



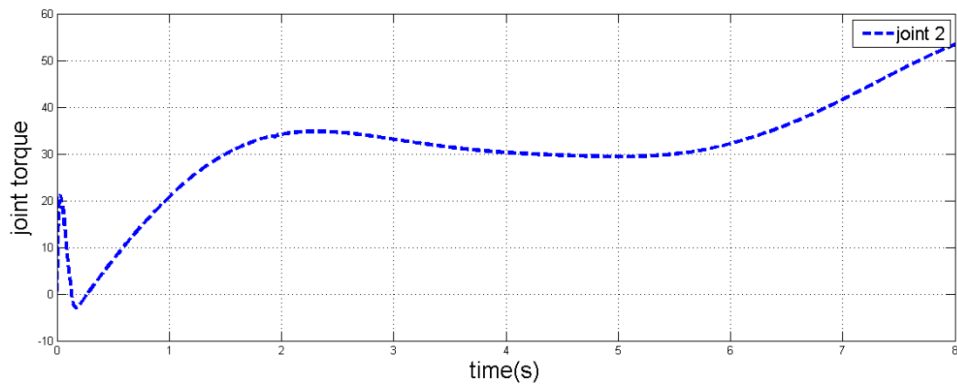
(c) Error at joint 3

Figure 4.25 Error at joints with disturbance (case 2)

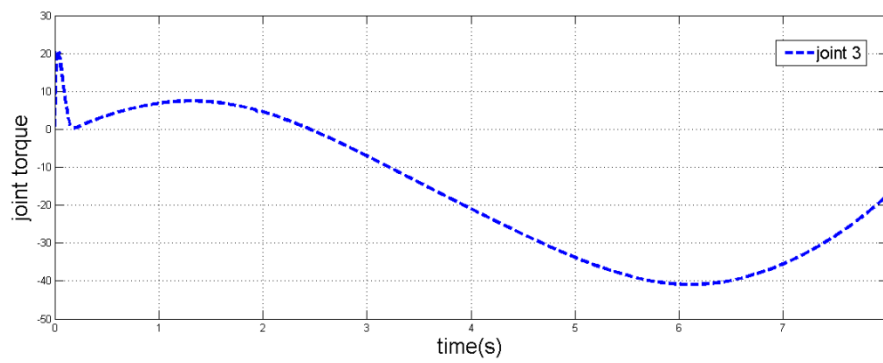
PD-SMC with disturbance observer with  $c=0.001$  has been implemented in second set of trajectory. Objective here is to make the output of the observer equal to the disturbance (joint friction torques). The corresponding joint torques are shown in Figure 4.26



(a) Control torque (N-m) at joint 1



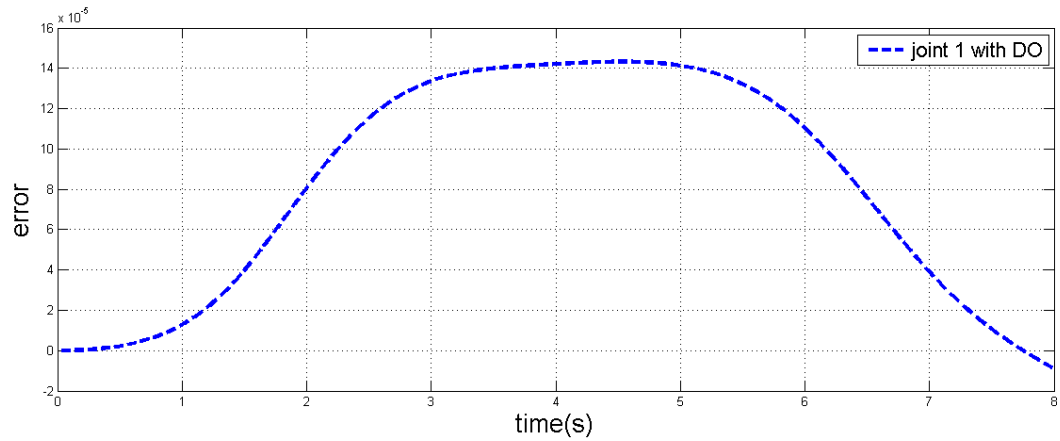
(b) Control torque (N-m) at joint 2



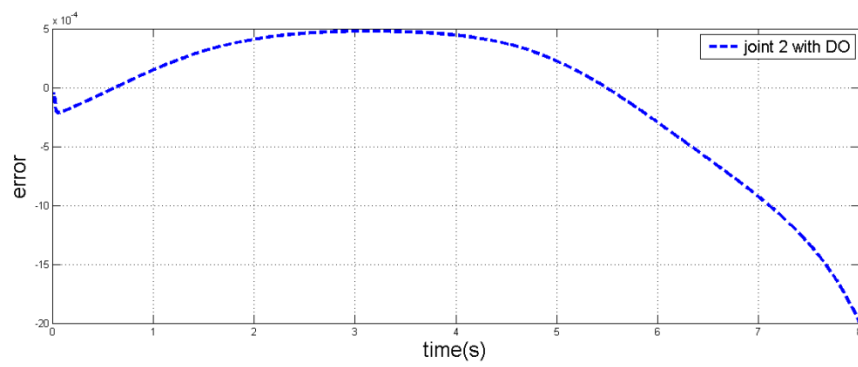
(c) Control torque (N-m) at joint 3

Figure 4.26 Joint Control Torques

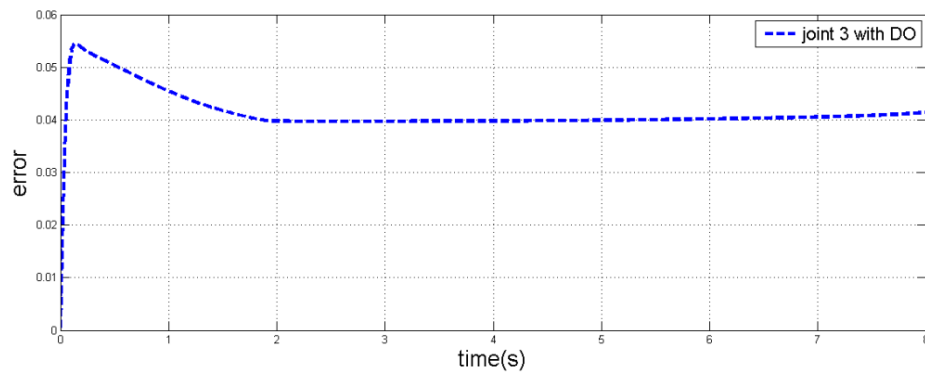
Figure 4.27 shows the joint errors along the trajectory with disturbance observer.



(a) Error at joint 1



(b) Error at joint 2



(c) Error at joint 3

Figure 4.27 Joint Error plots with disturbance observer case

Summary of comparisons of the maximum errors and torque obtained in the two control schemes are given in Table 4.2 for trajectory 1 with disturbance.

Table 4.2 Maximum torque required for the trajectory 1

CTC Joint torque (N-m)				SMC Joint torque (N-m)			
	Joint 1	Joint 2	Joint 3		Joint 1	Joint 2	Joint 3
With disturbance	20	35	10	With disturbance	20	70	40
Without disturbance	0.6	25	12	Without disturbance	17.5	25	5
With Observer	3	40	35	With Observer	3	45	10

## 4.6 Conclusion

Trajectory tracking performance for serial manipulator with two different control techniques in joint space has been presented. Simulation results shows the effectiveness of different controllers for a given trajectories. Results has been presented for two cases and it has been found that with the addition of the disturbance observer the torque at the joints has been reduced and the error has also been minimized as compared to the case without disturbance observer and PD-SMC has been found to be more effective as that of CTC. The presented control schemes can be implemented in a real time control of the fabricated prototype. **Further, the gain selection in the controllers can be made automatic by using a tuning approach in future studies.** The designed control schemes require a feedback data which is estimated using a digital camera system with image processing software. The issues are described in the next chapter.

## Chapter 5

# Visual Servo Control

### 5.1 Introduction

For remote handling, the viewing system is a fundamental tool, which is used to make a survey in hazardous environments for the status of blanket first wall and plasma facing components in fusion reactor vessels. Visual inspection systems are often mounted on the end-effector of in-vessel inspection robot for full volume and close visual inspection of first-wall and plasma facing components in the inner chamber of tokamak vacuum vessel. Sometimes, the camera is mounted at a fixed location like a laser-range finder or a hybrid system where a low resolution camera is located at end-effector and another high resolution camera is mounted at a fixed position. According to different demands of inspection task, there are two automatic scanning modes of surfaces: rough scanning and fine scanning. Rough scanning or wide-range scanning is quick and convenient for quick damage position location; Fine scanning is targeted for full volume and close visual inspection. The rough scanning mode requires a camera with large field of view while fine scanning requires a camera with high optical resolution to provide high-resolution images of the surface. The status of first wall is viewed by real-time monitoring or off-line images sequences. The position of defect tiles can be easily determined for first wall maintenance tasks.

Vision capabilities bestow a robot with sophisticated sensing mechanism which facilitates it to respond to various environments in an intelligent and flexible manner. Apart from touch, proximity and force sensing, vision is recognized as the most powerful sensor in terms of capability and performance improvement. Many applications exist today showing the need of integration between robotic system and machine vision system. This application is based on the idea of controlling the robotic system and also various inspection tasks depending on the amount of data that have been extracted from the captured image via camera. This chapter gives fundamental concepts of camera placement, different visual-servoing techniques and methodology employed in present work to predict the fault on the tile surfaces.

## 5.2 Types of Visual Servo Control

The main objective of visual servo control is to use the information acquired by vision sensors for feedback control of pose or motion of a robot. The vision sensors such as camera provide visual input. Such input is used as the feedback and different feats of the video/image is used to estimate the pose and motion of the robotic manipulator. There are different types of visual servo control which can be categorized as

1. Image based visual servo
2. Position based visual servo
3. Hybrid visual servo
4. Target tracking

Vision based robotic servo control is divided in to three sub-processes

1. Extraction of object of interest/region of interest (ROI)
2. Vision based control planning
3. Vision based actuation of robotic manipulator.

A brief description of the image and position based servo control approaches is first presented.

The visual control approach requires a digital camera along with the image acquisition and processing modules. In robotic manipulator either the camera is placed directly on the end-effector or at a fixed location. The basic block diagram of image based visual servo control (image based visual servo control) is shown in Figure 5.1. It comprises of two control loops. One of joint angle control of robot and other is vision based control loop. In joint angle control loop, joint angle sensors are used as the feedback device whereas in vision based control loop, image feats are used as the feedback signal. In this method, error is computed directly on the image of the feats extracted on the 2D image plane, without going through a 3D reconstruction. The robot should move so as to bring the current image feats to their desired values. The feats of this scheme are that it has computational advantages. In this scheme, the positioning accuracy of the system is less sensitive to camera calibration error and measurement noise in the images.

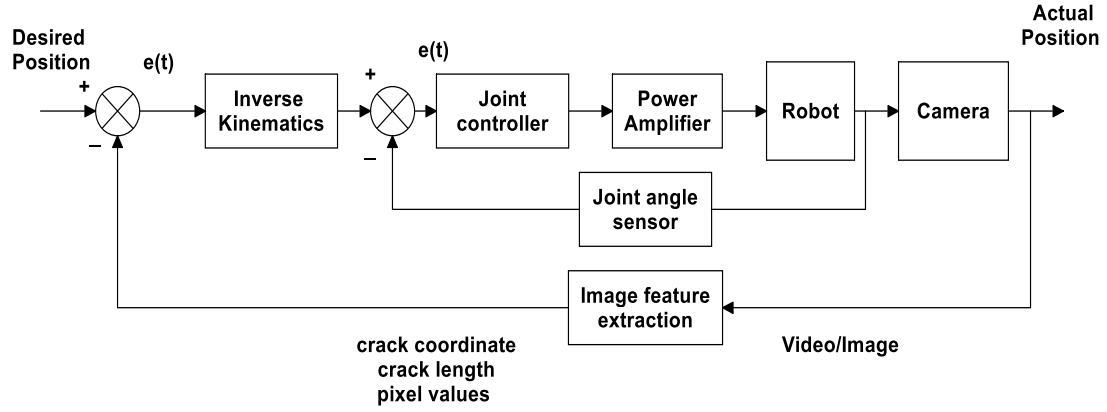


Figure 5.1 Simple block diagram of image based visual servo control

Image based servo control has the following steps

1. Image acquisition
2. Feat extraction
3. Comparison to generate error signal
4. Generation of motion of the camera/robot

The aim of image based visual servo control is to minimize a set of feat errors  $e(t)$  defined in the image data as  $e(t) = s(t) - s^*$ , where  $s(t)$  is a time varying vector of the visual feats and  $s^*$  represents the desired values of the feats which can be considered as constant when the target is fixed (graphite tile that the robot is approaching). The changes in  $s(t)$  depends on both camera motion and target motion. A velocity controller is designed by finding relationship between the time variation of  $s$  and the camera velocity as follows

$$\dot{s} = Lv_c \quad (5.1)$$

where  $L$  is a interaction matrix or feat Jacobian of size  $k \times 6$  where  $k$  as number of visual feat and  $v_c$  is camera velocity. The relationship between the camera velocity and time variation of error is obtained as:

$$\dot{e} = Lv_c \quad (5.2)$$

Considering  $\dot{e} = -\lambda e$ , we can write

$$v_c = -\lambda L^{-1}e \quad (5.3)$$

Where  $L^{-1}$  is the Moore Penrose pseudo inverse of  $L$ .



A robot controller takes joint position and velocity as inputs. Therefore, a relationship between a camera velocity and joint velocity  $\dot{\theta}$  is required.

For this, transformation of camera velocity to end-effector frame is used as follows.

$$v_e = {}^C_E [T]^{-1} v_c \quad (5.4)$$

Where  ${}^C_E [T]^{-1}$  is the inverse of velocity transformation matrix between camera frame and end-effector frame, which is constant matrix or eye-in-hand systems. This is given as

$${}^C_E [T] = \begin{bmatrix} {}^C_E R & [{}^C_E t]^* {}^C_E R \\ 0 & {}^C_E R \end{bmatrix} \quad (5.5)$$

Where  ${}^C_E R$  and  ${}^C_E t$  are rotational and translational relationship between camera frame and end-effector frame. Here

$${}^C_E t = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (5.6)$$

The final relation between angular velocity and feat error vector is given by

$$\dot{\theta} = -\lambda [L {}^C_E T^E J]^{-1} e \quad (5.7)$$

Where  ${}^E J$  is a Jacobian matrix written in end-effector frame. This is the control law of image-based visual servo. During each control loop, program calculates the feat errors and estimates interaction matrix  $L$  and ends the resultant angular velocity  $\dot{\theta}$  to the robot controller in order to minimize the feat errors. The constant  $\lambda$  is a gain value selected from 0 to 1. For a 3-D point, (X, Y, Z) in the camera frame its projection in 2-D image plane is given by

$$x = X / Z = (u - u_0) / f_u \quad (5.8)$$

$$y = Y / Z = (v - v_0) / f_v \quad (5.9)$$

Where  $u$ ,  $v$  gives the image point co-ordinates in pixels units and  $u_0$ ,  $v_0$  are the camera intensity parameter and  $f$  is focal length. The interaction matrix  $L$  is given as

$$L = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{z} & \frac{y}{z} & 1+y^2 & -xy & -x \end{bmatrix} \quad (5.10)$$

where  $z$  is the depth of point relative to camera frame.

In order to control robot in camera frame at least three points are required. In real visual servo system it is impossible to obtain the perfect  $L$  matrix due to imperfect depth value and intrinsic parameter of the camera. There are several methods available for estimating the interaction matrix. Figure 5.2 shows the position based visual servo control. In this scheme, along with image feat extraction, pose estimation is done and the Cartesian control law is used to control the inverse kinematics of the robot. In other words, the position based visual servo controls the error between the desired and actual poses and works directly in the 3-D workspace. The main advantage of this approach is the end-effector position can be control relative to the target directly. The main disadvantage is that the position and motion estimation depends on the camera calibrations errors and target model accuracy. In the present work due to the ease of image based visual control techniques the approach is used throughout.

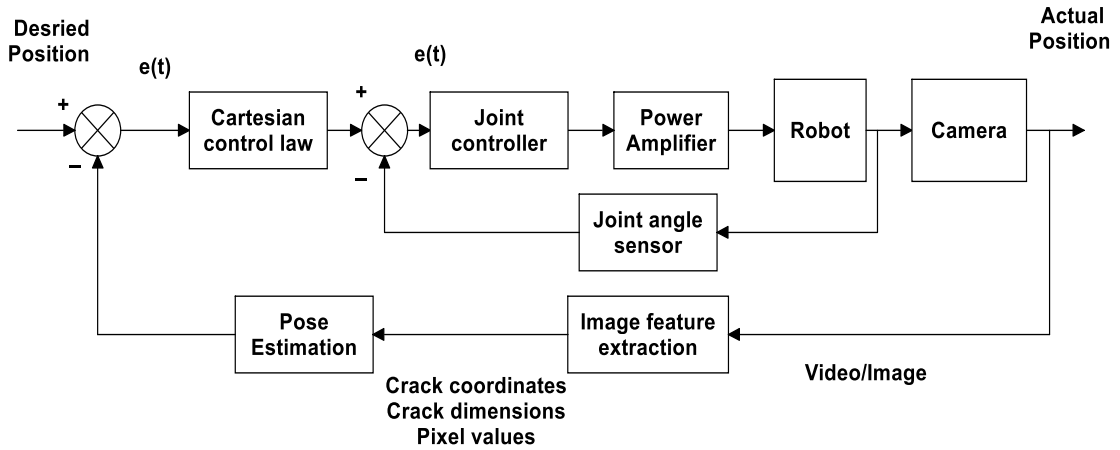


Figure 5.2 Simple block diagram of position visual servo control

The pose of the target can be described with Cartesian coordinates  $T(px_o, py_o, pz_o)$  with respect to target frame of reference. The Euler angles of the frame  $\theta_x, \theta_y, \theta_z$  are defined with respect to camera frame. The rotation matrix from camera frame to target frame can be developed by first rotating x-axis of camera by  $\theta_x$  followed by rotating y-axis of camera frame by  $\theta_y$  and z-axis of camera frame by  $\theta_z$ . Accordingly the rotation matrix from target frame to camera frame is written as

$$[R] = \begin{bmatrix} c_v c_z & -c_v s_z & s_y \\ c_x s_z + s_x s_y s_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - s_x s_y s_z & s_x c_z + s_x s_y s_z & c_x c_y \end{bmatrix} \quad (5.11)$$

By assuming the target co-ordinates  $p_x$ ,  $p_y$ ,  $p_z$  the relationship between the target and camera frames can be written as

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} [R] & \begin{matrix} px_0 \\ py_0 \\ pz_0 \\ 1 \end{matrix} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (5.12)$$

Where  $x_c$ ,  $y_c$ ,  $z_c$  are the co-ordinates is the same point on the camera frame

A hybrid visual servo refer to 2 ½ -D visual servo control. It evaluates the control errors partially in 3-D workspace and partially on the 2-D image plane.

