http://researchspace.auckland.ac.nz

ResearchSpace@Auckland

Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author’s right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage. http://researchspace.auckland.ac.nz/feedback

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form and Deposit Licence.

Note: Masters Theses

The digital copy of a masters thesis is as submitted for examination and contains no corrections. The print copy, usually available in the University Library, may contain corrections made by hand, which have been requested by the supervisor.
Abstract

While rehabilitation robots are not uncommon in the literature, they are undesirably inspired by industrial robot designs. Some of the shortcomings which are common to all these contemporary robots are, kinematic incompatibility, stiff actuation, non-backdrivability, high cost, unfriendly or intimidating appearance due to use of heavy and bulky electromagnetic actuators. Wearable robots, owing to their biologically inspired design, compliant actuation, backdrivability and safe use, are better candidates for rehabilitation robots compared to industrial robots. In recent years, wearable robots have received considerable attention and several instances such as exoskeletons, orthotics, and prosthetics have been proposed by researchers. However, there are certain challenges from the design and control perspective of wearable robots, which limit their wider implementation. Bio-inspired or biological design, kinematic compliance and holistic design optimization are the chief design issues, whereas, suitable actuation, development of appropriate physical and cognitive human-robot interaction are the essential control related concerns. Most of the skeletal joints in the human body are actuated by parallel action of a group of muscles and hence a bio-inspired wearable robot design is likely to be based on parallel mechanisms. Impending research issues associated with the use of parallel mechanism are small workspace, abundance of singularities and unavailability of forward kinematics solution. Ambulatory requirement of the wearable robots also calls for compact, light weight, and energy proficient technologies for actuators, sensors, and controllers.

This thesis explores the wide-ranging potential of wearable robots in rehabilitation in the pretext of a wearable ankle rehabilitation robot. In this research, a parallel mechanism based wearable robot for ankle rehabilitation was developed to study design and control related aspects of wearable robots in general. Arrangement of actuators, in the kinematically compliant design, had been carefully selected to allow natural foot-ankle motions while keeping the ankle joint position stationary.

A fuzzy based computational model was developed in this research to provide a unique solution for the forward kinematics of parallel robots. The proposed method is accurate and time efficient compared to previous methods proposed in the literature. The fast computation of forward kinematics has facilitated its online use in the controller replacing use of heavy inclinometers.
A complete design analysis had been carried out by mathematically formulating important performance indices affecting robot performance in three major aspects such as, kinematic, actuation and structural aspects. Initially, a single objective optimisation approach was adopted following past practice, wherein a performance index called *global condition number* was optimized. Analysis of the results shows that some of the objectives were of conflicting nature and hence the single objective approach could not optimize all the performance criteria simultaneously. Subsequently, robot design optimization was carried out using existing multi-objective optimization methods, namely, preference based optimization and the evolutionary algorithm (EA) based optimization. Interestingly, these existing optimization methods were also found to be unsuccessful due to the incompatible and contradictory nature of objectives, their large number and continuous solution space. Further investigation in the EA methodology revealed fundamental shortcomings in the existing NSGA II approach. As a result of subsequent research efforts, a major breakthrough was achieved through the development of a fuzzy dominance based evolutionary optimization method to address the inadequacies of existing EA approach. Finally, the robot design optimization was carried out using newly developed fuzzy sorting genetic algorithm (FSGA) and the wearable robot was constructed using the optimized design.

To improve the compliance of the wearable robot, light weight yet powerful actuators called *Pneumatic muscle actuators* (PMA) were used which exhibit skeletal muscle like behaviour. Construction of a dynamic model of the PMA was a difficult task owing to their non-linear and time dependent behaviour. Therefore, a Mamdani based fuzzy model was developed and optimized to accurately predict the PMA behaviour in the presence of an external force. The forward kinematics model of the robot and the dynamic model of PMA were finally incorporated in an overall fuzzy controller designed for the position control of the wearable robot.

Apart from the conceptualization of a wearable ankle robot design, optimization of two variants of fuzzy inference systems namely, Takagi-Sugeno fuzzy system and Mamdani fuzzy system as well as their distinctive uses in this thesis are important contributions of the present research. The major contribution of this research lies in the development of a fuzzy dominance based evolutionary optimization method which is a strong alternate to the predominantly used evolutionary algorithm NSGA II, which has been used in diverse optimization applications over the last two decades.
This thesis is dedicated to my beloved Guru
Swami Atmananda Saraswati
and Parents
Shanker Lal Jamwal and Geeta Devi Jamwal
Acknowledgements

I would like to express my sincere gratitude and appreciation to my mentor and supervisor Assoc. Prof. Shane Xie and acknowledge all his guidance and encouragement throughout my doctoral program. I am grateful for his useful comments, encouragements and the confidence bestowed on me during the course of PhD.

With admiration, I would like to acknowledge the valuable input to my research and collegial support extended to me by my co-supervisor Dr. K. C. Aw. He has always supported and augmented my research ideas wistfully. Thanks are due to Prof. Gordon Mallinson and the staff members of the Mechanical Engineering Department for creating a research environment by organising postgraduate workshops and research seminars.

I would like to put on record my thanks to the laboratory technicians Logan Stewart and Sarath Pathirana for their help and inputs in the construction of the prototype. The University of Auckland needs to be acknowledged for providing me with International Doctoral Scholarship to support my studies and stay in Auckland.

My sincere appreciation goes to Yun Ho Tsoi, Pavan Gamage, Jack Farrant, Tushar Guha and Shahid Hussain for providing important inputs to my prototype design and for sparing time for constructive discussions throughout my research.