### 5.3 Image acquisition

Image acquisition is one of the primary steps of image based visual servo control. In this step, a camera is used to collect visual information about different objects. Image acquisition comprises of different steps such as

1. Camera placement
2. Camera model
3. Vision data acquisition device
4. Feats of the camera
5. Lighting sources

There are different ways of camera placement. Figure 5.3 sows the different ways of camera positioning. The eye-in-hand camera is mounted on the end-effector to provide a close view of the target, while eye-to-hand camera is installed beside the robot to monitor the whole workspace with a broad seen of the target

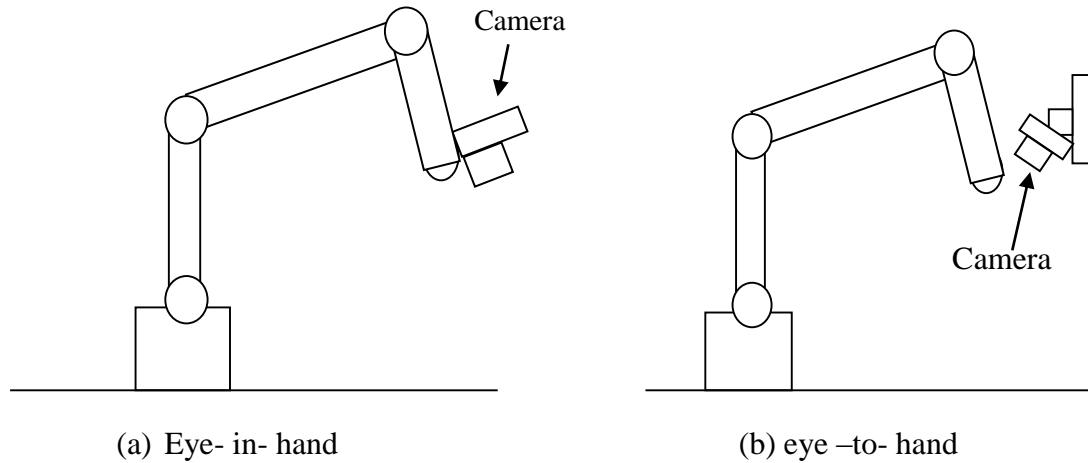


Figure 5.3 Types of camera placements

## 5.4 Image Processing

Digital image processing is a computerized process that helps to enhance the properties of an image into a usable form. The interest in digital image processing methods stems from two principal areas of application. These are the improvement of pictorial information for human interpretation and the processing of scene data for autonomous machine perception. There are two types of image processing approaches: (i) Low-level processing involve primitive operations that consist of image acquisition, image compression and a pre-processing method for noise filtering, edge extraction and image sharpening or the enhancement of certain object feats that are relevant to understand the image. (ii) High-level processing attempts to imitate human cognition (making sense of the image) and the human ability to make decisions according to the information contained in the image.

There are four modules in defect detection on tiles: (1) The noise reduction module, (2) the edge defection module, (3) the unit transformation module and (4) the defect detection and quantification module. The system is designed to prepare representative images from actual inspections. These representative images can be used to explore the image feats when selecting a suitable method for the noise reduction and edge detection modules. Moreover, the unit transformation module requires a representative image to determine an equation to convert the image unit from pixels to millimeters. The conditions of data collection must be specified to reduce the deviation of results during the development of the system, such as the camera specifications, pixel size and camera distance.

The noise reduction module is the first process of the system after the image is inputted into the system. This module supports the function of image adjustment to reduce

unnecessary backgrounds or noise from a digital image by using the pre-processing of a digital image processing technique. When exploring representative images for tiling inspection, non-uniform illumination caused by tile reflection is a major noise that must be reduced because it affects the step of edge detection module.

Edge detection module supports the function of edge detection of interested object on the image for defecting feat analysis in the defect detection and quantification module. The edge of tile on image is interested object in case study of tiling inspection. Generally, edge detection shows the image location of discontinuities in the gray level of an image. There are various edge detection methods that are used for different applications. Next, the coordinates of each vertical line and horizontal line are used to analyze the defect positions and to calculate the defect value in the defect detection and quantification module.

The unit transformation module supports the function of transforming the pixel units into millimeters for comparing in the same unit of an actual object. The ratio of mm-per-pixel depends on the photography conditions, such as the camera specification, camera distance and pixel size.

The defect detection and quantification module supports the function of defect feat analysis in the proposed system to specify the defect positions and quantify the defect value.

Figure 5.4 shows the flowchart of the various stages of image processing for identification of crack length and its coordinates. The various stages are explained first. In a digital camera system all data is in binary form and is stored in a computer file or a memory device. An image with different gray-levels  $f(x, y)$  at each pixel location is known as gray image. By super-imposing three images of red, green, blue each with varying intensity results in color image. A binary image is an image with each pixel either fully light or fully black. A gray image is converted to binary image using histogram of the image along with a cutoff value called threshold.

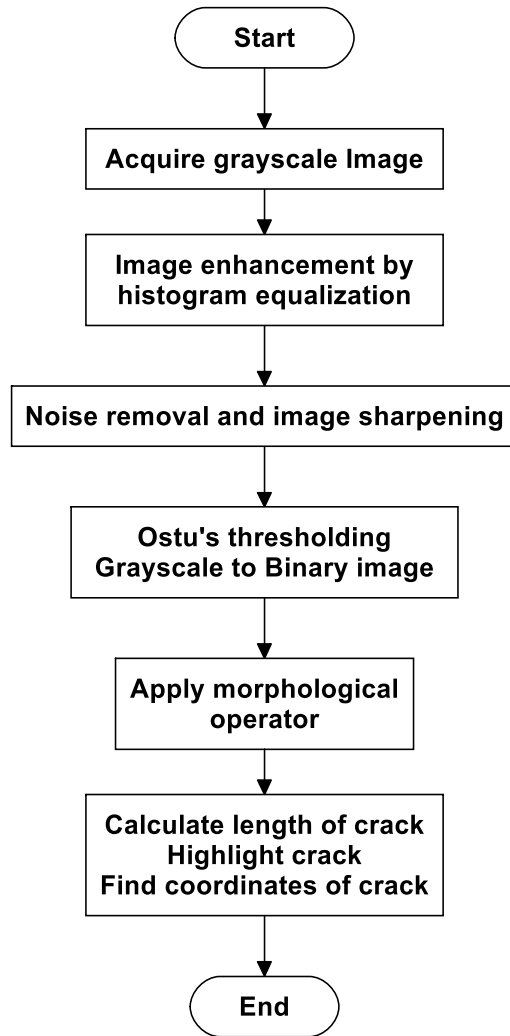


Figure 5.4 Flowchart of the image processing steps

A histogram of the pixel gray level determines the distributions of different gray levels. Image processing is divided to several steps such as histogram analysis, thresholding, masking, edge-detection, segmentation etc.

#### 5.4.1 Histogram Equalization

A histogram is a representation of the total number of pixels of an image at each gray level. Histogram information is used for thresholding as well as for classification of different images. An image of size  $p \times q$  has pixel intensity  $f(x, y)$  where,  $x = 1, 2, 3, \dots, p$  and  $y = 1, 2, \dots, q$  and pixel intensity values  $f(x, y)$  varies from 0 to  $L-1$ . Histogram is calculated as  $H(f) = N_f$ , the number of pixels in an image with intensity level  $f$ . From histograms various feats can be extracted. Histogram equalization is contrast enhancement technique in a spatial domain in image processing using histogram of image. Histogram

equalization usually increases the global contrast of the processing image. This method is useful for the images which are bright or dark.

### 5.4.2 Noise Removal and Image Sharpening

Noise removal and image sharpening techniques are one of the image enhancement techniques which highlight edges and fine details in a digital image. Image sharpening can be represented as

$$S_{i,j} = x_{i,j} + \lambda F(x_{i,j}) \quad (5.13)$$

Here,  $x_{i,j}$  is the original pixel value at coordinates  $(i, j)$  and  $F(.)$  is a high pass filter.  $\lambda$  is a tuning parameter.  $S_{i,j}$  is the sharpened image

### 5.4.3 Image Thresholding

Thresholding is the process of diving an image into different portions by picking a certain greyness level as a threshold by comparing each pixel with the threshold value and assigning the pixel to the different portions. Image thresholding can be mathematically represented as:

$$g(x, y) = \begin{cases} 0 & f(x, y) < T \\ 1 & f(x, y) \geq T \end{cases} \quad (5.14)$$

Ostu thresholding technique is the gold standard of thresholding technique in image processing. It is based on the simple concept of minimizing weighted within-class variance.

### 5.4.4 Morphological Operators

Morphological operators are the operations performed on a shape of an image. Morphological operators are used for object detection technique. These operators are based on set theory. There are different types of morphological operators such as

1. Dilation, where the background pixels that are connected to fore ground (object) are changed to the fore ground.
2. Erosion, where the fore ground pixels that are connected to background pixels are eliminated.
3. Opening, which is erosion operation followed by dilation .This causes smoothness of convex parts of objects
4. Closing, which is a dilatation operation followed by erosion .This causes smoothness of concave parts

### **5.4.5 Edge detection techniques**

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges.

It refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of object in a scene. That is an edge is identified by a rapid change in the pixel intensity of the image. Such rapid changes are detected by using first and second order derivatives. An edge is a boundary between the object and its background. The goal of every detector is to avoid false edges and detected edges should be closest to true edges. There are different kinds of edge detection methods in image processing literature. Some of the well-known edge detection techniques are

1. Sobel
2. Canny
3. Prewitt
4. Roberts
5. Laplacian of Gaussian
6. Zero crossing

Further details of these methods can be found in Appendix B with a crack detection example.

### **5.4.6 Template Matching**

Template matching is a technique of object recognition if a suitable line drawing of the scene is observed. Here the total number of lines, points, and inter-connection can be matched to a model. The technique requires prior knowledge of the object model for matching. Apart from this, discrete Fourier transforms, Hough transform can be used to complete the broken contours of the object.

## **5.5 Case study of simulated wall tiles**

In order to illustrate the approach of identifying faulty surfaces on a group of objects, a rectangular wall with a sequence of symmetrically placed uniform rectangular tiles mounted from left to right and top to bottom is considered. Some of the tile surfaces are



provided with markers spots indicating cracks in realistic conditions. Task is to identify such a abnormal tiles along with their locations with respect to camera (image) frame. Classical visual servoing system allows for carrying-out point to point motion of the robot using visual information. Image based visual servoing system has the following assumption

1. Robot is a perfectly positioning device.
2. No system dynamics is required.

By direct visual servoing the internal control loop of servo motors is removed. Hence, the visual servo control law directly provides the torque to the robot joints. Followings are the items used for testing (i) digital camera (ii) image processing software (Matlab/ LabVIEW) (iii) Computers programs.

### 5.5.1 Features of the camera

In order to capture the image, a digital camera (BASLER aceA640-90gm) is used. The specifications of the camera are summarized in table 5.1

Table 5.1 Camera specifications

Name of the camera	BASLER aceA 640-90gm
Resolution	659 x 494
Frame rate	90 fps
Interface	GigE/Ethernet
Power requirement	12V DC
Pixel size	7.4 $\mu\text{m}$ x 7.4 $\mu\text{m}$
Pixel bit depth	12 bit
Color	Monochrome
Sensor	Sony ICX424 CCD sensor
Size of sensor	4.88 mm x 3.66 mm

This camera is placed at a fixed location to observe the entire wall surface. The images are taken by this camera are stored as .jpg on a personal computer. Figure 5.5 shows the placement of camera system in eye-to-hand configurations for identification of crack on the tile surface. Here, the robot model is also shown with an end-effector. The camera field monitors both the end-effector and object surface. Also the object surface has total

twelve tiles placed four in each row. Intentionally some tiles surfaces are given with marker curves indicating cracks.

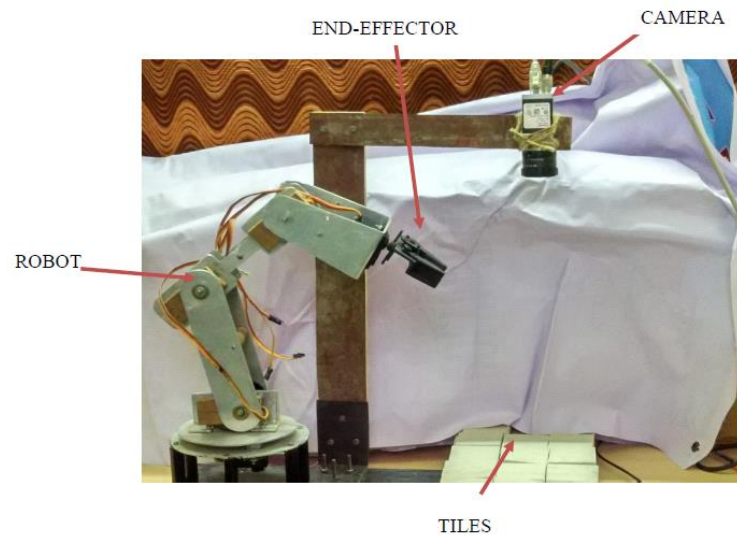


Figure 5.5 Developed experimental setup with camera placement

The camera is interfaced with the desktop PC via Ethernet cable and it is powered with 12 volt DC power supply. Figure 5.6 shows the flowchart of the methodology employed to predict the crack levels.

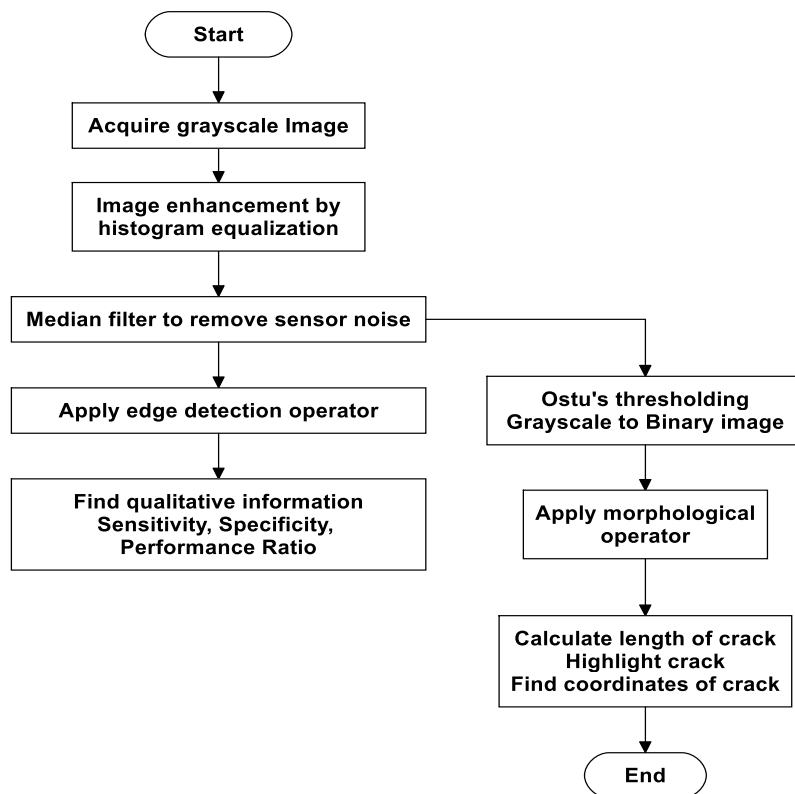


Figure 5.6 Flowchart of image processing steps for vision based fault identification

### 5.5.2 Image processing with a group of tiles having random cracks

Image of a group of tiles with faults in few was kept within the reach of the robot workspace and the image has been acquired by the camera. All the tiles have been numbered for identification of the faulty one. The Figure 5.7 shows the acquired image of the group of tiles with random faults marked for simulation sake. Further in order to detect the crack and extract various feats Matlab toolbox has been used for simulation. The processing part is carried out in various edge detection techniques. The image acquired is in grayscale form. Further the noise and disturbances has been removed (Figure 5.8) and image has been enhanced. Sharpening of images has been done by applying the high pass filter so as to increase the sharpness of the image by eliminating some of the lower frequencies from the edges. The grayscale image is converted to binary image which will help to highlight the crack, keeping rest of the feat and text hidden. Under image morphology dilation of the rectangular crossover in the tile has been eroded (see Figure 5.9). Then the image segmentation is performed with a purpose to separate the information contained in the image into smaller entities that can be used for required purpose as shown in Figure 5.10. This segmentation includes edge detection technique, texture analysis etc.