Last but not least, I would like to express my gratitude to my father for inspiring me to complete my PhD and my loving mother who was always worried about my well-being during my stay in Auckland. I could never have finished my thesis without the support and love of my wife Kalpana. I am indebted to her for her endless patience and infinite generosity to allow me to work on my thesis during extended work hours. My daughter Jwalpa and son Vyom need special mention for their tender love, which was always a cool breeze to refresh me after my long work days. All along this has been a blissful journey which has bestowed the knowledge and has further kindled the quest for more knowledge.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td><strong>Chapter 1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Rehabilitation and Robots</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Wearable Robots for Rehabilitation</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1 Design challenges</td>
<td>5</td>
</tr>
<tr>
<td>1.2.2 Actuation challenges</td>
<td>5</td>
</tr>
<tr>
<td>1.2.3 Control challenges</td>
<td>6</td>
</tr>
<tr>
<td>1.3 Research Objectives</td>
<td>6</td>
</tr>
<tr>
<td>1.3.1 Development of the Wearable Ankle Rehabilitation Robot</td>
<td>6</td>
</tr>
<tr>
<td>1.3.2 Optimization of the Forward Kinematics Solution</td>
<td>7</td>
</tr>
<tr>
<td>1.3.3 Design Optimization</td>
<td>7</td>
</tr>
<tr>
<td>1.3.4 Actuator Modelling</td>
<td>8</td>
</tr>
<tr>
<td>1.3.5 Posture Control of the Wearable Ankle Rehabilitation Robot</td>
<td>8</td>
</tr>
<tr>
<td>1.4 Thesis Delineation</td>
<td>9</td>
</tr>
<tr>
<td>1.5 Chapter Summary</td>
<td>10</td>
</tr>
<tr>
<td><strong>Chapter 2 Literature Review</strong></td>
<td>12</td>
</tr>
<tr>
<td>2.1 Wearable Robots</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Contemporary Ankle Rehabilitation robots</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Forward Kinematics of Parallel Robots</td>
<td>16</td>
</tr>
<tr>
<td>2.4 Parallel Robot Design Analysis</td>
<td>20</td>
</tr>
<tr>
<td>2.5 Multi-objective Optimization</td>
<td>23</td>
</tr>
<tr>
<td>2.6 System Modelling and Control of PMA</td>
<td>24</td>
</tr>
<tr>
<td>2.6.1 Analytical modelling</td>
<td>24</td>
</tr>
<tr>
<td>2.6.2 Numerical modelling</td>
<td>26</td>
</tr>
<tr>
<td>2.6.3 Artificial intelligence modelling</td>
<td>26</td>
</tr>
<tr>
<td>2.6.4 Pneumatic muscle actuator control</td>
<td>27</td>
</tr>
<tr>
<td>2.7 Discussion</td>
<td>28</td>
</tr>
<tr>
<td>2.8 Chapter Summary</td>
<td>29</td>
</tr>
<tr>
<td><strong>Chapter 3 Wearable Ankle Robot: Development and System Modelling</strong></td>
<td>31</td>
</tr>
<tr>
<td>3.1 Human Ankle, Potential Disorders and Physiotherapy</td>
<td>31</td>
</tr>
<tr>
<td>3.1.1 Ankle Complex</td>
<td>31</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Ankle Injuries and Physiotherapy</td>
</tr>
<tr>
<td>3.2</td>
<td>Design Specifications</td>
</tr>
<tr>
<td>3.3</td>
<td>Conceptual Design and Construction</td>
</tr>
<tr>
<td>3.4</td>
<td>Symbolic Kinematic Modelling</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Inverse Kinematics</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Forward Kinematics</td>
</tr>
<tr>
<td>3.5</td>
<td>Geometric Modelling</td>
</tr>
<tr>
<td>3.6</td>
<td>Dynamic Modelling</td>
</tr>
<tr>
<td>3.7</td>
<td>Chapter Summary</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Forward Kinematics Modelling Using Modified Fuzzy Inference</td>
</tr>
<tr>
<td>4.1</td>
<td>Forward Kinematics</td>
</tr>
<tr>
<td>4.2</td>
<td>Forward Kinematics Solutions</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Newton Raphson Method</td>
</tr>
<tr>
<td>4.3</td>
<td>Fuzzy Inference Approach</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Fuzzification</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Rule-base</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Inference Mechanism</td>
</tr>
<tr>
<td>4.4</td>
<td>Optimization of Fuzzy Inference System</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Problem Formulation</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Gradient Descent Optimization</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Genetic Algorithm Based Optimization</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Modified Genetic Algorithm Method</td>
</tr>
<tr>
<td>4.5</td>
<td>Training Data and Results</td>
</tr>
<tr>
<td>4.6</td>
<td>Discussion</td>
</tr>
<tr>
<td>4.7</td>
<td>Chapter Summary</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Design Analysis of the Wearable Ankle Robot</td>
</tr>
<tr>
<td>5.1</td>
<td>Kinematic Design</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Condition Number</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Singularity Analysis</td>
</tr>
<tr>
<td>5.2</td>
<td>Actuation Design</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Force Closure and Tensionability</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Robot Workspace Analysis</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Robot Actuator Forces</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1: Instances of wearable robots (a) Exoskeleton or extenders [9] (b) Orthotic robots [10] (c) Prosthetics [11]. .................................................................................................................................4

Figure 2.1: Existing ankle rehabilitation robots. (a) The Rutgers Ankle [51]; (b) The reconfigurable ankle rehabilitation robot [20]; (c) & (d) Parallel robots with central strut [5, 49] (e) Ankle rehabilitation interface developed in [17]. (Images reproduced from the sources mentioned above). .............................................................................................................15

Figure 2.2: (a) Changing foot and ankle positions in platform type parallel robots (b) Anatomically correct arrangement of actuators to maintain ankle joint stationary. Actuators are shown by red lines. .............................................................................................................16

Figure 3.1: (a) The Ankle joint anatomy and (b) ankle motion trajectories for right foot (±θ inversion and eversion, ±ψ plantarflexion and dorsiflexion, ±φ abduction and adduction). .......................................................................................................................32

Figure 3.2: Ankle joint motions illustrated for left foot..............................................................................33

Figure 3.3: Wearable ankle robot (a) a 3D solid model of the final concept (b) The robot after construction .................................................................................................................................37

Figure 3.4: Line sketch of an actuator and its position vectors on FP and MP ........................................39

Figure 4.1: Actuator lengths (a), measured and estimated pitch angles (b) and corresponding estimation error (c) for a set of data obtained from actual robot platform using Newton-Raphson Iterative method. .................................................................................................................48

Figure 4.2: Block diagram illustrating design of the fuzzy inference system .........................................49

Figure 4.3: The fuzzy Activation Function (low, medium and high) and the tuning parameters q1 & q2 ..................................................................................................................................................50

Figure 4.4: Plot of 2, 3 and 4 Activation Functions for fuzzy actuator lengths.................................55

Figure 4.5: Pareto optimal curve realizing a trade off between accuracy and computation time (listed in Table3) for eighteen fuzzy models, optimized using GD, GA and MGA.......63

Figure 4.6: Prediction accuracy from fuzzy models optimized using GA and MGA.................63

Figure 4.7: Comparison of error distribution between GA and MGA optimized twelve fuzzy models.................................................................................................................................65

Figure 4.8: Antecedent activation functions of the finally selected 4LMGA fuzzy system....66

Figure 4.9: Actuator lengths (a), measured and estimated pitch angles (b) and corresponding estimation error (c) for a set of data obtained from actual robot platform. .................67
Figure 5.1: Design stages of the ankle parallel robot and related performance indices. ........70
Figure 5.2: Interdependency of the performance indices..................................................71
Figure 5.3: Condition number distribution versus end-effector orientations.......................76
Figure 6.1: Geometrical parameters of the (a) moving platform and the (b) FP of the ankle robot .................................................................................................................................85
Figure 6.2: Condition number versus end-effector orientations (radians) at $\psi = 0^\circ$ ........91
Figure 6.3: Condition number versus end-effector orientations (radians) at $\psi = -20^\circ$ ....91
Figure 6.4: Condition number versus end-effector orientations (radians) at $\psi = 20^\circ$ .......92
Figure 6.5: Initial geometrical parameters of the moving platform (a) and the FP (b) ..........92
Figure 6.6: Condition number (K) distribution in the robot workspace after GA-based optimization .................................................................................................................................93
Figure 6.7: Condition number (K) distribution in the robot workspace obtained using modified GA .........................................................................................................................................93
Figure 6.8: Final geometrical parameters of the moving platform (a) and the FP (b) .......94
Figure 7.1: Pareto front solutions for two-objective optimization problem .......................98
Figure 7.2: Results obtained from NSGA II ........................................................................105
Figure 7.3: Activation functions and their arrangements for two objectives wherein function (a) is preferred over function (b) ..................................................................................................................110
Figure 7.4: Preferred (A) and extreme regions (B&C) of Pareto front.................................112
Figure 7.5: Activation functions and their placement in the solution space for fronts .......116
Figure 7.6: Fuzzy fronts and degrees of fulfillments of the rules in the solution space for EFSGA...............................................................................................................................................117
Figure 7.7: EFSGA fuzzy fronts and their placement in the solution space .........................118
Figure 7.8: BFSGA Fuzzy fronts and their placement in the solution space .......................119
Figure 7.9: BFSGA Fuzzy fronts and degrees of fulfillments of the rules in the solution space ...............................................................................................................................................120
Figure 7.10: BFSGA Fuzzy fronts and their placement in the solution space ......................120
Figure 7.11: BFSGA Fuzzy fronts incorporating preference, $f_1$ is preferred over $f_2$ ........120
Figure 7.12: Flowchart representation of the FSGA process ..............................................122
Figure 7.13: Preference incorporation during fuzzification of objectives. Here (a) is preferred over (b) ...........................................................................................................................................123
Figure 7.14: Fuzzy objectives (a) Condition number (b) Robot workspace (c) Stiffness, N/m (d) Norm of actuator forces, N (e) Moment error, Nm (f) Maximum actuator force, N
Figure 7.15: Simulation results following simple FSGA approach shows that the final solution obtained lies in the third front for two different experiments (a) quick conversion, (b) slow conversion .................................................................................... 125

Figure 7.16: Results from EFSGA approach for three vital criteria with enlarged final front solutions shown in the inset. Color sequence represents fronts in increasing order starting from blue, red, green, magenta, cyan, black and grey. .................................................. 127

Figure 7.17: Results from BFSGA approach for combination of three criteria ................................. 128

Figure 8.1: Experimental set-up for single PMA testing. .................................................................................... 136

Figure 8.2: Placement of sensors in the experimental set-up. .................................................................................... 137

Figure 8.3: The overall system configuration used for the PMA testing .................................................. 138

Figure 8.4: PMA characteristics during inflating and deflating. ................................................................. 138

Figure 8.5: PMA characteristics at varying loads ......................................................................................... 139

Figure 8.6: Overview of AI models used to predict pressure inside PMA .................................................. 140

Figure 8.7: Optimization of Fuzzy models which are used to predict pressure inside PMA.......................... 140

Figure 8.8: Artificial neural network showing three layers (input, hidden and the output layers) and eight hidden layer neurons. .................................................................................... 142

Figure 8.9: Prediction errors from ANN model using testing and training data ........................................ 143