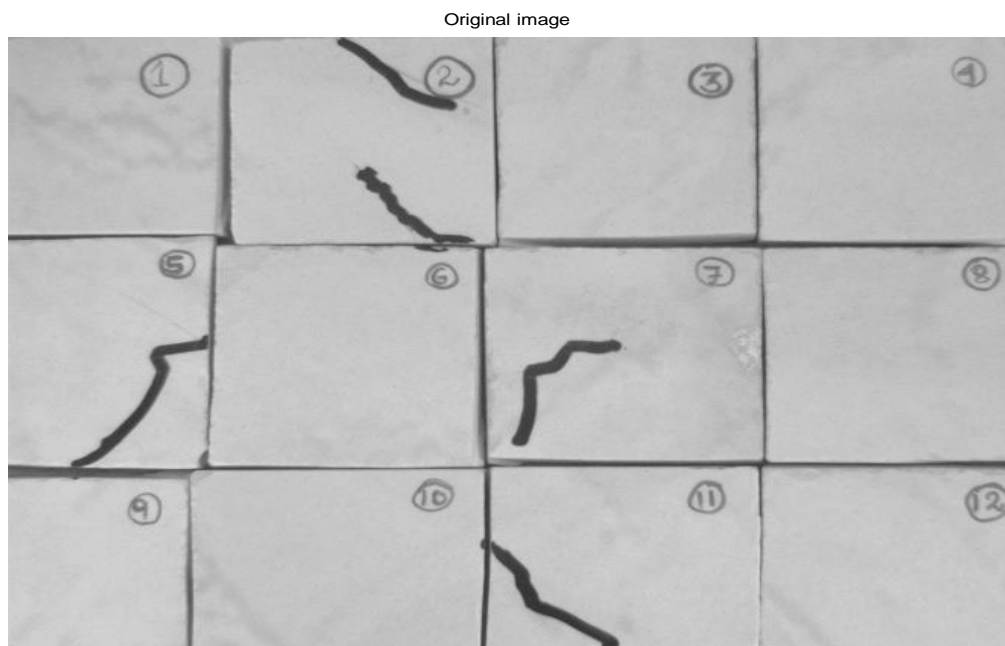


Figure 5.7 Captured image

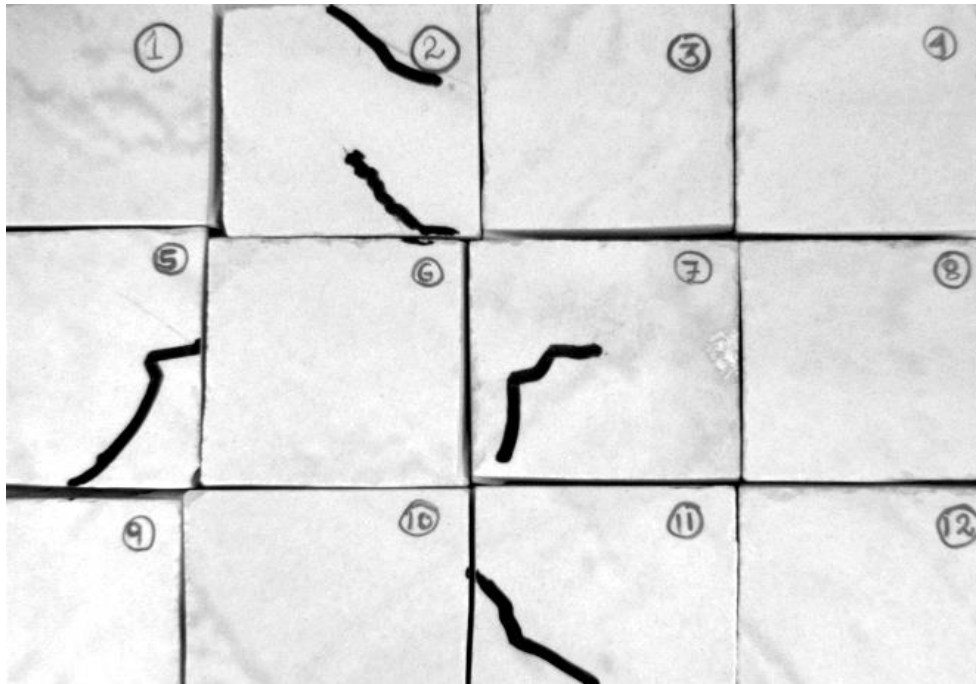


Figure 5.8 Enhanced image

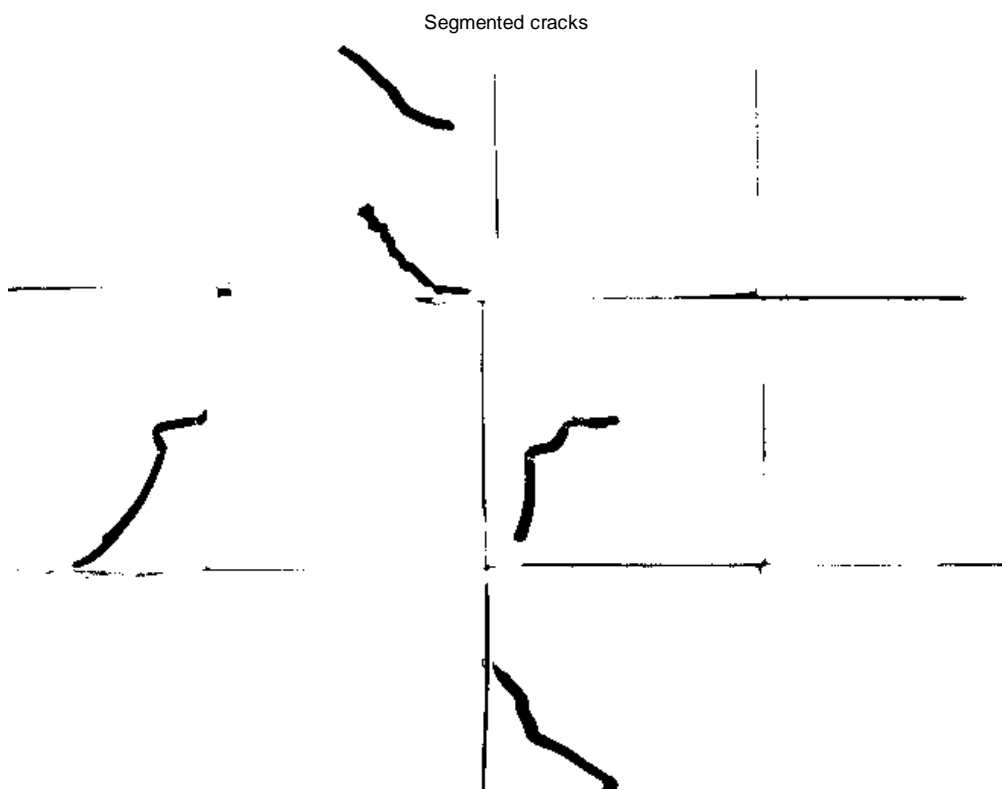


Figure 5.9 Segmented images

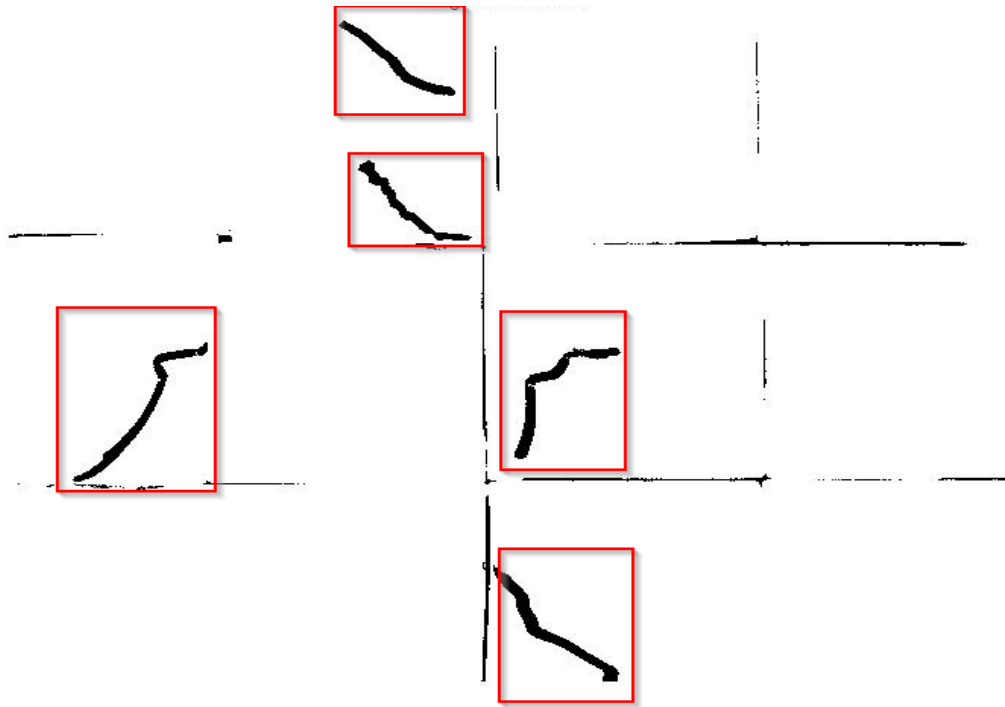


Figure 5.10 Crack detection

Table 5.2 shows the crack dimensions and time taken for simulation using the Matlab program.

Table 5.2 Computational results for crack detection technique

Tile Number	Crack dimension (cm)	Time taken to compute (sec)
2	2.9	1.0267
5	2.75	0.999238
7	2.69	0.778542
11	2.89	0.96582

### Image Processing using LabVIEW

In LabVIEW, the same images have been captured using vision assistant toolbox. A set of pre-stored images of tile surfaces are processed. If an image of size  $p \times q$  has pixel intensity  $f(x, y)$  where  $x=1,2,3,\dots,p$  and  $y=1,2,3,\dots,q$  and pixel intensity values varies from 0 to  $L-1$  then the histogram can be calculated using  $H(f) = N_f$  in which  $N_f$  denotes the number of pixels in an image with intensity level  $f$ .

The PC containing LABVIEW vision assistant toolbox captures the image and converts into binary form. Vision Assistant is a tool for prototyping and testing image processing applications. To prototype an image processing application the custom algorithm can be

built. Such script in feat records every step of processing algorithm. After completing the algorithm it can be tested on other images.

The methodology applies the edge detection operation to identify the new edges resulting from cracks. The objective is to distinguish the different images having slight changes in the crack patterns. This may be of interest in replacement of defective objects (tiles) even they have visually minute size cracks or retaining a tile without replacement even it has appreciable crack size as seen with naked eye. Figure 5.11 shows the field of view of the camera as observed from vision assistant image capturing module for tiles having no crack (initial condition). The corresponding image histogram is shown in Figure 5.12 for the entire field of view.

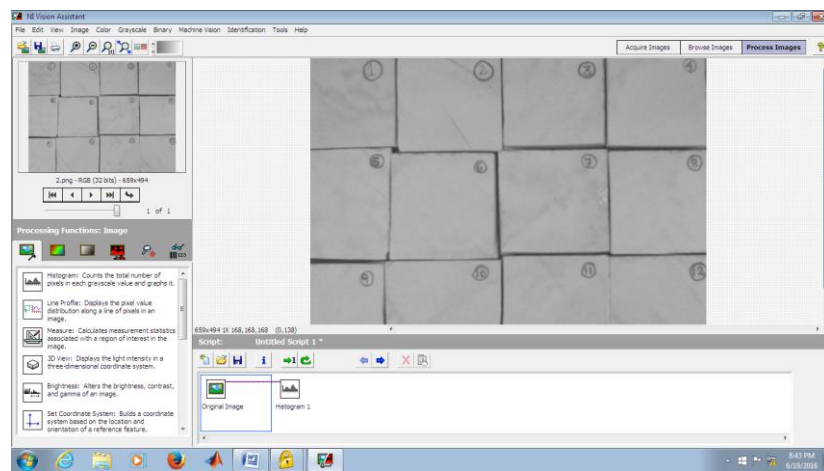


Figure 5.11 Vision Assistant toolbox screenshot for tiles without crack

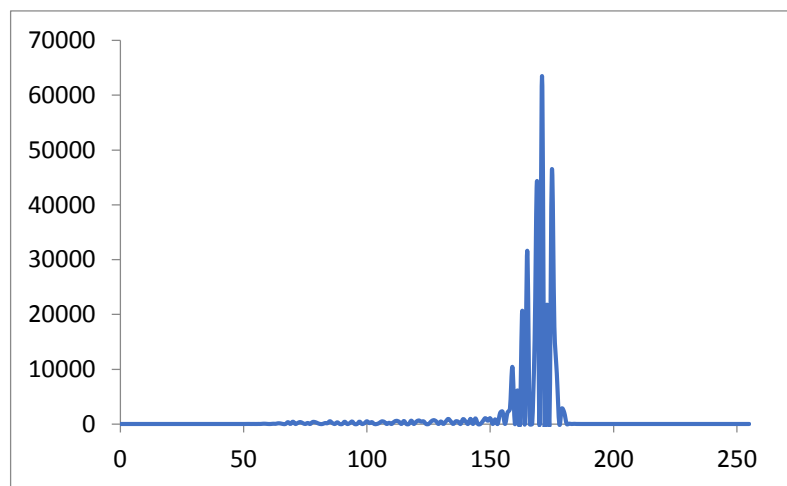


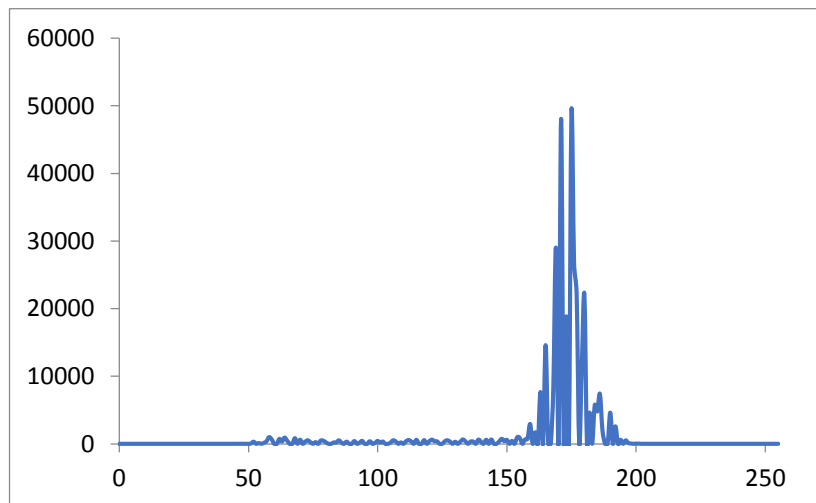
Figure 5.12 The corresponding histogram for the grayscale image

A similar image histogram (Figure 5.13(b)) is obtained for the image (Figure 5.13(a)) with tiles having some random cracks on specific tile locations. It is observed that there is a

marked difference in histograms which can be quantified to state the extent of damage for replacement.



(a) Screenshot of entire tile surface with random faults



(b) Corresponding image histogram from Vision Assistant

Figure 5.13 Histogram Analysis

In the next phase, the field of view is focused to a specific tile for its further analysis. In present case tile no 5 is selected and its crack nature is slightly altered with various crack paths as shown in Figure 5.14.

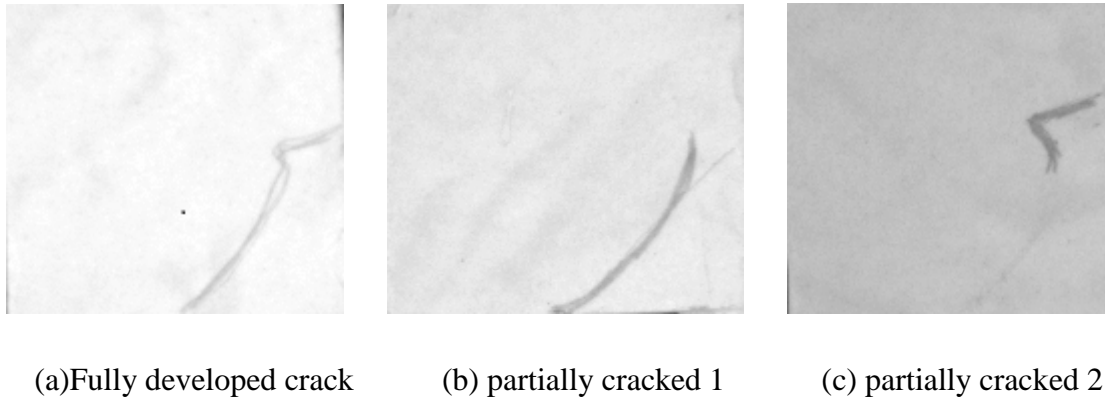
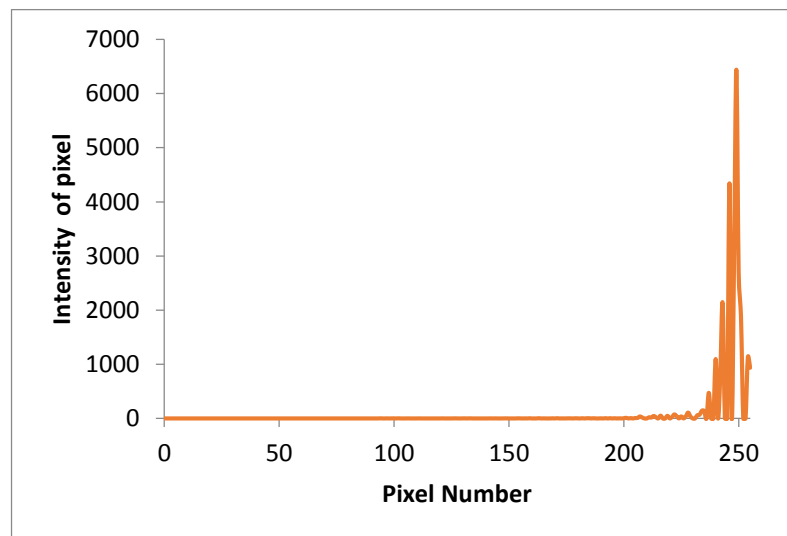


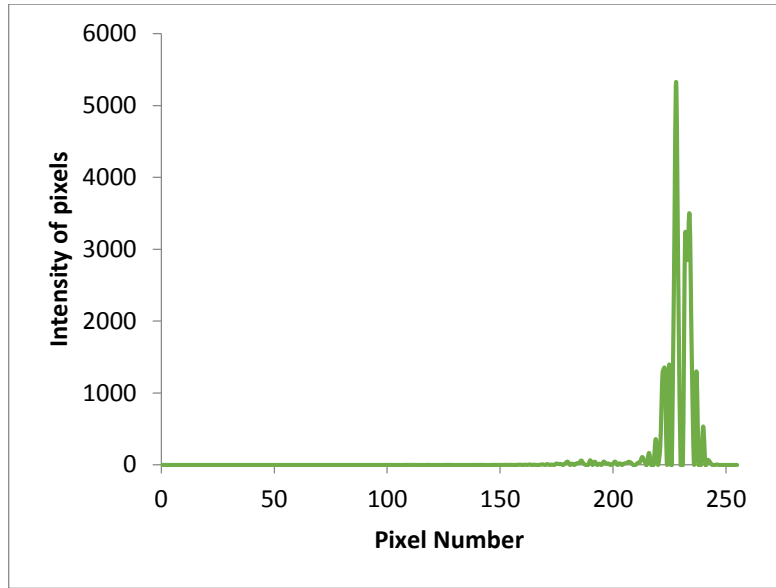
Figure 5.14 Gray scale Images of the tile 5

The corresponding histograms for each of the above images are shown in the Figure 5.15. These histograms are plotted from the image analysis data obtained from vision assistant toolbox.

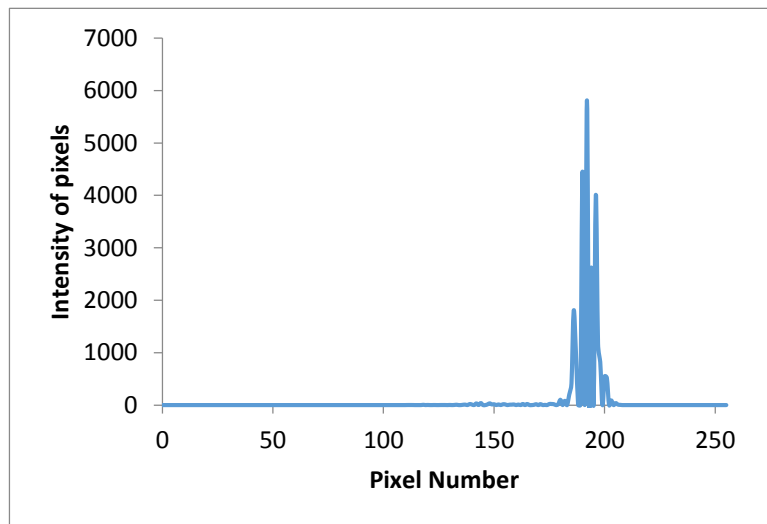


(a) Case1 Full crack





(b) Case 2 Partial crack 1



(c) Case 3 Partial crack 2

Figure 5.15 Histogram analysis of tile 5

From the histogram analysis, it is observed that there is a change of intensity of pixels in images with different nature of cracks. Using these histograms one can distinguish the various images based on the histogram statistical parameters such as mean, variance, kurtosis etc.

Figure 5.16 shows the block diagram of image acquisition without using Vision Assistant sometimes called as low-level programming. In this diagram the image is scanned entirely from left to right and top to bottom to obtain the co-ordinates of the cracked region.