Figure 8.10: Initial design of the antecedent fuzzy variables used in the TS fuzzy modelling. ......................... 145

Figure 8.11: The fuzzy activation functions (low, medium and high) and the tuning parameters q1 & q2................................................................................................................................. 146

Figure 8.12: Surface plot showing relationship between pressure, position and force on PMA. ..................... 149

Figure 8.13: Antecedent fuzzy variables used in the TS fuzzy modelling after MGA based optimization ................................................................................................................................. 150

Figure 8.14: Trajectory following response of TS fuzzy system using (a) training and (b) testing data ......................................................................................................................................... 151

Figure 8.15: Trajectory following errors from TS fuzzy system using (a) training and (b) testing data ......................................................................................................................................... 151

Figure 8.16: Fuzzification of the pressure as a consequent variable. ............................................................ 151

Figure 8.17: Antecedent fuzzy variables used in the Mamdani based fuzzy modelling after MGA based optimization ................................................................................................................................. 155

Figure 8.18: Consequent fuzzy variable (Pressure) used in the Mamdani based fuzzy modelling after MGA based optimization ................................................................................................................................. 156
Figure 8.19: Trajectory following response of the Mamdani fuzzy system using (a) training and (b) testing data. ........................................................................................................156

Figure 8.20: Trajectory following errors of the Mamdani fuzzy system using (a) training and (b) testing data. ..............................................................................................................157

Figure 8.21: Hysteresis in PMA closely modelled by Fuzzy modelling. .................................157

Figure 8.22: Fuzzification of the antecedent and consequent variables of FLC2. ..................159

Figure 8.23: The overall fuzzy control scheme used for the wearable robot. ......................161

Figure 8.24: The overall system configuration of the wearable robot. ..............................161

Figure 8.25: Individual response of actuators to a step input signal for a real system. ........163

Figure 8.26: Trajectory following errors for individual actuators in joint space (step input signal)...........................................................................................................................163

Figure 8.27: Trajectory following response of individual actuators for a ramp input signal (0.1Hz). ..................................................................................................................................164

Figure 8.28: Trajectory following errors of individual actuators for a ramp input signal (0.1Hz). ..........................................................................................................................164

Figure 8.29: Trajectory following response of individual actuators for a ramp input signal (0.2Hz). ......................................................................................................................165

Figure 8.30: Trajectory following errors of individual actuators for a ramp input signal (0.2Hz). ......................................................................................................................165

Figure 8.31: Trajectory following response of individual actuators for a sinusoidal input signal (0.1Hz). ..................................................................................................................166

Figure 8.32: Trajectory following errors of individual actuators for a sinusoidal input signal (0.1Hz). ..................................................................................................................167

Figure 8.33: Trajectory following response of individual actuators for a sinusoidal input signal (0.2Hz). ..................................................................................................................167

Figure 8.34: Trajectory following errors of individual actuators for a sinusoidal input signal (0.2Hz). ..................................................................................................................168

Figure 8.35: Trajectory following response of individual actuators for dorsiflexion-plantarflexion trajectory while the robot is being used by a healthy subject in passive mode. .............................................................................................................169

Figure 8.36: Trajectory following response in Euler angles for three types of trajectories, namely, (a) dorsiflexion-plantarflexion, (b) Adduction-abduction and (c) inversion-aversion; and trajectory following errors (d) while the robot is being used by a healthy subject in passive mode. .................................................................................................................................169
Figure A.1: Antecedent fuzzy activation functions for 2CGD ............................................. 179
Figure A.2: Antecedent fuzzy activation functions for 2LGD ............................................. 180
Figure A.3: Antecedent fuzzy activation functions for 2CGA ............................................. 180
Figure A.4: Antecedent fuzzy activation functions for 2LGA ............................................. 181
Figure A.5: Antecedent fuzzy activation functions for 2CMGA .......................................... 181
Figure A.6: Antecedent fuzzy activation functions for 2LMGA .......................................... 182
Figure A.7: Antecedent fuzzy activation functions for 3CGD ............................................. 182
Figure A.8: Antecedent fuzzy activation functions for 3LGD ............................................. 183
Figure A.9: Antecedent fuzzy activation functions for 3CGA ............................................. 183
Figure A.10: Antecedent fuzzy activation functions for 3LGA ........................................... 184
Figure A.11: Antecedent fuzzy activation functions for 3CMGA ........................................ 184
Figure A.12: Antecedent fuzzy activation functions for 3LMGA ........................................ 185
Figure A.13: Antecedent fuzzy activation functions for 4CGD ........................................... 185
Figure A.14: Antecedent fuzzy activation functions for 4LGD ........................................... 186
Figure A.15: Antecedent fuzzy activation functions for 4CGA ........................................... 186
Figure A.16: Antecedent fuzzy activation functions for 4LGA ........................................... 187
Figure A.17: Antecedent fuzzy activation functions for 4CMGA ........................................ 187
Figure A.18: Antecedent fuzzy activation functions for 4LMGA ........................................ 188
List of Tables