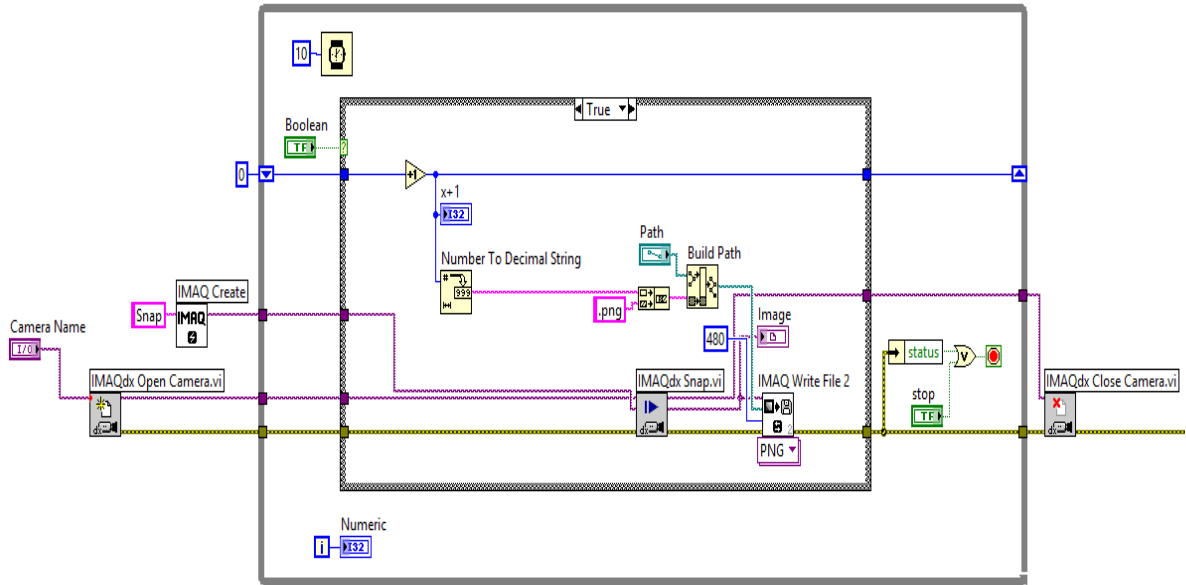


Figure 5.16 Block diagram of image acquisition using LabVIEW

Figure 5.17 shows the corresponding front panel illustrating the coordinates of the required cracked regions.

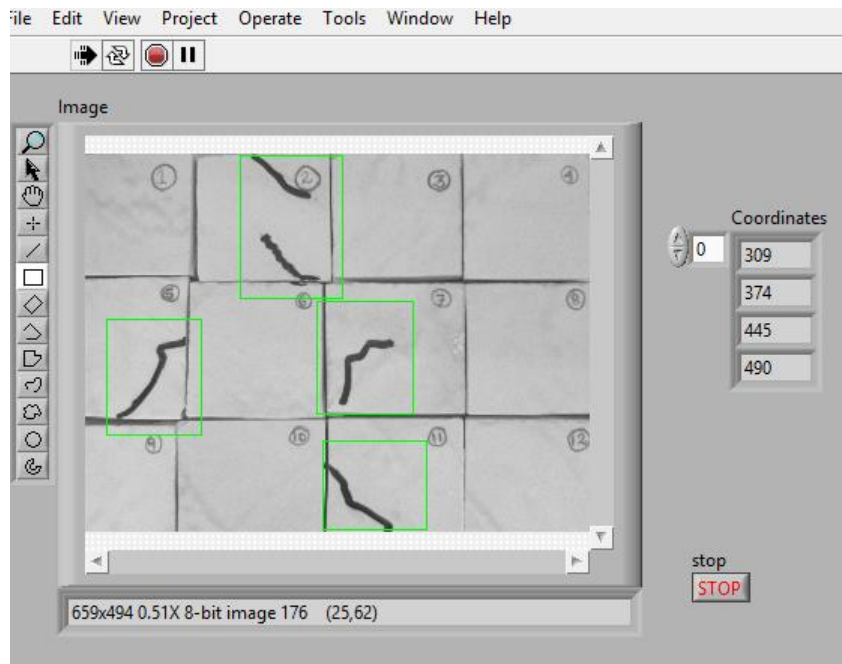


Figure 5.17 Front panel detecting the cracked regions

Table 5.3 shows the computational results for crack detection algorithm which includes crack dimension, time for computation and pixel coordinates of the crack.

Table 5.3 Computational results for crack detection technique

Tile Number	Crack dimension (cm)	Time taken to compute (sec)	Coordinates
2	2.9	1.0267	(215,189,323,178)
5	2.75	0.999238	(21,202,143,343)
7	2.69	0.778542	(311,215,417,339)
11	2.89	0.96582	(317,368,417,494)

MeasureAs all the tiles are of uniform dimensions, it is easy to identify the equivalent x, y coordinates in mm of both the cracks as well as the edges of the corresponding tile. Figure 5.18 shows a conceptual view of tile with keyways slots to insert them on to the wall.

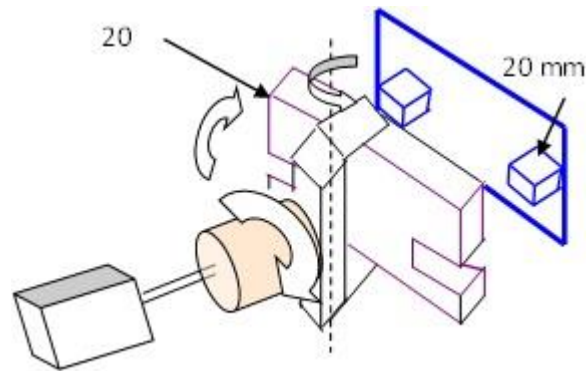


Figure 5.18 Schematic of tile placement on the wall

## 5.6 Transformation from camera to end-effector

In the visual servo control, the final step is to transform the coordinates of the object from camera frame  $\{C\}$  to end-effector frame  $\{E\}$ . Let the object frame be  $\{O\}$  and the robot base frame be  $\{R\}$  as shown in the Figure 5.20.

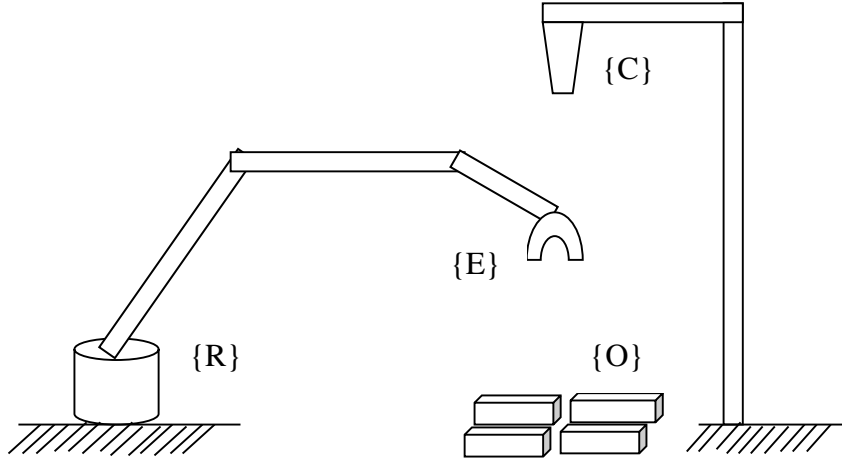


Figure 5.19 Important frames in eye-to-hand configurations

The transformations required for this eye-to-hand system are described below:

$${}^E T_R {}^O T_E = {}^E T_R {}^C T_E {}^O T_C \quad (5.15)$$

Here the transformation matrix  ${}^E T_R$  is obtained from the kinematics of the manipulator as described in chapter 3

The end-effector to camera matrix is fixed at every location as a camera is held stationary. The resulting transformation from end-effector to object frames is given by

$${}^O T_E = [{}^C T_E] [{}^O T_C] \quad (5.16)$$

Here  ${}^O T_C$  is obtained from image processing module while  ${}^C T_E$  is formulated by measuring the distances between the two frames in x, y, z direction. The matrix  ${}^C T_E$  is obtained as follows:

$${}^C T_E = {}^C T_R {}^R T_E \quad (5.17)$$

In the present case, the matrix  ${}^C T_R$  is measured according to the Figure 5.28 and is written as follows:

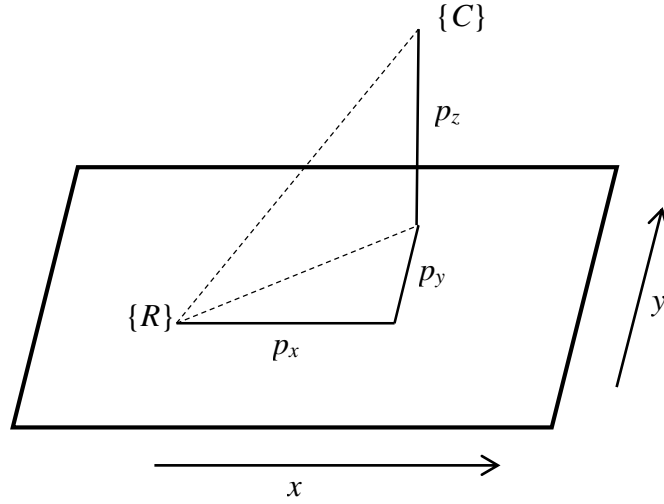


Figure 5.20 Location of camera with respect to robot base

$${}^C_R T = \begin{bmatrix} 1 & 0 & 0 & 275mm \\ 0 & 1 & 0 & 330mm \\ 0 & 0 & 1 & 50mm \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

The transformation from reference frame to end-effector is given from kinematics as follows

$${}^R_E T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

Where  $n_x, o_x, a_x$  etc. are the normal, orientation and approach vector of the end-effector with respect to fixed base.

## 5.6 Conclusion

This chapter provides a comprehensive analysis of vision based crack detection mechanism where a camera is used to capture images of tiles (either normal or cracked). MATLAB based simulation results and LabVIEW based real-time experimental results have been provided in this chapter. Also the frame transformation matrices are introduced for the visual servo control of the end-effector to reach the damaged or defected object. Next chapter introduces the real time control issue of the joint servos based on the results of present chapter using the scaled prototype of the manipulator system.

## Chapter 6

# Design, Fabrication and Experimental Analysis

This chapter basically deals with fabrication and testing of the articulated 5 DOF manipulator for replacements activities. It gives elaborate description of robot of fabrications details, and electronic and sensor interfaces. Initially few preliminary design considerations have been take as per the feasibility and ease of the design and further modified to suits the present requirement as per available specification. The final design is made keeping in mind various factors such as expensiveness, ease of manufacturing, weight, better availability of electronic components and also the design should be within stipulated time.

While doing the concept design, Solidworks are used as design tools along with motion simulation in ADAMS software. The dimensions stated in the design specification are considered while fabricating the final prototype. Further, for actuation of this fabricated model with joint servos, electronic interface is used from computer. The input given to the manipulator is the results from image processing as well as through inverse kinematics. Further with the command the robot moves to the desired location and performs the required task. Thus a thorough analysis of each and every module has been carried and tried to coordinate with each other which gives uniqueness to the whole project.

The fundamental features of the final design used for manufacturing the robot have been presented first.

### 6.1 Concept Design of the articulated robotic manipulator

The manipulator is primarily designed using five parts as described in the chapter 3 under kinematics. Those five links are: the base, the shoulder, the elbow which is divided into two parts upper-arm and forearm and lastly the end-effector housing. All the links are interconnected by revolute joints thus making it an articulated arm. The joint between the base part and the shoulder forms the rotational motion, the further joints between shoulder

and elbow creates rotational motion, the joint between the upper –arm and forearm creates the elbow rotation and finally at the end-effector we have two wrist rotational motion one is pitching and another is rolling .The material used for fabrication is Aluminum Alloy sheet of 3mm thickness throughout and the fabrication of each of this components from solid models are explained below.

### 6.1.1 Base

The base with Figure 6.1 of the manipulator is the most crucial part as it has to take the whole load of the arm, thus attention has been taken to make it practically strong and sturdy. Further the material to be fabricated has been chosen to be strong enough so that it wouldn't deform under load and will be ease to withstand all shock and vibrations. Figure 6.1 shows the three projections along with isometric view of the base.

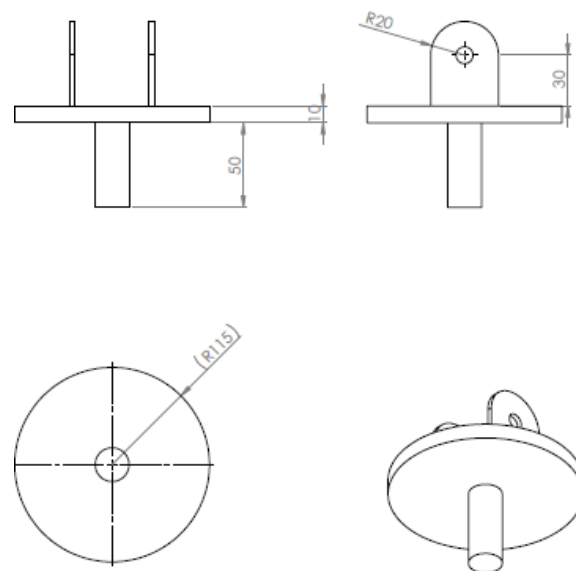


Figure 6.1 Drawing view and solid model of the base

### 6.1.2 Lower Arm

Like the base the body Figure 6.2 shows the dimensional views of the lower arm and care has been taken to make it very strong and robust as such it houses the servo motor to drive the link. The lower arm is driven by a servo at the joint connecting the base with it. The servo motors often have shafts with geared pinions. So it is desirable to generate the grooves with some wheels having toothed hole inside. In the present task plastic pinions are

considered as this being a low duty scaled model. The objective here is to transmit the motion of servos fully without loss at the joint in a smooth manner.

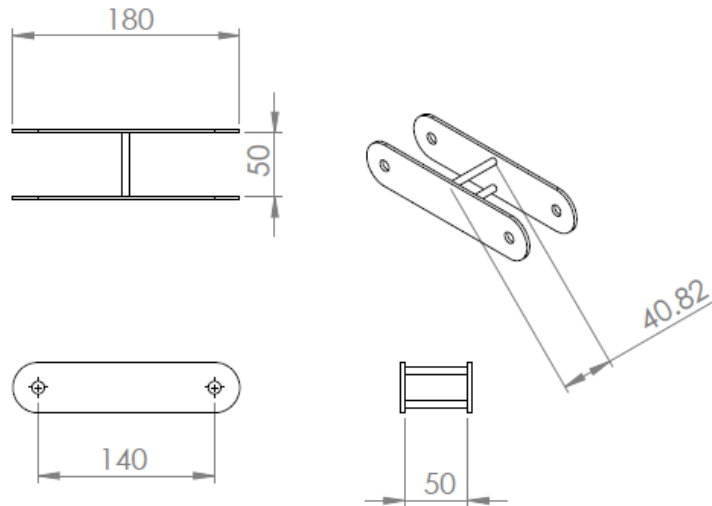


Figure 6.2 Drawing views and solid model of the lower arm

### 6.1.3 Upper Arm

The upper arm which is the link 2 is shown in Figure 6.3. It is designed to be somewhat lighter in weight from that of base and body. All this has been joined and interconnected with the servo motor directly at the joints.

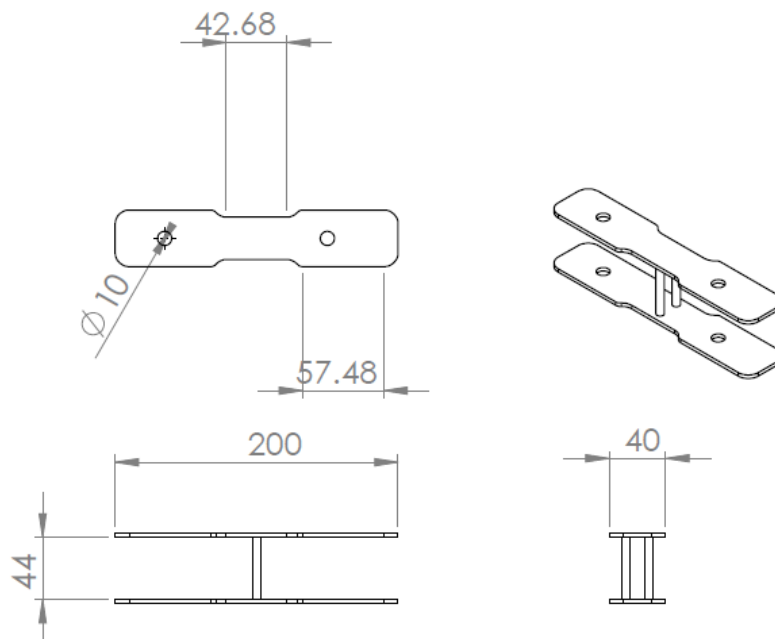


Figure 6.3 Drawing views and isometric view of the upper arm



As it is seen, a reduced cross section with the same length is considered as that of the previous link to minimize the weight.

#### 6.1.4 End-Effector Casing

The casing or hand for the end-effector is shown in Figure 6.4 is the last link of the manipulator. The gripper for pick and place has been secured on this link. Here, a servo motor will be placed to provide a roll motion to the gripper. The pitching motion is given by the servo placed at the beginning of the casing.

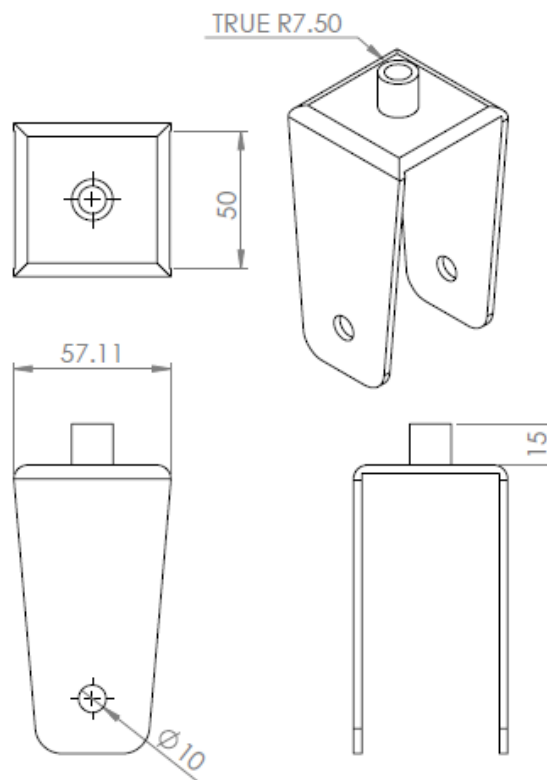


Figure 6.4 Drawing views and solid model of the end-effector casing

#### 6.1.5 End-effector

A two Finger gripper has been used as an end-effector. It works on the principle of parallelogram mechanism, which gives the fingers a better gripping capacity as they move parallel to each other. Currently the gripper can take a load of 300 grams but the weight

can be further extended with the use of high torque motors. The gripper has maximum opening of 5 cms. The gripper or end-effector is employed in the present task is meant only for the uncontrolled force gripping action. Therefore a readymade light weight gripper with fingers actuated by a micro servo may be directly employed at the end-effector casing. The Figure 6.5 shows the assembly model of the all five components without gripper.