Table 3.1: Range of motion and moment requirement at the human ankle.........................33
Table 3.2: Typical range of motion at the human ankle and corresponding robot capabilities ...........................................................................................................................................................38
Table 4.1 Order of TS fuzzy system and consequent parameters.................................52
Table 4.2 Types of fuzzy models studied and their tunable parameters.........................55
Table 4.3 Prediction accuracies (average of MSE in three Euler angles) obtained using various fuzzy models and three kinds of optimization methods.................................62
Table 4.4 Summary of previous work on forward kinematics of parallel robots using Artificial Intelligence (AI) approaches.................................................................................................................65
Table 6.1: Comparison of robot designs obtained using single objective optimization approach (geometrical dimensions are in degrees and mm)....................................................95
Table 7.1: Results from preference based multi-objective optimization.........................104
Table 7.2: Representative design optimization results from NSGA II..............................106
Table 7.3: Limitation of the non-dominance concept.......................................................107
Table 7.4: Activation scores of linguistic variables..........................................................113
Table 7.5: Activation scores of linguistic variables..........................................................117
Table 7.6: Final robot design solutions from *EFSGA* and *BFSGA* compared with NSGA II final result (F1: Condition number, F2: Percentage workspace utilization, F3: Stiffness (N/m), F4: Norm of actuator forces (N), F5: Normalized error in moment (Nm), F6: Maximum actuator force (N)). Robot design parameters q1, q2...q8 have been explained in (Figure 6.1) ...................................................................................................................132
Table 8.1: The complete weight matrix and the bias used in the hidden layer of ANN........143
Table 8.2: Rule-base table used in FLC2 to determine fuzzy pressure values.................160
Chapter 1 Introduction

Robots are used to augment productivity, accuracy and precision in applications which are repetitive, inhospitable and impractical for human beings. Applications of robots are apparent in factories, laboratories, warehouses, military and shipment activities. Although robots are used in major aspects of life, their use in close human proximity, such as service and healthcare applications, is still limited due to design, interaction and safety issues. Given the rising elderly population worldwide and obligations to meet the escalating demand for healthcare services, there is a growing demand for robotic solutions. This research is in accordance to this new development and aims to investigate design, safety and control issues of wearable rehabilitation robots in the pretext of an ankle rehabilitation robot. This Chapter presents an overview of various research issues encountered in the course of developing a wearable robot. Importance of robots in rehabilitation will be briefly discussed followed by the basic information on the ankle joint anatomy to determine specifications of the ankle robot. Research goals and objectives will be presented, followed by a brief discussion on the structure of this thesis.

1.1 Rehabilitation and Robots

Rehabilitation in a broad sense means a practice, by which different forms and grades of human physical disorders can be reinstated. The disorder could be the result of an injury or a stroke. Conventionally, to restore the range of motion and strength of limbs, rigorous and repetitive exercises are performed under supervision of a therapist. Over time, these exercises improve motor functions by enhancing neuro-plasticity and neuro-recovery of affected limbs. Commonly, during a rehabilitation treatment, cooperative and intensive efforts of therapists and patients are required over prolonged sessions of treatment in a clinic, and very often patients are required to continue the prescribed exercises at home for a speedy recovery. It has been documented that using conventional ways of treatment, the recuperation is slow and sometimes continues for more than a year [1]. There are three stakeholders of the treatment namely, patients, therapists and the rehabilitation process and all three suffer from the drawbacks of conventional treatment. Difficulties from the patient’s perspective are:
1.1 Rehabilitation and Robots

1. Travelling difficulty: Patients with disability are often struggling to frequently travel to rehabilitation clinics or medical centres. Special arrangements are normally required which may affect the injury.

2. Prolonged clinical sessions: Clinical visits are time consuming which include travel time and the waiting time.

3. Fatigue: Exercises advised by therapists are often monotonous and tiring which cause fatigue to patients. As a result of this, patients are not able to achieve stipulated rehabilitation goals.

Therapists face following problems during conventional rehabilitation treatment:

1. Fatigue: Strenuous and repetitive efforts during manual treatments cause fatigue in therapists.

2. Inadequate output: Therapists can only attend to limited number of patients due to their physical involvement in prolonged sessions of rehabilitation treatment.

3. Decision inaccuracy: Due to lack of a well structured and documented history of the patient’s improvement, therapists normally advise further treatment based on their own perception which may not be always correct.

Rehabilitation process which is also a stake holder of the treatment, suffers from following drawbacks:

1. Inconsistency in treatment: Lack of repetitive efforts from patients, while applying motion and force trajectories in the manual treatment, results in the inconsistent treatment.

2. Therapeutic subjectivity: Therapists evaluate recuperation and recovery based on their perception which may lead to undesired subjective decisions in the course of treatment.

3. Slow recuperation: Due to its monotonous and tiresome nature, the conventional therapy lacks active participation from patients which ultimately results in slow recovery.

Robots can play an important role in improving the rehabilitation process by assisting therapists and patients in a number of ways. Robot assisted treatment could be received while performing routine duties such as reading writing, and working with computer. Visual and haptic effects can be appended with robots to make exercises interactive and motivating [1]. It has been noted that active participation from patients, enhances the effectiveness of the treatment [2]. Patients can avoid frequent clinical visits and receive rehabilitation treatment sitting at home or workplace using robots. Apparently, using robots, patients can save on
time, efforts and money [2]. By using robots at rehabilitation clinics, physical efforts of therapists can also be reduced to a large extent. Therapists can treat more number of patients since their involvement in the treatment becomes less intensive with the support of robots [3]. It has been noted that using robots (equipped with sensors), vast data related to the treatment can be collected and processed into useful information. Using this information, the evaluation and treatment of injuries can be performed more objectively. When patients are using rehabilitation robots at home, remote connections can be easily established with robots used by patients and get the required information about patient’s progress. Based on patient’s improvement information therapists can make decisions about further treatments and these can be communicated to patients at their homes. Owing to the high repeatability of robotic treatment, more consistent physical therapy can be extended to patients, improving the overall effectiveness of the treatment. With the introduction of visual and haptic effects, patient’s active participation can be increased which will further expedite the recuperation. Therefore, application of robots in rehabilitation process can benefit all the stake holders such as, patients, therapists and the rehabilitation process.