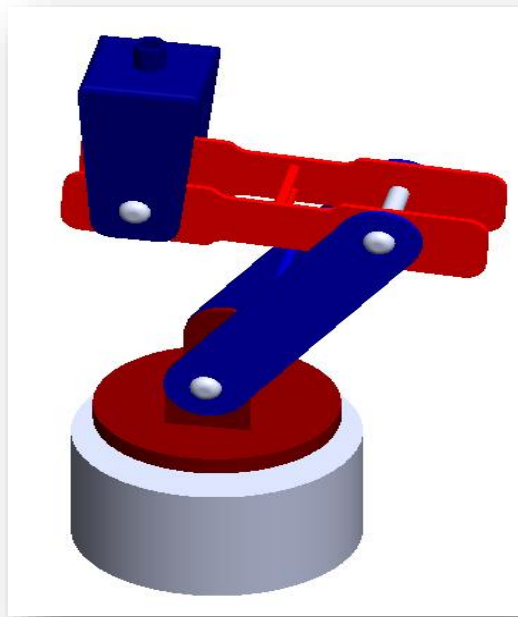


Figure 6.5 Assembly model in SolidWorks

The assembly-model can be used for performing the kinematic simulations and static stress analysis to understand further. A summary of the manipulator design parameter is provided in Table 6.1

Table 6.1 Manipulator summary

Parameters	Description
No of joints	5
No of DOF	5
Range of Motion	Wrist Pitch $-140^0$ - $+140^0$ Wrist Roll $-90^0$ - $+90^0$ Elbow $-90^0$ - $+90^0$

	Shoulder $-90^0$ - $+90^0$
Locomotion	Articulated Links
AC	6 DC servos motion
Weight	3.4 kg
Dimension	Base Radius -115mm Lower Arm-140mm Upper Arm-120mm End-Effectors-100mm
Gripper	2 finger gripper with high strength plastic
Payload	0.3 kg

## 6.2 Fabrication of the prototype

Designing a robotic system from the scratch is much a time consuming and lengthy process .Thus, various existing standard models have been studied in detail and researched before starting the fabrication work. Care has been taken to understand the problem concept and how feasible design has to be made for a better solution. Fabrication involved a lot of machining process, which was carried out in the institute workshop for about a year. All the primary processes involved in the fabrication of the robot has been explained in this section. These are machining of the robotic body, fabrication of the rigid base, specification of the servo drives, various electronic components, and vision sensor, and battery, further assembly of the machined parts and procured product, setting up the control system and programming it, as well and testing of the robotic arm.

### 6.2.1 Machining of the Robotic Arm

Various parts of the robotic arm witness forces at different direction and magnitude. Thus suitable materials has been selected to fabricate each part .Parts such as shoulder arm, elbow arm end-effector casing etc. However for the base part the wood has been selected because of easy machining and sturdy stability. Figure 6.6 shows the fabricated prototype prepared.

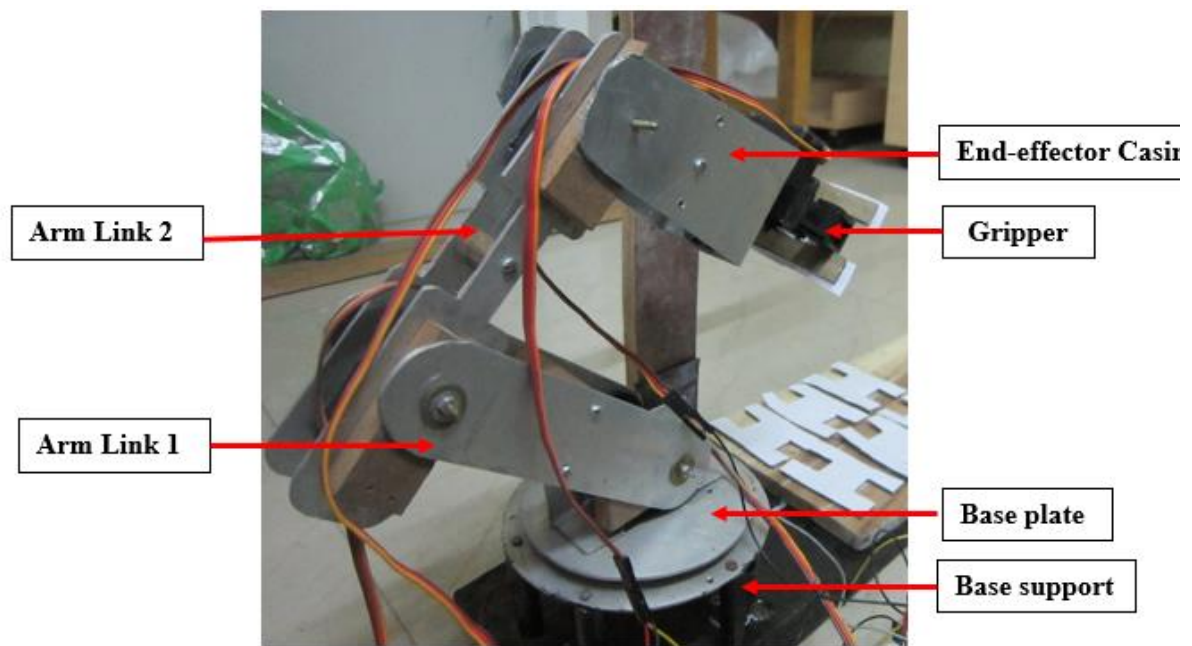


Figure 6.6 Fabricated Robot Arm

The details of various components and there material are given below:

Part Name : Base of the manipulator

Material : Thick wooden block

Machining Process : Hack-saw to cut the wood into required shape

Wooden stands has been prepared to hold the whole body strongly

Attached with the base plate in order to provide swivel motion.

Part Name : Body of the manipulator (Arm Link 1)

Material : Aluminum

Machining Process : Jig-saw to cut the sheet into the required shape

Drill holes and recess of various shapes to fit the motors into the joint

Part Name : Body of the manipulator (Arm Link 2)

Material : Aluminum

Machining Process : Jig-saw to cut the sheet into the required shape

Drill holes and recess of various shapes to fit the motors into the joints

Part Name : End-Effector Casing of the Manipulator

Material : Aluminum

Machining Process : Jig-saw to cut the sheet into the required shape

Drill holes and recess of various shapes to fit the motors into the joints

Part Name : End-Effector Casing of the Manipulator

Material : Aluminum

Machining Process : Jig-saw to cut the sheet into the required shape

Drill holes and recess of various shapes to fit the motors into the joints

### **6.3 Electronics Interface**

This section provides the joint servo motor guidance procedures for driving the end-effector in Cartesian space. In order to achieve a proper interfacing with the mechanical system through activation of joint servos, there are several electronic components employed in the present work. These are

- I. DC power supply to drive both servo motors and the electronic circuit.
- II. A microcontroller based electronic circuit receiving input signals and drive the motors using PWM mode.
- III. Joint motors of different torque rating.
- IV. An inverse kinematics based program to compute the required joint angles to drive the end-effector from home positions to the tile edges in case of replacement requirement.
- V. Remote observation screen.

### 6.3.1 Microcontroller Platform

There are wide varieties of microcontrollers currently available in the market. Recently the most widely used microcontroller is ATmega328. It has the following features:

1. Inbuilt 8-bit ADC
2. Inbuilt I2C and SPI protocol
3. Easier to program and implement
4. Community support
5. Hardware and software are open source

A low cost open source micro-controller board is used. The Arduino IDE is a multi-application development platform for creating embedded programs that run on the various Arduino microcontrollers. The Arduino Uno used in the present work has a processing speed of 16 MHz with a 8-bit architect and 256 RAM programmable with Arduino language.

The features of the Arduino Uno used are summarized in Table 6.1.

Table 6.2 Salient features of Arduino Uno

Microcontroller	ATmega328P
Operating voltage	5V
Input voltage range	7 to 12 V
Input voltage limit	6 to 20 V
Number of digital IO pins	14
Analog input pins	6
DC current per I/O pin	40 mA
Flash memory	32 kB
SRAM	2 kB
EEPROM	1 kB
Clock Speed	16 MHz

Figure 6.7 shows the physical configurations of Arduno Uno board.

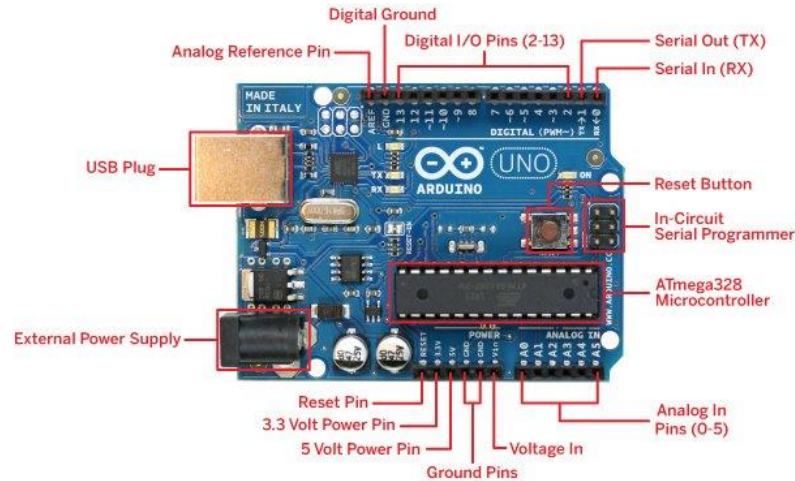


Figure 6.7 Arduino UNO schematic

Open-source hardware implies electronic product whose architect is free and open to everyone. Among the recent evolution of these activities, Arduino, a family of micro computing platform, is considered a pronominal open-source hardware. Arduino is based on easy-to-use hardware including Atmel AVR microcontroller and simple. Regarding the program development, a cross-platform called Arduino IDE is used with a code editor, compiler, and serial communication interface with Arduino board. The programming language is similar to C/C++. As API (Application Programming Interface) includes functions to communicate with Arduino boards through serial port, it is suitable for building graphical interface for hardware control.

### 6.3.2 Servo Motors

Servo motors employed in the present work are of low torque rating type and operated by 5 volt DC power supply. Generally they have

1. Gear reduction unit
2. Position sensing device (potentiometer)
3. Control circuit

Servo motors provide precise positioning of a joint through a specific angle. However their rotation is restricted between fixed angles of 0 to 180 degree. The servo motor has three pins: V<sub>CC</sub>, GND and PWM. The supply voltage is of 5V and the PWM pin is connected to the microcontroller output pin. The rotation direction of the servo motor is controlled with the help of PWM pulse generated by the microcontroller. Duty cycle is a metric measured in percentage as a ratio of the interval of time when the signal was high

to the period of time through which these signals repeat. The interval when the signal exists is exactly the pulse duration. So the required angle is obtained by servo from pulse-width modulation. Most commonly servo waits for signal every 20 milliseconds that corresponds to the frequency of 50 Hz .Pulse duration of 1ms refers to 0°position, 1.5 ms is 90° i.e. neutral position and 2ms to 180 degree of rotation respectively. Figure 6.8 shows various components of commercial servos.

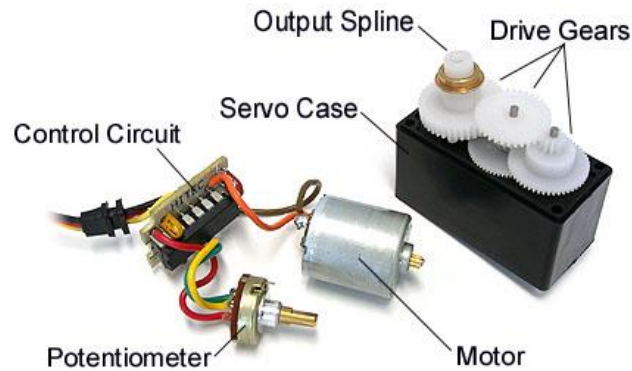


Figure 6.8 Different components of servo motor

Figure 6.9 shows the pin connection of Arduino with the six joint servos including gripper servo employed in the present work. Here two finger gripper is also driven by a motor.

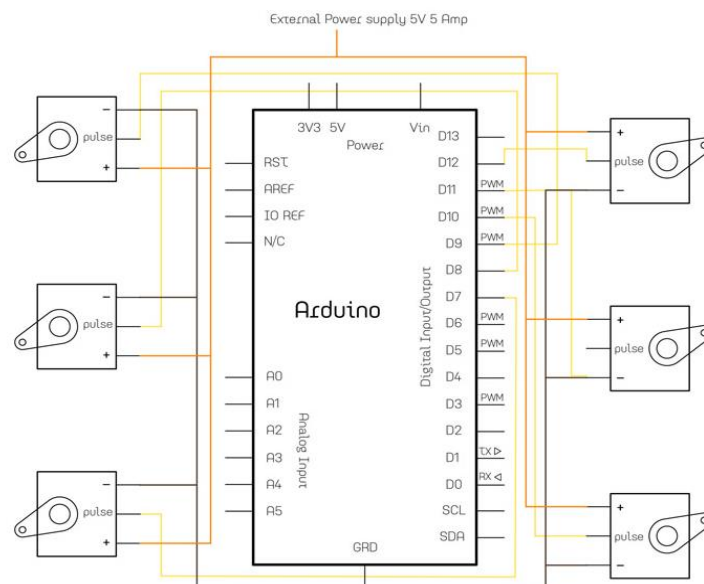


Figure 6.9 Arduino and servo motor connection configuration



In the present work two metallic geared servos, three plastic geared servos and one micro servo are employed in the assembly of the manipulator. The configurations of these motors are given in Table 6.2

Table 6.3 Configurations of the servos

Parameters	Metal Gear	Plastic Gear	Micro Servo
Stall Torque(kg, cm)	16	7	2.5
Weight(grams)	48	41	17
Speed (sec/deg)	0.14 - 0.18	0.16 – 0.20	0.12 - 0.14

## 6.4 Driving Tests

Inspection and replacement forms the major task in the proposed project. The previous chapter covered a thorough analysis of the inspection task i.e. identification of the faulty tile among the group of tiles and getting the coordinates of the faulty areas. As all the tiles are of similar dimension, the next step is to direct the end-effector of the robotic arm to move towards the defective tile. Initially the robot is made to stay at the home position, with the information from the image processing the object coordinate is calculated and through inverse kinematics the required angles are given to the joint servos. The end-effector is then made to proceed to the target object for gripping it. Slowly the robot holds it and releases the faulty tile in the wastebasket and takes a new one and moves towards the same location. The Figure 6.10 below shows the line diagram of whole physical set up implemented.

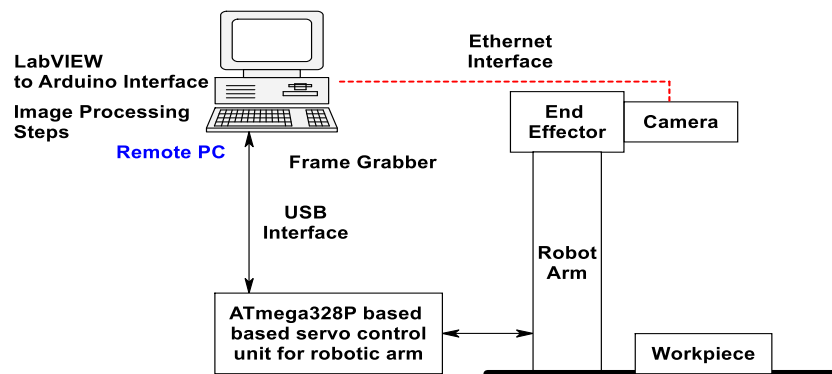


Figure 6.10 Schematic diagram of the whole setup of the present work

The robotic arm has to pick a defective object within the arm's workspace provided it is within camera field of vision. Figure 6.11 shows the step by step sequence of operations in reaching faulty tile and replacing it with new tile. In the present case there are 12 tiles with slots for keyways on the tile surface for tight fixation on to the wall. In real practice, the tiles are fixed to the wall surface by screws or other means. Also the surface may be a curved surface representing a toroidal shape of a vacuum vessel. As shown in Figure 6.11 the information in the faulty tile co-ordinates.

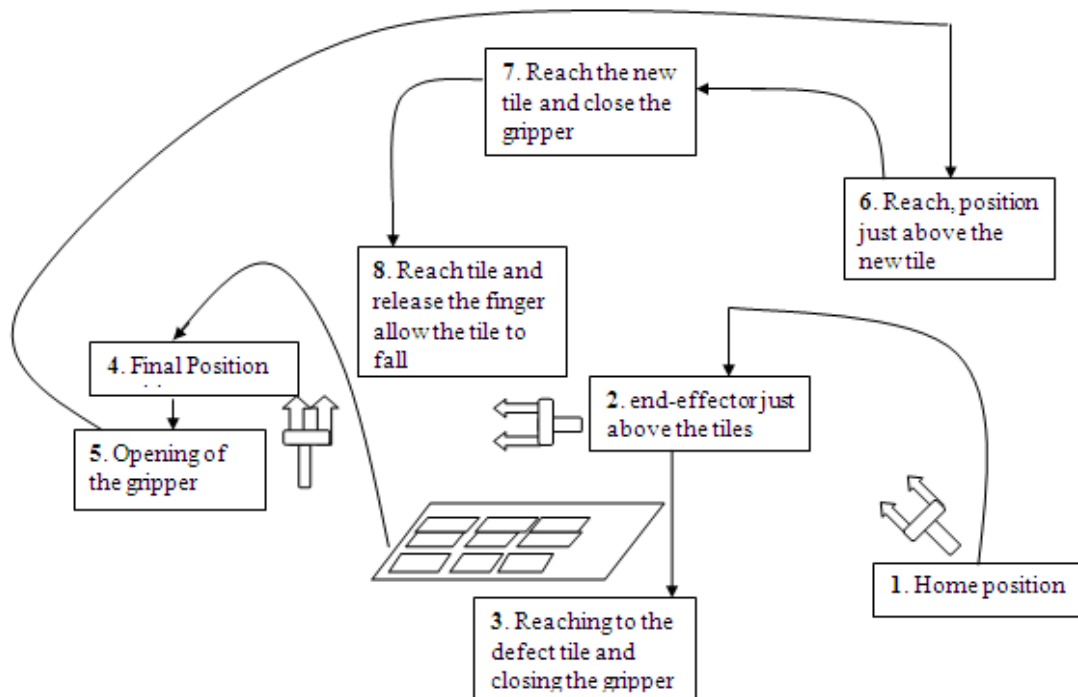


Figure 6.11 Process sequence of tile replacement (Home position → Tile surface → Waste tile bin → new tile bin → Tile surface)

As shown in above Figure, the information of the faulty tile co-ordinate received from image processing module are used to perform remaining steps. Initially, end-effector of the manipulator is moved from home position to surface just above the faulty tile and further it is picked up and moved to final position. As seen from step 6 to 8, a new tile is picked up from a fixed co-ordinates position and moved back to the previous tiles surface for completing the replacement. For these operation, initially robot end-effector is manually moved in steps and further for achieving these co-ordinates, the required joints

angles are computed by inverse kinematics. The fingers of the gripper (end-effector) reach the top and bottom edges of this tile and the gripper motor closes the fingers to apply the pressure for holding. Force Control issues are not considered here. All the steps of the proposed algorithms has been listed below:

- Home position
- End-effector just above the tile
- Reaching to the defect tile and closing the gripper.
- Reaching the final position
- Opening of the gripper
- Reaching to the position just above the new tile
- Releasing the finger and allowing the tile to fall

The objective of tile replacement using programmed robot is to remotely guide it and replace. The various steps in the movement of the end-effector depicted in Figure 6.11 are programmed using Arduino IDE. Figure 6.12 shows the screenshot of Arduino based programming environment used in the present work.



Figure 6.12 Screenshot of Arduino program

There are six motors are to be controlled simultaneously through a power supply received from USB port of the desktop PC. Figure 6.13 shows the photograph of Arduino based electronic circuit via a breadboard. A breadboard is used to supply DC power to each individual motor from a common power source. Finally, after testing, a PCB can be used instead of bread board.

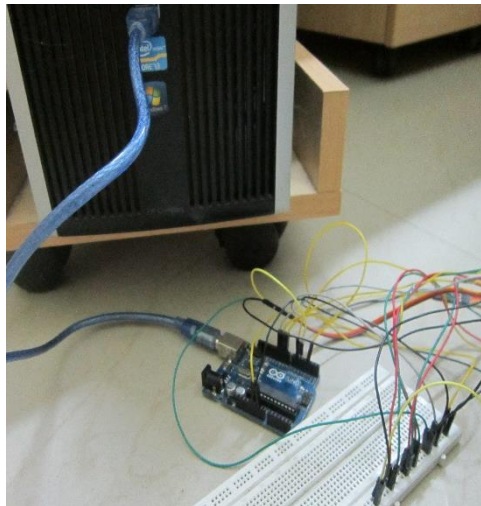


Figure 6.13 Arduino circuit connection

A code for driving the servos is given below:

```
=====
#include <Servo.h>
Servo myservo1; // create servo object to control a servo
                // twelve servo objects can be created on most boards
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
int pos = 0;    // variable to store the servo position
int startpoint[]={ 160,90,0,180,0};
int point1[]={ 0,150,0,150,180};
int point2[]={ 90,120,60,90,0};
//int endpoint[]={ 120,0,0,0,0};
void setup()
{
```

```

myservo1.attach(9); // attaches the servo on pin 9 to the servo object
myservo2.attach(10);
myservo3.attach(11);
myservo4.attach(12);
myservo5.attach(13);
}
void loop()
{
  // in steps of 1 degree
  myservo1.write(startpoint[0]); // tell servo to go to position in variable 'pos'
  delay(2000);
  myservo2.write(startpoint[1]);
  delay(2000);
  myservo3.write(startpoint[2]);
  delay(2000);
  myservo4.write(startpoint[3]);
  delay(2000);
  myservo5.write(startpoint[4]);
  delay(2000); // waits 1000ms for the servo to reach the position
  myservo1.write(point1[0]); // tell servo to go to position in variable 'pos'
  delay(2000);
  myservo2.write(point1[1]);
  delay(2000);
  myservo3.write(point1[2]);
  delay(2000);
  myservo4.write(point1[3]);
  delay(2000);
  myservo5.write(point1[4]);
  delay(2000);
  delay(1000); // waits 1000ms for the servo to reach the position
  myservo1.write(point2[0]); // tell servo to go to position in variable 'pos'
  myservo2.write(point2[1]);
  myservo3.write(point2[2]);
  myservo4.write(point2[3]);
  myservo5.write(point2[4]);
  // delay(1000); // waits 1000ms for the servo to reach the position
  // myservo1.write(endpoint[0]); // tell servo to go to position in variable 'pos'
  // myservo2.write(endpoint[1]);
  // myservo3.write(endpoint[2]);
  // myservo4.write(endpoint[3]);
  // myservo5.write(endpoint[4]);
  delay(1000); // waits 1000ms for the servo to reach the position
  // while();
}

```

---

The above program is executed for a test drive to replace the tile. However in practice, there are several issues such as sudden dropping of the tile, sudden movements from one position to other (jerky motion) and also teaching the program for replacement of general faulty tile have still not considered in this work.

## 6.5 Conclusion

This chapter provides the design details of fabricated model along with its testing using control platform. Step by step procedure to carry to the required task has also been explained. Details of electronic hardware implemented such as microcontroller, servo

motors, power supply, Arduino board has been provided. Real time testing has been carried out and the results were been demonstrated.

In the summary, vision system once identifies the image, after image processing if any fault is identified, the control system (in the present case through laptop with arduino code) drives the end-effector to the defected tile and lifts of it in sequential format. In present work however, a pertained sequence of points from holding, moving and replacing is only attempted, by random trails. In other words, vision system is ultimately used for predicting the faulty object and this information needs to be linked to robot joint the controller.

## Chapter 7

# Conclusions

### 7.1 Summary

In the present work an attempt has been made to comprehend an articulated arm for visual inspection and handling tasks. The proposed robot in the present work can be used for various activities such as quality inspection task, replacement activities in hazardous environments. Hence to meet the growing demands, extensive research is being carried out in the field of robotics and automation. It requires a thorough understanding of fundamentals of mathematical modeling such as kinematics and dynamics, control, vision, electronics etc. In this regard, the mathematical models for kinematics and dynamics of the 5 DOF articulated serial manipulator along with various control strategies for trajectory tracking, and introducing machine vision have been carried out and presented. The forward kinematics, inverse kinematics, workspace analysis, forward dynamics and inverse dynamics of the proposed arm were presented. Also, a CAD model was prepared using commercial software SolidWorks<sup>TM</sup> and is imported to ADAMS<sup>TM</sup> for motion and static force analysis.

Later in the control chapter, two different types of controllers have been used in two different cases under disturbance, no disturbance, and with disturbance observer conditions. The 5-DOF manipulator with rigid wrist configuration was considered links has been considered for control to reduce the complexity of the problem. Two set of trajectories have been chosen and under various conditions simulation have been performed using Matlab. For controlling in the joint space, computed torque control and Sliding Mode Control algorithms are implemented with a disturbance observer which reduces the effect of common disturbances such as joint friction and payload. These disturbances were represented with an idealized expression in term of joint angle variables Competent work have also been done in the area of machine vision and image processing which forms the base of any visual inspection system. For this a industrial grade Basler

Digital camera have been used for acquiring the image along with Matlab and LabVIEW interface. Essential aspect of image based position control have been described. The data set of real time tile images have been taken with and without defects. Image processing for crack detection have been done in Matlab using edge detection technique. Both qualitative and quantitative analysis such as location of crack coordinate, crack length, computational time ,comparative study of various edge detection technique, histogram analysis have been carried out . Real time crack analyses have also been carried using LabVIEW vision assistant toolbox.

Finally the scaled prototype has been fabricated and joint errors were driven with an Arduino based computed program. The effectiveness of the code in driving the end-effector (2 finger gripper) towards the affected tile and in lifting it up and replacement has been presented.

The following are the brief conclusions out of the present work:

- The Kinematic analysis of the 5 axis serial arm manipulator with DH parameter have been formulated and computed in Matlab. For Inverse kinematics solutions numerical technique i.e. Newton Raphson methods was illustrated. A solution to joint angle values computed with Matlab is found to be accurate up to 6 decimal number.
- Workspace analysis and Jacobian analysis of Manipulator was also be carried out
- Formulation of dynamic model was prepared using Euler –Lagrange method and equation of motion is derived for first three link
- Inverse dynamics showing the required torque for a given trajectory and forward dynamics showing the joint displacement and velocity have been simulated using Matlab.
- Two control scheme namely CTC, PD-SMC have been studied, for trajectory tracking of serial manipulator. Analyses with those control schemes have been done with and without disturbance observer and results have been found to be interesting.
- In the vision module, image processing for crack detection and location have been performed using both Matlab and LabVIEW .Popular edge detection techniques have been employed to identify the faulty tiles correctly. Various methods under edge detection are also been compared in Matlab. For real time implementations LabVIEW have been used to get the coordinate of the crack area.



- A scaled prototype of the proposed manipulator has been fabricated based on the final design of the arm and basic experiment has been conducted. Actuation has been provided to each and every joint using servos programmed in Arduino and pick and place operation has been tested.

## 7.2 Future scope

As a future scope of the work, several additional studies may be conducted. Few of this are listed below:

- Increasing the degree of freedom of the arm can facilitate it for more number of applications in inspection task as in real time vessel inspection in toroidal chambers, underground mines surveillances etc.
- Reconfigurable mechanism of the present manipulator with precise and sensor enabled gripper and joints can make the system more robust.
- Development of the real time implementation of dynamic model with control strategies is required .Control theories can be more improved by addition of intelligent gain-tuning using artificial network.
- Improvisation of crack detection algorithm and image based position control in real time implementation can give more uniqueness to the present work by making the system more accurate and robust one.
- An improved experimental work has to be done with coordinates of the object captured by the camera need to be mapped to the robot's domain prior to the implementation of the inverse kinematics model.
- To make the system fully automatic various RF modules such as Xbee, joint sensor at each and every joint, tactile sensors at the gripper can be used and code for running the manipulator can be further improvised.

# Bibliography

- [1] Robert Bogue, "Robots in the nuclear industry: a review of technologies and applications," *Ind. Robot Int. J.*, vol. 38, no. 2, pp. 113–118, Mar. 2011.
- [2] E. J. Van Henten, E. J. Schenk, L. G. van Willigenburg, J. Meuleman, and P. Barreiro, "Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot," *Biosyst. Eng.*, vol. 106, no. 2, pp. 112–124, Jun. 2010.
- [3] H. Wang, W. Chen, Y. Lai, and T. He, "Trajectory planning of tokamak flexible in-vessel inspection robot," *Fusion Eng. Des.*, vol. 98–99, pp. 1678–1682, Oct. 2015.
- [4] Y. Perrot, J. J. Cordier, J. P. Friconneau, L. Gargiulo, E. Martin, J. D. Palmer, and A. Tesini, "ITER articulated inspection arm (AIA): R&d progress on vacuum and temperature technology for remote handling," *Fusion Eng. Des.*, vol. 75–79, pp. 537–541, Nov. 2005.
- [5] L. Gargiulo, J. J. Cordier, J. P. Friconneau, C. Grisolia, J. D. Palmer, Y. Perrot, and F. Samaille, "Towards operations on Tore Supra of an ITER relevant inspection robot and associated processes," *Fusion Eng. Des.*, vol. 82, no. 15–24, pp. 1996–2000, Oct. 2007.
- [6] A. Mutka, I. Draganjac, Z. Kovacic, Z. Postruzin, and R. Munk, "Control system for reactor vessel inspection manipulator," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, 2009, pp. 1312–1318.
- [7] M. Houry, L. Gargiulo, C. Balorin, V. Bruno, D. Keller, H. Roche, N. Kammerer, Y. Measson, F. Carrel, and V. Schoepff, "Diagnostics carried by a light multipurpose deployer for vacuum vessel interventions," *Fusion Eng. Des.*, vol. 86, no. 9–11, pp. 1868–1871, Oct. 2011.
- [8] M. Houry, P. Bayetti, D. Keller, L. Gargiulo, V. Bruno, J. C. Hatchressian, C. Hernandez, J. P. Martins, Y. Measson, Y. Perrot, and F. X. Russotto, "Development of in situ diagnostics and tools handled by a light multipurpose carrier for tokamak in-vessel interventions," *Fusion Eng. Des.*, vol. 85, no. 10–12, pp. 1947–1951, Dec. 2010.
- [9] I. Ribeiro, C. Damiani, A. Tesini, S. Kakudate, M. Siuko, and C. Neri, "The remote handling systems for ITER," *Fusion Eng. Des.*, vol. 86, no. 6–8, pp. 471–477, Oct. 2011.
- [10] X. B. Peng, Y. T. Song, C. C. Li, M. Z. Lei, and G. Li, "Conceptual design of EAST flexible in-vessel inspection system," *Fusion Eng. Des.*, vol. 85, no. 7–9, pp. 1362–1365, Dec. 2010.
- [11] A. Kitamura, K. Nakai, T. Namekawa, and M. Watahiki, "In-cell maintenance by manipulator arm with 3D workspace information recreated by laser rangefinder," *Nucl. Eng. Des.*, vol. 241, no. 7, pp. 2614–2623, Jul. 2011.
- [12] X. Peng, J. Yuan, W. Zhang, Y. Yang, and Y. Song, "Kinematic and dynamic analysis of a serial-link robot for inspection process in EAST vacuum vessel," *Fusion Eng. Des.*, vol. 87, no. 5–6, pp. 905–909, Aug. 2012.
- [13] D. F. Valcárcel, D. Alves, P. Card, B. B. Carvalho, S. Devaux, R. Felton, A. Goodyear, P. J. Lomas, F. Maviglia, P. McCullen, C. Reux, F. Rimini, A. Stephen, L. Zabeo, and K.-D. Zastrow, "The JET real-time plasma-wall load monitoring system," *Fusion Eng. Des.*, vol. 89, no. 3, pp. 243–258, Mar. 2014.
- [14] P. S. Pagala, M. Ferre, and L. Orona, "Evaluation of modular robot system for maintenance tasks in hot cell," *Fusion Eng. Des.*, vol. 89, no. 9–10, pp. 2309–2313, Oct. 2014.

- [15] L. Snoj, I. Lengar, A. Čufar, B. Syme, S. Popovichev, S. Conroy, and L. Meredith, "Calculations to support JET neutron yield calibration: Modelling of the JET remote handling system," *Nucl. Eng. Des.*, vol. 261, pp. 244–250, Aug. 2013.
- [16] H. Wang, Y. Lai, W. Chen, and Q. Cao, "Design and implementation of motion planning of inspection and maintenance robot for ITER-like vessel," *Fusion Eng. Des.*, vol. 101, pp. 111–123, Dec. 2015.
- [17] C.-H. Chen and H.-P. Huang, "Pose estimation for autonomous grasping with a robotic arm system," *J. Chin. Inst. Eng.*, vol. 36, no. 5, pp. 638–646, Jul. 2013.
- [18] E. Villedieu, V. Bruno, P. Pastor, L. Gargiulo, Y. T. Song, Y. Cheng, H. Feng, C. Liu, and S. S. Shi, "An Articulated Inspection Arm for fusion purposes," *Fusion Eng. Des.*
- [19] S. Manzoor, R. Ul Islam, A. Khalid, A. Samad, and J. Iqbal, "An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 3, pp. 351–362, Jun. 2014.
- [20] Y. Perrot, L. Gargiulo, M. Houry, N. Kammerer, D. Keller, Y. Measson, G. Piolain, and A. Verney, "Long-reach articulated robots for inspection and mini-invasive interventions in hazardous environments: Recent robotics research, qualification testing, and tool developments," *J. Field Robot.*, vol. 29, no. 1, pp. 175–185, Jan. 2012.
- [21] K. Kaltsoukalas, S. Makris, and G. Chryssolouris, "On generating the motion of industrial robot manipulators," *Robot. Comput.-Integr. Manuf.*, vol. 32, pp. 65–71, Apr. 2015.
- [22] J. Santolaria, J.-J. Aguilar, J.-A. Yagüe, and J. Pastor, "Kinematic parameter estimation technique for calibration and repeatability improvement of articulated arm coordinate measuring machines," *Precis. Eng.*, vol. 32, no. 4, pp. 251–268, Oct. 2008.
- [23] P.-C. Tung, M.-C. Wu, and Y.-R. Hwang, "An image-guided mobile robotic welding system for SMAW repair processes," *Int. J. Mach. Tools Manuf.*, vol. 44, no. 11, pp. 1223–1233, Sep. 2004.
- [24] J. Soares, A. Vale, and R. Ventura, "A Multi-purpose Rescue Vehicle and a human–robot interface architecture for remote assistance in ITER," *Fusion Eng. Des.*, vol. 98–99, pp. 1656–1659, Oct. 2015.
- [25] P. Chen and Q. Cao, "Development of a Remote Handling Robot for the Maintenance of an ITER-Like D-Shaped Vessel," *Sci. Technol. Nucl. Install.*, vol. 2014, pp. 1–9, 2014.
- [26] E. Robbins, S. Sanders, A. Williams, and P. Allan, "The use of virtual reality and intelligent database systems for procedure planning, visualisation, and real-time component tracking in remote handling operations," *Fusion Eng. Des.*, vol. 84, no. 7–11, pp. 1628–1632, Jun. 2009.
- [27] A. Vale, D. Fonte, F. Valente, and I. Ribeiro, "Trajectory optimization for autonomous mobile robots in ITER," *Robot. Auton. Syst.*, vol. 62, no. 6, pp. 871–888, Jun. 2014.
- [28] T. Maruyama, A. Aburadani, N. Takeda, S. Kakudate, M. Nakahira, and A. Tesini, "Robot vision system R&D for ITER blanket remote-handling system," *Fusion Eng. Des.*, vol. 89, no. 9–10, pp. 2404–2408, Oct. 2014.
- [29] K. Dhanapal, A. P. Singh, V. Rakesh, C. Rajagopalan, B. P. C. Rao, S. Venugopal, and T. Jayakumar, "Remote Devices for Inspection of Process Vessel and Conduits," *Procedia Eng.*, vol. 64, pp. 1329–1336, 2013.