There are rehabilitation robots currently in use such as MIT-MANUS for the upper limb rehabilitation [1], LOKOMAT for gait training [4] and Rutgers Stewart platform and other parallel robots [5] for ankle joint rehabilitation. However, these robots share certain common issues concerning their design and actuation which require further investigation.

1.2 Wearable Robots for Rehabilitation

Study of previous rehabilitation robot designs reveals that most of these robots are yet another form of industrial robots. Robots used in the industries are employed to replace human in operations involving tedious, repetitive and hazardous activities. Contrary to the rehabilitation robots, industrial robots are highly stiff, heavy, non-backdrivable and their design and control schemes are specific to a single task e.g. pick and place, locate and drill etc. Rehabilitation robots are different [6] from their industrial counterparts in application and operation and hence different approach is required for their design, actuation, modelling and control. Rehabilitation robots function in close proximity of the human operator interacting both, physically as well as cognitively, therefore they are desired to be compact, light weight, comfortable, compliant, safe, and user friendly in operation. This calls for ergonomic and optimized designs, supported by powerful yet light weight actuators and intelligent and adaptive robot controllers. The design, actuation and control of these robots are challenging
1.2 Wearable Robots for Rehabilitation

tasks requiring multi-disciplinary skills and in-depth knowledge of human anatomy and rehabilitation objectives.

The desired characteristics of rehabilitation robots can be realized in the realm of Wearable robots, a class of robots which has evolved over the last two decades since its first application [7]. As shown in Figure 1.1, there are three instances of wearable robots namely, extenders, orthoses and prosthetics [8]. Extenders or the exoskeletons extend human strength to perform tasks beyond normal human capacity. Orthotic robots are employed to reinstate lost or weak functions of a limb following an injury or neurological condition. Construction and controls of orthotic robots are designed to emulate the anatomy of the affected limb. Prosthetics are the robotic aid used to stand-in for lost limbs following an amputation.

![Figure 1.1: Instances of wearable robots (a) Exoskeleton or extenders [9] (b) Orthotic robots [10] (c) Prosthetics [11].](image)

It has been recommended in literature, that wearable robots are more appropriate to be used for rehabilitation treatments compared to their industrial counterparts [8, 12]. Rehabilitation robots are a hybrid case of extenders and orthotic robots and can be explained on the basis of their force interaction with the human subject. Depending on the application, the physical human-robot interface can have two kinds of interactions namely an internal force interaction, wherein patient efforts are predominant or an external force interaction, wherein robots supply required rehabilitation forces. Wearable robots, when used as extenders to enhance human capabilities, interact on the external force interaction basis. Conversely, the orthotic robots, used for the function compensation of the injured or non-functional limbs, work on the basis of the internal force interaction. Rehabilitation robots work on both kinds of forces interactions. During an initial phase of rehabilitation treatment, the interaction is purely based on an external force, similar to extenders, and over the time
1.2 Wearable Robots for Rehabilitation

while the subject is improving; the interaction gradually switches to the internal force interaction.

Wearable robotics is gradually becoming an emerging research field and a few applications of rehabilitation robots have been proposed to reinstate activities of daily life (ADL) in people suffering from motor disorders [12]. However, there are certain challenges from the design, actuation and control perspective of wearable robots, which constrain their use as rehabilitation robots. These aspects can be further elaborated as follows:

1.2.1 Design challenges

Bio-inspiration plays an important role in the conceptualization and optimization of the wearable robot designs. Design challenges for a wearable robot come from:

1) Bio-inspired design: A bio-inspired design is required to comply and conform to the anatomy of human limbs and to be able to maintain the kinematic compatibility during motions.

2) Parallel actuation: Actuation of most of the skeletal joints in the human body is achieved by parallel action of a group of muscles and hence a bio-inspired wearable robot design is likely to be based on parallel mechanisms. Challenges arising from the use of parallel mechanism are small workspace, abundance of singularities and unavailability of forward kinematics (FK) solution.

3) Design optimization: In order to obtain a compact, miniaturized and portable robot, its design parameters are required to be optimized. Owing to the large number of objectives to be optimized in a continuous solution space, existing evolutionary algorithms (EA) have been found less effective [13]. In this scenario, It is required that newer optimization methods, based on EA be investigated and devised. Use of EA for the optimization of design [8] is another important application of bio-inspiration in wearable robots.

1.2.2 Actuation challenges

Actuators used in wearable robots should exhibit characteristics which are close to the skeletal muscle. Challenges in the context of actuation come from:

1) Selection of actuators: The desired attributes of the actuators, which are compliance, flexibility, high power to weight ratio and safe actuation, are difficult to achieve.
2) Actuator modelling: Pneumatic muscle actuators (PMA) possess the desired characteristics for a bio-inspired actuator, however it is a challenge to model their non-linear and time dependent dynamics.

1.2.3 Control challenges

Force and position control of wearable robots is a difficult task owing to:
1) The parallel mechanism based design: Robot orientations are difficult to obtain using simultaneous control of several actuators acting in parallel.
2) The use of PMA for actuation: The dynamic modelling of PMA characteristics is difficult to obtain.
3) Rehabilitation application: Their application into rehabilitation treatments involves rigorous human-robot interactions. These interactions are of two kinds, namely, physical as well as cognitive interactions. It is difficult to formulate control rules to achieve appropriate interactions.

Apart from the above mentioned challenges, portability and ambulatory requirement of the wearable robots require compact, light weight, and energy proficient technologies for actuators, sensors, and controllers [12, 14, 15].