- [30] C.-H. Huang, C.-S. Hsu, P.-C. Tsai, R.-J. Wang, and W.-J. Wang, "Vision based 3-D position control for a robot arm," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 1699–1703.
- [31] C. Hua, Y. Wang, and X. Guan, "Visual tracking control for an uncalibrated robot system with unknown camera parameters," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 1, pp. 19–24, Feb. 2014.
- [32] J.-K. Oh, G. Jang, S. Oh, J. H. Lee, B.-J. Yi, Y. S. Moon, J. S. Lee, and Y. Choi, "Bridge inspection robot system with machine vision," *Autom. Constr.*, vol. 18, no. 7, pp. 929–941, Nov. 2009.
- [33] K.-L. Lin and J.-L. Fang, "Applications of computer vision on tile alignment inspection," *Autom. Constr.*, vol. 35, pp. 562–567, Nov. 2013.
- [34] H.-G. Moon, J.-H. Kim, H.-G. Moon, and J.-H. Kim, "Intelligent Crack Detecting Algorithm On The Concrete Crack Image Using Neural Network," *ISARC Proc.*, vol. 2011 Proceedings of the 28th ISARC, Seoul, Korea, pp. 1461–1467, 2011.
- [35] J. G. Victores, S. Martínez, A. Jardón, and C. Balaguer, "Robot-aided tunnel inspection and maintenance system by vision and proximity sensor integration," *Autom. Constr.*, vol. 20, no. 5, pp. 629–636, Aug. 2011.
- [36] M. H. Korayem and F. S. Heidari, "Simulation and experiments for a vision-based control of a 6R robot," *Int. J. Adv. Manuf. Technol.*, vol. 41, no. 3–4, pp. 367–385, Apr. 2008.
- [37] R. V. Sharan and G. C. Onwubolu, "Client-server control architecture for a vision-based pick-and-place robot," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 226, no. 8, pp. 1369–1378, Aug. 2012.
- [38] P. Bellandi, F. Docchio, and G. Sansoni, "Roboscan: a combined 2D and 3D vision system for improved speed and flexibility in pick-and-place operation," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 5–8, pp. 1873–1886, Jul. 2013.
- [39] L. Nele, E. Sarno, and A. Keshari, "An image acquisition system for real-time seam tracking," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 9–12, pp. 2099–2110, Jul. 2013.
- [40] B. P. Larouche and Z. H. Zhu, "Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control," *Auton. Robots*, vol. 37, no. 2, pp. 157–167, Jan. 2014.
- [41] A. Mohammed, L. Wang, and R. X. Gao, "Integrated Image Processing and Path Planning for Robotic Sketching," *Procedia CIRP*, vol. 12, pp. 199–204, 2013.
- [42] C. Laofor and V. Peansupap, "Defect detection and quantification system to support subjective visual quality inspection via a digital image processing: A tiling work case study," *Autom. Constr.*, vol. 24, pp. 160–174, Jul. 2012.
- [43] S.-N. Yu, J.-H. Jang, and C.-S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," *Autom. Constr.*, vol. 16, no. 3, pp. 255–261, May 2007.
- [44] G. K. Choudhary and S. Dey, "Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks," in *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, 2012, pp. 404–411.
- [45] S. Ekvall, D. Kragic, and F. Hoffmann, "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling," *Image Vis. Comput.*, vol. 23, no. 11, pp. 943–955, Oct. 2005.
- [46] R. Sibois, K. Salminen, M. Siuko, J. Mattila, and T. Määtä, "Enhancement of the use of digital mock-ups in the verification and validation process for ITER remote handling systems," *Fusion Eng. Des.*, vol. 88, no. 9–10, pp. 2190–2193, Oct. 2013.

- [47] B. He, L. Han, Y. Wang, S. Huang, and L. Liu, "Kinematics analysis and numerical simulation of a manipulator based on virtual prototyping," *Int. J. Adv. Manuf. Technol.*, vol. 71, no. 5–8, pp. 943–963, Dec. 2013.
- [48] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, "Nonlinear disturbance observer design for robotic manipulators," *Control Eng. Pract.*, vol. 21, no. 3, pp. 253–267, Mar. 2013.
- [49] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 932–938, Aug. 2000.
- [50] A. Nikoobin and R. Haghighi, "Lyapunov-Based Nonlinear Disturbance Observer for Serial n-Link Robot Manipulators," *J. Intell. Robot. Syst.*, vol. 55, no. 2–3, pp. 135–153, Dec. 2008.
- [51] C.-S. Liu and H. Peng, "Disturbance Observer Based Tracking Control," *J. Dyn. Syst. Meas. Control*, vol. 122, no. 2, pp. 332–335, May 1997.
- [52] D. Ginoya, P. D. Shendge, and S. B. Phadke, "Disturbance observer based sliding mode control of nonlinear mismatched uncertain systems," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 26, no. 1–3, pp. 98–107, Sep. 2015.
- [53] Z. Song, J. Yi, D. Zhao, and X. Li, "A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach," *Fuzzy Sets Syst.*, vol. 154, no. 2, pp. 208–226, Sep. 2005.
- [54] F. Piltan, A. Hosainpour, S. Emamzadeh, I. Nazari, and M. Mirzaie, "Design Sliding Mode Controller of with Parallel Fuzzy Inference System Compensator to Control of Robot Manipulator," *IAES Int. J. Robot. Autom. IJRA*, vol. 2, no. 4, pp. 149–162, Dec. 2013.
- [55] S. Rahmdel, F. Piltan, S. Mehrara, and R. Bayat, "Sliding Mode Methodology Vs. Computed Torque Methodology Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control Courses," *Int. J. Eng. IJE*, vol. 6, no. 6, pp. 142–177, Jun. 2012.
- [56] K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 3, pp. 328–342, May 1999.
- [57] P. R. Ouyang, J. Acob, and V. Pano, "PD with sliding mode control for trajectory tracking of robotic system," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 2, pp. 189–200, Apr. 2014.
- [58] F. Piltan, N. Sulaiman, Z. Tajpaykar, and P. Ferdosali, "Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC With Tunable Gain," *Int. J. Robot. Autom.*, vol. 3, no. 2, pp. 195–210, 2011.
- [59] J. P. Kolhe, M. Shaheed, T. s. Chandar, and S. e. Talole, "Robust control of robot manipulators based on uncertainty and disturbance estimation," *Int. J. Robust Nonlinear Control*, vol. 23, no. 1, pp. 104–122, Jan. 2013.
- [60] M. Van and H.-J. Kang, "Robust fault-tolerant control for uncertain robot manipulators based on adaptive quasi-continuous high-order sliding mode and neural network," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, p. 954406214544311, Feb. 2015.
- [61] S. Pezeshki, S. Badalkhani, and A. Javadi, "Performance Analysis of a Neuro-PID Controller Applied to a Robot Manipulator," *Int. J. Adv. Robot. Syst.*, p. 1, 2012.

## Bibliography

- [62] Thomas R. Kurfess, *Robotics and Automation Handbook*, CRC Press

- [63] Robert J. Schilling, Fundamentals of Robotics Analysis and Control, Prentice-Hall of India.
- [64] Richard M. Murray, Zexiang Li, S. Shanker Sastry, A Mathematical Introduction to Robotic Manipulator, CRC Press, 2009
- [65] John J. Craig, Introduction to Robotics-Mechanics and Control, Pearson Pub.
- [66] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villoni, Giuseppe Oriole, Robotics Modelling, Planning and Control, Advanced Textbook in Control and Signal Processing, Springer
- [67] Saeed Nikku, Introduction to Robotics, John Wiley & Sons.
- [68] M. Vidyasagar, Mark W. Spong, Robot Dynamics and Control, John Wiley & Sons

# Appendix A

## Dynamic Model of the manipulator

Such dynamic equations are obtained fundamentally from expressions of energy terms. Following part gives the expressions for kinetic and potential energies.

### A1. Mass, coriolis and gravity from kinematic relationships

The kinetic energy of a rigid body in 3-D motion, about its center of mass is given as:

$$K = \frac{1}{2}mv_G^2 + \frac{1}{2}\omega h_G \quad (\text{A.1})$$

where  $h_G$  is the angular momentum of the body about G.

The kinetic energy of a rigid body in plane motion is given as:

$$K = \frac{1}{2}mv_G^2 + \frac{1}{2}\bar{I}\omega \quad (\text{A.2})$$

where  $\bar{I}$  is the inertia tensor given as follow:

$$\bar{I} = \begin{bmatrix} \int (y^2 + z^2) & -\int xydm & -\int xzdm \\ -\int xydm & \int (x^2 + z^2) & -\int yzdm \\ -\int xzdm & -\int yzdm & \int (x^2 + y^2) \end{bmatrix} \quad (\text{A.3})$$

It should be noted that all the diagonal terms are the moment of inertia and rest other terms represent the product of inertia. The kinetic energy of a particle with differential mass  $dm$  in link  $n$  is:

$$dK_n = \frac{1}{2}(\dot{x}_n^2 + \dot{y}_n^2 + \dot{z}_n^2)dm = \frac{1}{2}\text{trace}(\mathbf{V}_n \mathbf{V}_n^T)dm \quad (\text{A.4})$$

And  $\mathbf{V}_n$  can be defined as  $\left( \sum_{j=1}^n U_{nj} \dot{q}_j \right) \mathbf{r}_n$  therefore:

$$dK_n = \frac{1}{2} Tr \left[ \sum_{p=1}^n U_{np} \dot{q}_p r_n^n \left( \sum_{r=1}^n U_{nr} \dot{q}_r r_n^n \right)^T \right] dm \quad (A.5)$$

which is further simplified as:

$$dK_n = \frac{1}{2} Tr \left[ \sum_{p=1}^n \sum_{r=1}^n U_{np} \left( r_n^n r_n^{nT} dm \right) U_{nr}^T \dot{q}_p \dot{q}_r \right] \quad (A.6)$$

The total kinetic energy of the n-link is then calculated by integrating kinetic energy of the differential mass element across the link:

$$K_n = \int dK_n = \frac{1}{2} Tr \left[ \sum_{p=1}^n \sum_{r=1}^n U_{np} \left( \int r_n^n r_n^{nT} dm \right) U_{nr}^T \dot{q}_p \dot{q}_r \right] \quad (A.7)$$

The term  $\int r_n^n r_n^{nT} dm$  is defined as the pseudo-inertia matrix for link n and is represented by  $J_n$

$$J_n = \int r_n^n r_n^{nT} dm = \begin{bmatrix} \int x_n^2 dm & \int x_n y_n dm & \int x_n z_n dm & \int x_n dm \\ \int x_n y_n dm & \int y_n^2 dm & \int x_n z_n dm & \int y_n dm \\ \int x_n z_n dm & \int x_n z_n dm & \int z_n^2 dm & \int z_n dm \\ \int x_n dm & \int y_n dm & \int z_n dm & \int dm \end{bmatrix} \quad (A.8)$$

Finally, the total kinetic energy for the manipulator is the sum of the kinetic energies of the link:

$$\begin{aligned} K &= \sum_{n=1}^i K_n = \frac{1}{2} \sum_{n=1}^i Tr \left[ \sum_{p=1}^n \sum_{r=1}^n U_{np} \left( \int r_n^n r_n^{nT} dm \right) U_{nr}^T \dot{q}_p \dot{q}_r \right] \\ &= \frac{1}{2} \sum_{n=1}^i \sum_{p=1}^n \sum_{r=1}^n [Tr (U_{np} J_n U_{nr}^T) \dot{q}_p \dot{q}_r] \end{aligned} \quad (A.9)$$

## Potential Energy

The potential energy of link n is defined as follows:



$$P_n = -m_n g^T (T_0^n \bar{r}_i) \quad (\text{A.10})$$

Where  $\bar{r}_i$  is the location of the center of mass of the link with respect to the frame representing the link and

$g^T = [g_x \ g_y \ g_z \ 0]$  is the gravity row vector expressed in base frame

The potential energy of the robot arm is then:

$$P = \sum_{n=0}^n P_n = \sum_{n=0}^n [-m_n g^T (T_0^n \bar{r}_i)] \quad (\text{A.11})$$

which is a function of  $q_n$ .

In order to simplify the solution, the dynamic equations can be determined using Lagrangian mechanics. The Lagrange function allows the forces to be determined using the system's energy. This eliminates the need to solve for acceleration components. Since energy is a scalar quantity, it also makes the equation easier to manage. The Lagrangian function takes the form of:

$$L = K - P = \frac{1}{2} \sum_{n=1}^n \sum_{j=1}^n \sum_{k=1}^i [Tr(U_{nj} J_n U_{nk}^T) \dot{q}_j \dot{q}_k] + \sum_{n=1}^i m_n g^T (T_0^n \bar{r}_i) \quad (\text{A.12})$$

As

$$\sum_{j=nk=1}^i \sum_{j=1}^i Tr(U_{jk} J_j U_{jn}^T) \ddot{q}_k + \sum_{j=nk=1}^i \sum_{j=1}^i \sum_{l=1}^i Tr\left(\frac{\partial U_{jk}}{\partial q_m} J_j U_{jn}^T\right) \dot{q}_k \dot{q}_m - \sum_{j=n}^i m_j g U_{jn} \bar{r}_j^j = \tau_n \quad (\text{A.13})$$

$$\tau_n = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_n} \right) - \frac{\partial L}{\partial q_n} \quad (\text{A.14})$$

For n link case

$$U_{nj} \equiv \frac{\partial U_{nj}}{\partial q_k} = \begin{cases} {}^0T_{j-1}Q_j^{j-1}T_{k-1}Q_k^{k-1}T_n & n > k > j \\ {}^0T_{k-1}Q_j^{k-1}T_{j-1}Q_j^{j-1}T_n & n > j > k \\ 0 & n < j \\ & n < k \end{cases} \quad (A.15)$$

The dynamic model of an n-link arm is then presented as:

$$\tau_n = \sum_{k=1}^i M_{nk} \ddot{q}_k + \sum_{k=1}^i \sum_{m=1}^i C_{nkm} \dot{q}_k \dot{q}_m + G_n \quad (A.16)$$

Where:

$$M_{nk} = \sum_{j=\max(n,k)}^i \text{Tr}(U_{jk} J_j U_{jn}^T) \quad (A.17)$$

$$C_{nkm} = \sum_{j=\max(n,k,m)}^i \text{Tr}(U_{jkm} J_j U_{jn}^T) \quad (A.18)$$

$$G_n = \sum_{j=n}^i m_j g U_{jn}^j \bar{r}_j \quad (A.19)$$

In simplified form we can write as :

$$\begin{aligned} M(\theta) (\ddot{\theta}) + C(\theta, \dot{\theta}) + \\ G(\theta) = \tau \end{aligned} \quad (A.20)$$

## A.2 Motion equation of a robot arm with 3 rotary Joints

In present work the wrist is assumed to be locked with its two joints and first three degrees of freedom are only considered. The following set of matrix and vector terms are obtained.

Inertia Matrix,  $M(\theta)$  of the arm:

$$M(\theta) = \begin{bmatrix} M_{11} & M_{21} & M_{31} \\ M_{12} & M_{22} & M_{32} \\ M_{13} & M_{23} & M_{33} \end{bmatrix} \quad (A.21)$$

$$M_{11} = \text{Tr}(U_{11} J_1 U_{11}^T) + \text{Tr}(U_{21} J_2 U_{21}^T) + \text{Tr}(U_{31} J_3 U_{31}^T) \quad (A.22)$$

$$\mathbf{M}_{12} = \mathbf{M}_{21} = \text{Tr}(\mathbf{U}_{22} \mathbf{J}_2 \mathbf{U}_{21}^T) + \text{Tr}(\mathbf{U}_{32} \mathbf{J}_3 \mathbf{U}_{31}^T) \quad (\text{A.23})$$

$$\mathbf{M}_{13} = \mathbf{M}_{31} = \text{Tr}(\mathbf{U}_{33} \mathbf{J}_3 \mathbf{U}_{31}^T) \quad (\text{A.24})$$

$$\mathbf{M}_{22} = \text{Tr}(\mathbf{U}_{22} \mathbf{J}_2 \mathbf{U}_{22}^T) + \text{Tr}(\mathbf{U}_{32} \mathbf{J}_3 \mathbf{U}_{32}^T) \quad (\text{A.25})$$

$$\mathbf{M}_{23} = \mathbf{M}_{32} = \text{Tr}(\mathbf{U}_{33} \mathbf{J}_3 \mathbf{U}_{32}^T) \quad (\text{A.26})$$

The gravity vector has been expressed as:

$$\mathbf{g}(\theta, \dot{\theta}) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = [g_1 \quad g_2 \quad g_3]^T \quad (\text{A.27})$$

Where,

$$g_1 = 0 \quad (\text{A.28})$$

$$g_2 = -\left(m_1 g U_{11}^1 \bar{r}_1 + m_2 g U_{21}^2 \bar{r}_2 + m_3 g U_{31}^3 \bar{r}_3\right) - \left(m_1 g U_{22}^2 \bar{r}_2 + m_3 g U_{32}^3 \bar{r}_3\right) \quad (\text{A.29})$$

$$g_3 = \left(m_3 g U_{33}^3 \bar{r}_3\right) \quad (\text{A.30})$$

# Appendix B

## Edge detection algorithms

Image segmentation using edge detection is a famous approach for identifying cracks on the surface. Following are some of the edge detection techniques.

### B.1. Sobel edge detection

The Sobel edge detector consists of a pair of  $3 \times 3$  convolution kernels. One kernel is simply the other rotated by  $90^\circ$ . The kernels are represented as:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (\text{B.1})$$

The magnitude can be represented as  $|G| = \sqrt{G_x^2 + G_y^2}$

The angle of orientation of edge can be represented as  $\theta = \tan^{-1} \left( \frac{G_y}{G_x} \right)$

### B.2 Roberts edge detection

The kernels of roberts edge detection can be represented as  $G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$   $G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

The magnitude can be represented as  $|G| = \sqrt{G_x^2 + G_y^2}$

The angle of orientation of edge can be represented as  $\theta = \tan^{-1} \left( \frac{G_y}{G_x} \right) - \frac{3\pi}{4}$

### B.3 Prewitt edge detection

Prewitt edge detection detects horizontal and vertical edges. The mask of prewitt edge detection can be represented as

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (\text{B.2})$$

#### B.4 Laplacian of Gaussian (LoG)

In LoG, a Gaussian kernel is used for edge detection. The Gaussian kernel can be represented as

$$g(x, y, t) = \frac{1}{2\pi t^2} \exp\left(-\frac{x^2 + y^2}{2t^2}\right) \quad (\text{B.3})$$

There are various object recognition techniques available by using its feats. Some feats used for object identification are gray levels, perimeter, area, diameter, number of holes, similar morphological characteristics, aspect ratio, thickness etc.

#### B.5 Image Analysis of an individual tile with a single crack

MATLAB has been used for image processing and image analysis .Further in this section we have taken two sets of individual cracked tile image for analysis and visualization for the most effective method under various edge detection method .Under this analysis the crack area has been identified along with the crack dimension as well as the simulation time required by the code has also been computed.

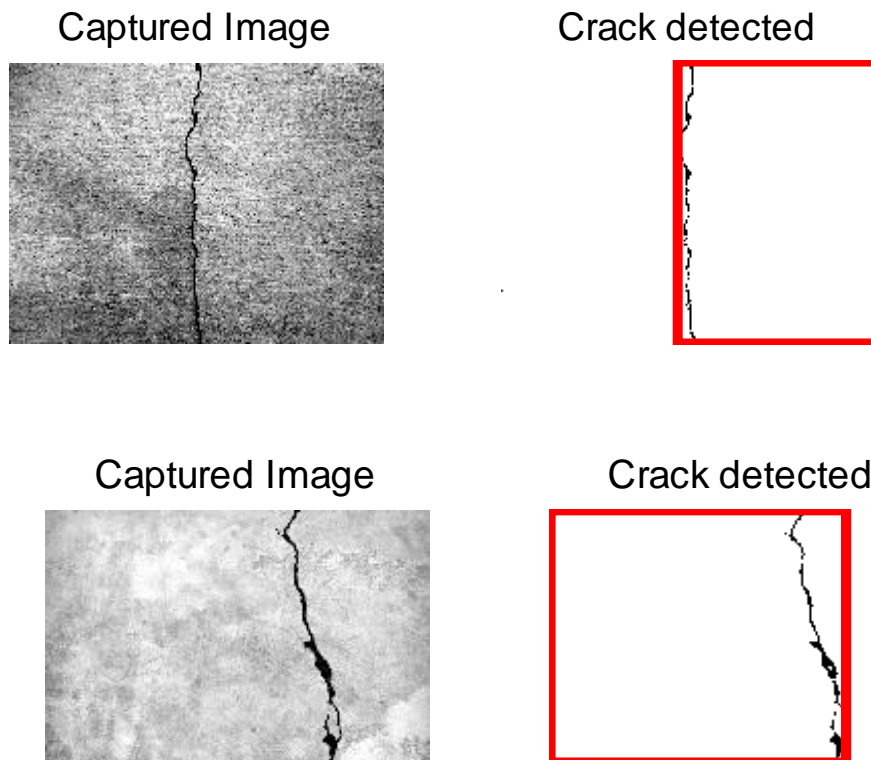


Figure B.1 Crack detection using MATLAB considering offline images

A computer program with Matlab image processing toolbox functions is employed to receive the image as input and gives the output as measurement of the length of the crack along with the computation time involved. Table B.2 shows the computational results of Figure B.1.