1.3 Research Objectives

The goal of this research is to investigate various research issues concerning wearable rehabilitation robots in the pretext of a wearable ankle rehabilitation robot. Therefore, in this research, a parallel mechanism based wearable robot for ankle rehabilitation will be developed to study design, actuation and control related aspects of wearable robots. Although, design, actuation and control aspects of wearable robots shall be investigated, more emphasis will be given to their design aspects due to its vitality. The research goal can be further explained with the help of following research objectives:

1.3.1 Development of the Wearable Ankle Rehabilitation Robot

A prototype of wearable ankle rehabilitation robot shall be developed to provide a physical platform to perform experiments and investigations during the course of this research. A bio-inspired design shall be developed after carefully studying the basic anatomy of the human ankle joint along with its motion and force capabilities. The wearable ankle robot design shall be adaptable to users of varying physical capabilities such as range of motion, ankle joint stiffness and strength. Unlike previous ankle rehabilitation robots
1.3 Research Objectives

mentioned in the literature [5, 16-20], the wearable ankle robot shall be, light weight, portable, and kinematically compatible to the ankle joint motions.

1.3.2 Optimization of the Forward Kinematics Solution

This research objective develops a new computational model for providing FK solutions, which will be used in real time for the control of the proposed wearable ankle robot. This new model shall be developed using fuzzy inference systems. To enhance computational efficiency and accuracy of the fuzzy based FK model, its parameters shall be optimized using a modified genetic algorithm (MGA).

The wearable ankle robot, which is being proposed in this thesis, shall essentially be a parallel robot. In the context of parallel robots, it has been widely reported in the literature that their FK solution cannot be obtained in a closed form due to their parallel actuation of joints [21]. It has been shown that more than one solution of the joint actuation exist, to achieve a given orientation of the end-effector of the parallel robot [22]. Existing computational models, available in the literature to solve FK of parallel robots, are computationally expensive and thus cannot be used for online control of robots [23]. Approaches based on numerical methods and artificial intelligence (AI) shall be investigated to develop a new computational model, capable of solving FK of wearable robots, which can offer reduced computational complexity while providing improved accuracy.

1.3.3 Design Optimization

A comprehensive design analysis of the wearable ankle robot shall be carried out wherein important robot design performance indices shall be discussed which will be used to evaluate competing robot designs. The design analysis shall be executed by carefully studying three major aspects of robot design such as, kinematic design, actuation design and structural design aspect. Robot design performance indices shall be mathematically formulated to allow their methodical analysis. After formulation of performance indices, design optimization of the wearable robot shall be carried out. Two major optimization approaches namely, single objective optimization and multi-objective optimization shall be used for the design optimization. While using single objective optimization approach, all the design objectives shall be normalized and added to form a single objective whereas in multi-objective optimization approaches they will be optimized independently. Evolutionary algorithms (EA) shall be used while performing a multi-objective optimization.
1.3 Research Objectives

It will be illustrated in Chapter 7 that EA loses its capabilities of discrimination between vital few and useful many, if the number of objectives is large and the solution space is continuous. In the light of this limitation the existing EA cannot be used in the multi-objective optimization of wearable robots where both the above conditions hold true. Therefore in the present research the existing EA shall be revisited and new multi-objective optimization approach shall be devised.

Lastly, the design of wearable ankle robot being developed shall be adaptive to accommodate subjects of different physique, gender and age.

1.3.4 Actuator Modelling

This research objective aims to develop a dynamic model of PMA to appropriately model their time dependent and non-linear characteristics. It will be shown in Chapter 3 that, in order to achieve kinematic conformity and actuation compliance, flexible PMA are required to be used in series with cables to actuate the wearable robot. Artificial intelligence (AI) based methods shall be investigated to model PMA with enhanced accuracy and interpretability. Wearable rehabilitation robots work on both kinds of interactions i.e. internal and external force interactions and hence it becomes important to consider the external force acting on PMA while carrying out dynamic modelling. During present modelling work, experiments shall be designed and carried out to record effects of the external force on PMA characteristics. This data shall be utilized while developing the AI based dynamic modelling of PMA.

While quite a few research attempts have been made to model PMA, earlier models are not accurate enough and lack interpretability [24]. Previously developed dynamic models are also not robust for disturbance in the form of external force acting on PMA [25].

1.3.5 Posture Control of the Wearable Ankle Rehabilitation Robot

In order to use the wearable robot in ankle rehabilitation treatments, a new fuzzy logic based controller will be developed. The controller is required to actuate wearable ankle robot in order to realize range of motion therapy and achieve rehabilitation goals. It is desired that the robot motions remain robust against external disturbances which are expected in the form of forces exerted by patients. Owing to the non-linear and time-dependent characteristics of the PMA coupled with parallel robot complexities, it can be asserted that the conventional controllers cannot be used in the present case. The dynamic model of PMA, discussed in the previous paragraph, shall be used in an overall control scheme to achieve the position control
of the wearable ankle robot. Initially, experiment shall be conducted without human intervention to investigate position accuracy of the robot against simultaneous actuation of its links. Subsequently, experiments with a healthy subject shall be carried out to examine controller’s performance when the robot interacts with an external environment.

1.4 Thesis Delineation

The research work, carried out in the order of objectives mentioned in the preceding Section, is carefully documented in this thesis. Research process and its various outcomes are described by organizing them into nine chapters including a Chapter on introduction.

A state-of-the-art literature review is documented in Chapter 2. The focuses of the review are placed on topics that are essential to the development of wearable robots. They include contemporary rehabilitation robots, kinematics and dynamic analysis of parallel robots, design analysis of parallel robots, multi-objective optimization, and PMA modelling and control. Chapter 3 proposes a wearable robot for ankle rehabilitation based on the understanding of human ankle anatomy. This chapter includes an introduction to the conceptual design, its construction, and the symbolic modelling of the robot structure. The inverse kinematics analysis will be carried out in the Chapter. The Chapter concludes with the geometrical modelling and the dynamic analysis of the wearable ankle robot.