Table B.1: Computational results

Figure	Crack dimension (cm)	Time taken to compute (sec)
1	22.3	1.0267
2	9.1546	0.999238

With the set of two crack images, various edge detection techniques such as Canny, Prewitt, Sobel, Roberts, Laplacian of Gaussian (LoG) and Zero-cross are implemented. The results below show the results of various operators used.

#### Image Set 1

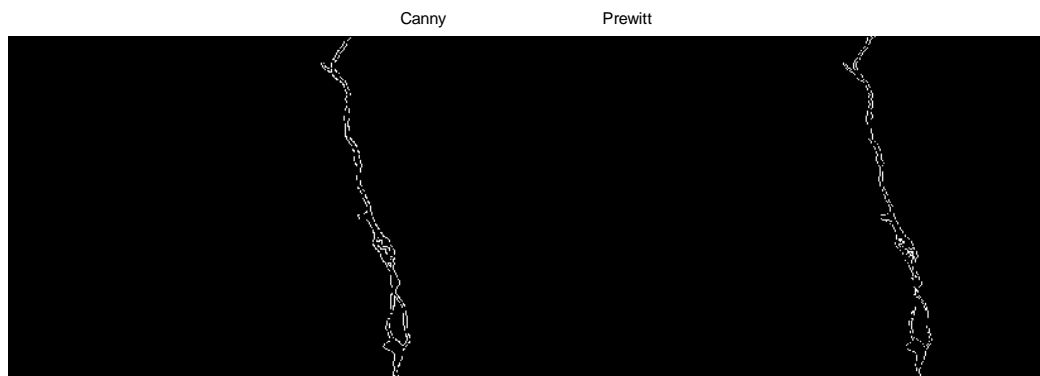


Figure B.2 Crack detection using Canny and Prewitt edge detection technique



Figure B.3 Crack detection using sobel and Roberts edge detection techniques

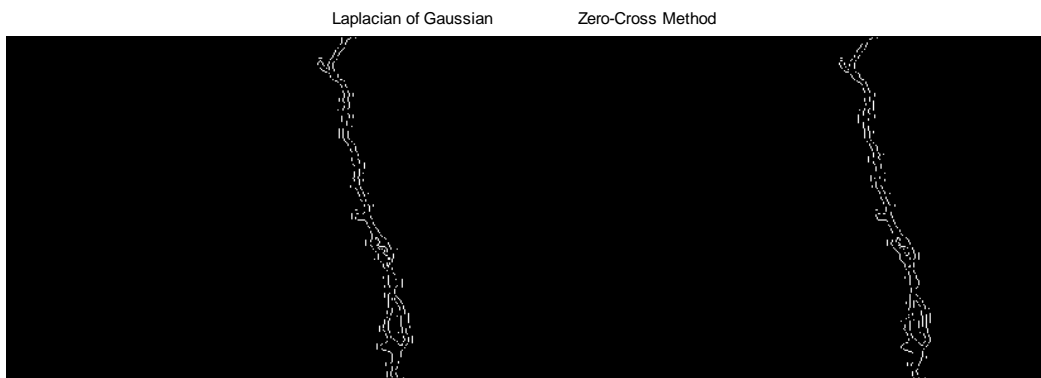


Figure B.4 Crack detection using Laplacian of Gaussian (LoG) and Zero-cross detection technique

### Edge detection of image 2



Figure B.5 Crack detection using Canny and Prewitt edge detection technique



Figure B.6 Crack detection using sobel and Roberts edge detection techniques

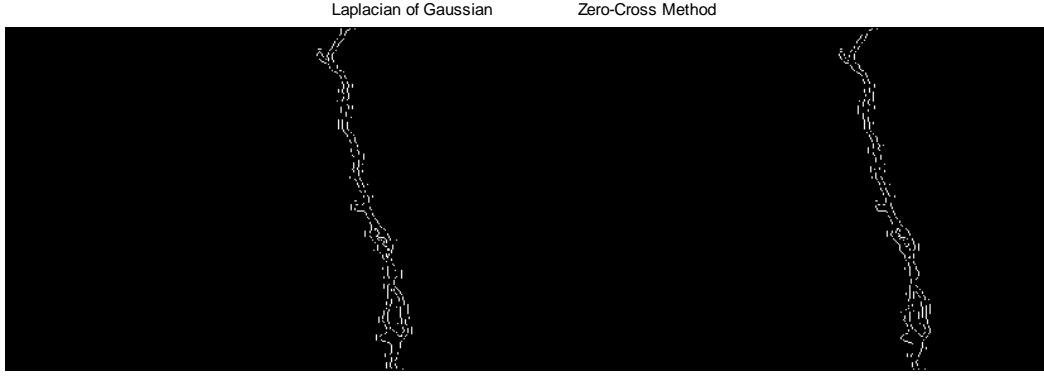


Figure B.6 Crack detection using Laplacian of Gaussian (LoG) and Zero-cross detection technique

### Performance evaluation of edge detection techniques

To calculate the performance of different edge detection techniques, two parameters i.e mean square error (MSE) and Peak signal to noise ratio (PSNR) are calculated in each cases.

#### Mean Square Error

MSE can be calculated as

$$MSE = \frac{\sum_{M,N} (I_1(m,n) - I_2(m,n))^2}{M,N} \quad (B.4)$$

Here,  $I_1(m,n)$  is the original image,  $I_2(m,n)$  is the edge detected image, m and n are rows and column of the image.

#### Root Mean Square Error

RMSE can be calculated as

$$RMSE = \sqrt{\frac{\sum_{M,N} (I_1(m,n) - I_2(m,n))^2}{M,N}} \quad (B.5)$$

Here,  $I_1(m,n)$  is the original image,  $I_2(m,n)$  is the edge detected image, m and n are rows and column of the image.

#### Peak Signal to Noise Ratio (PSNR)



Peak signal-to-noise ratio, is ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR usually expressed in terms of the decibel (dB) scale. PSNR can be represented as

$$PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right) \quad (B.6)$$

R is the maximal variation in the input image data. If it has an 8-bit unsigned integer data type, R is 255. The following Matlab code is employed to obtain the accuracy means of each of the various edge detection techniques. The input for the program is any image with crack while the output is shown in the form of comparative estimates of MSE, RMSE, PSNR. Tables B.2 and B.3 shows the output of the program for two images with cracks considered earlier.

Table B.2: Computational result for image1

	<b>Canny</b>	<b>Prewitt</b>	<b>Sobel</b>	<b>Roberts</b>	<b>Zerocross</b>	<b>LoG</b>
<b>MSE</b>	0.9851	0.9855	0.9856	0.9858	0.9811	0.9811
<b>RMSE</b>	0.9952	0.9927	0.9928	0.9929	0.9905	0.9905
<b>PSNR</b>	48.23	48.22	48.22	48.22	48.24	48.24
<b>Number of edge pixels</b>	1310	1382	1253	1542	1784	1784

Table B.3: Computational result for image2

	<b>Canny</b>	<b>Prewitt</b>	<b>Sobel</b>	<b>Roberts</b>	<b>Zerocross</b>	<b>LoG</b>
<b>MSE</b>	0.9882	0.9895	0.9893	0.9893	0.9846	0.9846
<b>RMSE</b>	0.9941	0.9948	0.9946	0.9946	0.9923	0.9923
<b>PSNR</b>	48.21	48.21	48.21	48.21	48.23	48.23
<b>Number of edge pixels</b>	1286	1098	1087	1395	1299	1299

It can be seen that the number of edge pixels as well as signal to noise ratio mean are considerably higher in the case of Zerocross and LoG methods and can be employed further.

# Appendix C

## Program Listing

### *C.1 Inverse Kinematics*

```
% defining time step
t=0:0.1:1;

% defining trajectory in Cartesian space
px=(6.*t-3).*cos(2.*(6.*t-3));py=sin(3.*(6.*t-3));pz=cos(3-6.*t);

% Plotting the trajectory
plot3(px,py,pz,'-b');
xlabel('p_x');ylabel('p_y');zlabel('p_z');

% link parameter value
d1=50; a2=140;a3=120;d5=100;thp=-pi/2;thw=pi/6;
ti=0;

% initializing the for loop with a counter variable i
for
    i=1:length(t)
        thw=pi/6;
        th1=atan2(py,px); // calculating the theta 1
        px1=sqrt(px^2+py^2);
        pz1=pz-d1;
        pxd=px1-d5*cos(thp);
        pzd=pz1-d5*sin(thp);
        cth3=((pxd^2+pzd^2-a2^2-a3^2)/(2*a3*a3));
        sth3=sqrt(1-cth3);
        th3=atan2(sth3,cth3); // calculating the theta 3
        th2=atan2(pz1,px1)-atan2(a3*sin(th3),a2+a3*cos(th3)); // calculating the theta 2
        th4=thp-th2-th3; // calculating the theta 4
        th5=thw-th1; // calculating the theta 5
        T1(i)=th1;
        T2(i)=th2;T3(i)=th3;T4(i)=th4;T5(i)=th5;
        ti=ti+0.1;
    end
```

### *C.2 Workspace*

```
% defining constant (link parameter)
d1=50; a2=140; a3=120;a 4=0; d5=100

% defining maximum and minimum variable to 5 joint variable
q1=linspace(-140*pi/180,140*pi/180,200);
q2=linspace(-pi/2,pi/2,200);
q3=linspace(-pi/2,pi/2,200);
q4=linspace(-pi/2,pi/2,200);
q5=linspace(-pi/4,pi/4,200);
```

```
[th1,th2]=meshgrid(q1,q2);[th3,th4]=meshgrid(q3,q4);th5=meshgrid(q5);

% defining the end effector position vector
c1=cos(th1);c2=cos(th2);c3=cos(th3);c5=cos(th5);c4=cos(th4);
s1=sin(th1);s2=sin(th2);s3=sin(th3);s5=sin(th5);s4=sin(th4);
c234=cos(th2+th3+th4); s234=sin(th2+th3+th4);
c23=cos(th2+th3); s23=sin(th2+th3);
px= c1.*(d5.*c234+a3.*c23+a2.*c2);
py= s1.*(d5.*c234+a3.*c23+a2.*c2);
pz = -d5.*s234-a3.*s23+d1-a2.*s2;

% plotting the workspace envelop
mesh (px,py,pz);
```

### C.3 Jacobian

```
% define the constant parameter
syms d1 th1 th2 th3 th4 th5 a2 a3 d5

% px,py, pz values from forward kinematics

px =(cos(th1)*(a2*cos(th2)+a3*cos(th2+th3)+d5*cos(th2+th3+th4)));
py = (sin(th1)*(a2*cos(th2)+a3*cos(th2+th3)+d5*cos(th2+th3+th4)));
pz = d1-a2*sin(th2)-a3*sin(th2+th3)-d5*sin(th2+th3+th4);

% Matlab function for Jacobian Calculation

J = jacobian([px; py; pz], [th1; th2; th3; th4; th5])
```

### C.4 Manipulability and dexterity index

**Step 1:** Define a counter variable i

**Step 2:** Define the maximum and minimum value of theta at each joint.

**Step 3:** Define the px, py, pz value i.e is the position of the end-effector.

**Step 4:** Enter the Jacobian matrix **J** which is in order  $3 \times 5$ .

**Step 5:** Following condition to get the manipulability index:

$$J1=J';$$

$$J2=J*J1;$$

$$J3 = \text{sqrt}(\det(J2));$$

$$\text{Mani}(i)=J3;$$

**Step 6:** Following condition to get the dexterity index:

```

kindex (i) =1/cond (J);
i=i+1;

```

**Step 7:** Finally plot it in X-Y, Y-Z, Z-X plane.

### ***C.5 CTC/PD-SMC Control law***

**Step 1:** Define a trajectory which is at least twice differentiable.

**Step 2:** Differentiate it twice and define the value of position gain and velocity gain

**Step 3:** Enter the mass matrix, Coriolis and gravity vector

**Step 4:** Calculate the error and differentiation of error such as:

```

e (1) = qd (1) -x (1) ; e (2) =qd (2) -x (2) ; e (3) =qd (3) -x (3) ;
ep (1) = qdd (1) -x (4) ; ep (2) =qdd (2) -x (5) ; ep (3) =qdd (3) -x (6) ;

```

**Step 5:** Enter the disturbance equation such as friction and payload

**Step 6:** Define the CTC law as follows:

```

s1=qddd (1) +kp*e (1) +kv*ep (1) ;
s2=qddd (2) +kp*e (2) +kv*ep (2) ;
s3=qddd (3) +kp*e (3) +kv*ep (3) ;

```

Define the PD-SMC law as follows:

In PD-SMC law along with position gain and velocity gain SMC gain  $H$

$\lambda$  as sliding slope constant,  $\phi$  a diagonal matrix determine the boundary layer  
has to defined and is given as:

```

s1=qdd (1) +kp*e (1) +kv*ep (1) +A (1) *D (1) ;
s2=qdd (2) +kp*e (2) +kv*ep (2) +A (2) *D (2) ;
s3=qdd (3) +kp*e (3) +kv*ep (3) +A (3) *D (3) ;

```

**Step 7:** Torque is calculated as follows:

```

Torque = Mass matrix*S (control law vector) + Coriolis vector +
Gravity vector;

```

**Step 8:** The differential equations are solved by Runge-Kutta 4<sup>th</sup> order numerical method with a built in function ODE45 available in Matlab.

### ***C.6 Edge Detection Techniques***

```

I=imread('image3.jpg');
% Image adjust
Istrech = imadjust(I,stretchlim(I));
%Convert RGB image to gray
Igray_s = rgb2gray(Istrech);
subplot(121)
imshow(Igray_s,[])
title('Captured Image','fontsize',14)

```

```

% Image segmentation by thresholding
%use incremental value to run this selection till required threshold 'level' is achieved
level = 0.08;
Ithres = im2bw(Igray_s,level);
%figure,imshow(Ithres)
% title('Segmented cracks')
% Image morphological operation
BW = bwmorph(Ithres,'clean',10);
%figure,imshow(BW)
% title('Cleaned image')
BW = bwmorph(Ithres,'thin', inf);
%figure,imshow(BW)
% title('Thinned image')
BW = imfill(Ithres, 'holes');
%figure,imshow(BW)
% title('Filled image')
BB = edge(BW,'canny');
%figure, imshow(BB)
% title('Edge detection')

[x,y] = ind2sub(size(BW),find(BW==0));
X = [x(1) y(1)];
Y = [x(size(x)) y(size(y))];
D = pdist2(X,Y,'euclidean');

%2.54 centimeters per inch;
% 96 pixels per inch
distance_centimeters = D * 2.54 / 96
s = regionprops(BW,'BoundingBox','Area');
area = [s.Area].';
[~,ind] = max(area);
bb = s(ind).BoundingBox;
subplot(122)
imshow(BW)
title('Crack detected','fontsize',14)
rectangle('Position', bb,'EdgeColor','red','linewidth',4);
toc
time = toc-tic

```

# Vitae

## Madhusmita Senapati

---

CONTACT INFORMATION	House No- D/155 Sector-18, Rourkela-769003, Odisha, India mobile:-+91 9040635247, +91 7064031430 email:- <a href="mailto:madhusmitamechanical2@gmail.com">madhusmitamechanical2@gmail.com</a>
AREA OF INTREST	Robotics & Automations, Machines & Mechanism & Mechatronics
EDUCATION	<p><b><i>National Institute of Technology, Rourkela</i></b> Jan. 2014 – Present <i>Master of Technology by Research</i> in Mechanical Engineering CGPA – 8.94/10 Thesis submitted, awaiting review results.</p> <p><b><i>Government College of Engineering, Keonjhar</i></b> Aug.2009 – June 2013 <i>Biju Patnaik University of Technology</i>, Rourkela, Odisha <i>Bachelor of Technology</i> in Mechanical Engineering CGPA – 7.56/10</p> <p><b><i>Ispat English Medium School, Rourkela</i></b> Jun.2007 – Mar. 2009 ICSE, New Delhi Pre-University - 75%</p> <p><b><i>Ispat English Medium School, Rourkela</i></b> Apr.2007 ICSE, New Delhi Matriculation - 89%</p>
TOOLS	Matlab, CATIA V5, Pro-E, SolidWorks, ADAMS, Arduino, LabVIEW
PROJECTS	<p><b><i>Design and Control of an Articulated Robotic Arm using Visual Inspection for Replacement Tasks.</i></b> Jan.2014 - Jun.2016 Design of a 5-axis articulated robotic arm with eye-to-hand vision system for inspection of the faulty tile and also its replacement. It includes kinematics and Dynamics Analysis, Control, Image Processing, Arduino Control etc.</p> <p><b><i>Fabrication and Testing of a Standing Split AC with water purifying facility</i></b> Jan.2009 - Jun.2009</p>

WORK  
EXPERIENCE

*Junior Research Fellow*

Oct.2013 – Mar 2015

*Project Title* – Design and Testing of an Robotic Arm for in-vessel  
Remote maintenance

*Principal Investigator* - BRFST

CONFERENCES &  
WORKSHOPS

1. Recent Advances in Manufacturing and Robotics 2016, BITS Pilani, Hyderabad Campus.
2. Short Term Course on "Principle and Computation Techniques in Multibody Dynamics", NIT Rourkela December,2013
3. Short Term Course on "Aesthetics of Scientific Documentation", NIT Rourkela

PUBLICATIONS

1. Madhusmita Senapati, J.Srinivas, V.Balakrishnan,"Visual Servoing and Motion control of a robotic device in inspection and replacement tasks", International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Vol-2, pp-324-328, 2014.
2. Madhusmita Senapati, J.Srinivas, V.Balakrishnan,"Gripper Considerations for In-Vessel Tile Handling Tasks in a Fusion Reactor Vessel", International Journal of Research in Advent Technology, E-ISSN: 2321-9637, Vol. 2, pp.-197-200, July 2014.
3. Madhusmita Senapati, J.Srinivas, "Design and Simulation of a 5-axis articulated manipulator for object replacement activities" Recent Advances in Manufacturing and Robotics 2016, BITS Pilani, Hyderabad Campus.