A new fuzzy inference approach to solve the FK of the wearable robot is introduced in Chapter 4. It has been emphasized that by optimizing the parameters of fuzzy inference system, the accuracy of FK solution can be greatly enhanced. Consequently, a MGA has been proposed in this chapter to optimize the fuzzy inference approach.

Chapter 5 investigates the design analysis of the wearable robot. In this Chapter, the mathematical formulation of the key performance indices (PI) of the proposed wearable ankle robot is presented. The design analysis process is categorized into three levels namely, kinematic design, the actuation design and the structural design. The interdependence of PIs along with their dependence on the geometrical parameters of the robot is investigated.

Chapter 6 formulates the design optimization problem of the proposed ankle robot mathematically. The key geometrical parameters of the robot to be optimised are identified. A single objective approach is introduced which uses the MGA to solve the design problem and its results are presented.

To improve the design of the wearable robot, Chapter 7 investigates new developments on multi-objective design optimization considering the six important performance indices. A
preference based method is applied for optimization and the results are analysed in the Chapter. Evolutionary algorithms (EA) are also developed in this Chapter. Non-dominated Sorting Genetic Algorithm II (NSGA II) is implemented to solve the design optimization problem of the proposed robot. The performances of the algorithms are compared and analysed.

A new concept of fuzzy dominance is proposed in this Chapter to address the shortcomings found in the non-dominance criterion. Design optimization of the wearable robot is conducted using two selection approaches based on fuzzy dominance, namely, equitable fuzzy sorting genetic algorithm (EFSGA) and biased fuzzy sorting genetic algorithm (BFSGA) and optimization results from these are analyzed and compared with those obtained using NSGA II.

Chapter 8 describes the use of artificial intelligence to develop non-linear dynamic models for the PMA. In this Chapter, artificial neural network and two instances of fuzzy logic namely, Sugeno fuzzy inference and Mamdani fuzzy inference are investigated and their prediction capabilities are compared. Mamdani inference based fuzzy model is recommended on account of its better interpretability over other methods. The non-linear dynamic model so obtained is further incorporated into an iterative fuzzy logic controller. This controller is used to achieve desired orientation of the wearable ankle robot in the presence of the external force. Performance of the fuzzy logic based controller is evaluated in details.

Finally, Chapter 9 concludes all the important results obtained from the present research work. Apart from the outcomes and the conclusions, this Chapter also outlines the contributions of the research to literature. A discussion on the future research for the wearable ankle robot is also provided to help prospective researchers. During the course of this thesis several publications in international journals and conferences including an invited book chapter were made [23, 26-34]. Several Sections of thesis are therefore based on these publications

1.5 Chapter Summary

This Chapter discussed the benefits of an automated rehabilitation treatment in context to patients, therapists and the rehabilitation process at large. Requirement of a wearable solution for rehabilitation was highlighted after discussing the limitations of contemporary designs of
rehabilitation robots. Instances of wearable robots were explained and the importance challenges concerning design, actuation and control of wearable robots were emphasized.

Main research objectives were discussed with brief notes on the motivation behind them. The first objective, set for the research, was to develop a wearable ankle rehabilitation robot which shall be used for the study issues related to the wearable rehabilitation robots at large. Development of a computational model to obtain fast and accurate solution of the FK for this robot was considered as the next objective. Thereby the use of heavy sensors in estimating robot pose shall be avoided. Due to its distinctiveness, analyses of the wearable robot design and its subsequent optimization were considered to be the major focus of this research. Given the time dependent and non-linear behaviour of PMA, their dynamic modelling was treated as an important objective. Actuator modelling was also a prerequisite to the design of the overall controller for wearable ankle robot. Design and implementation of a fuzzy logic based controller for position control of the robot was performed as a concluding research activity.
Chapter 2 Literature Review

A comprehensive literature review is carried out to identify the fundamental research issues involved in the development of wearable robots. Wearable ankle rehabilitation robot is being considered as a case study in the present research and thus existing ankle rehabilitation robots are discussed in the light of their design, construction and control. Subsequently, previous work on FK solution of parallel robots is reviewed and a classification of approaches is presented thereon. Following this, design analysis and optimization of parallel robots is discussed and related past research work is presented in a chronological order. A survey of multi-objective optimization methods is also included in the present discussion. Previous research related to the modelling and control of PMA is reviewed and documented by appropriately categorizing. The key issues in the above research areas are discussed which provided a foundation for the present research.

2.1 Wearable Robots

Wearable rehabilitation robotics is a new research area which has stemmed out from the field of active orthoses. These active orthoses are training robots which work in parallel with human body and have mechanical actuation to apply forces to human limbs. The first ever instance of an active orthoses was found in [35] wherein a complete lower limb exoskeleton was conceptualized and developed. University of Wisconsin prototype [36] was another such example, which employed universal joints at the hip and ankle to provide kinematic compatibility between orthosis joints and skeletal joints. In the later developments, some of the active orthoses such as LOKOMAT [37], MANUS [1] and Autoambulator [38] were developed, tested in clinics and are now commercially available. Nevertheless, the field of active orthoses is still an active research area and improvements in their design and interaction control approach are being investigated. Recently, an improved design of robotic orthosis, called active leg exoskeleton (ALEX) was developed at the University of Delaware for gait training of stroke survivors [39]. Lower extremity powered exoskeleton (LOPES) employing Bowden cable based actuation system is another instance of a recent development in rehabilitation exoskeleton [40]. Active orthoses have also been proposed for the ankle joint
by researchers in past, however these orthoses were mainly developed for the purpose of locomotor rehabilitation of stroke survivors to overcome foot-drop problem [41, 42]. Since these orthoses were intended for gait corrections, only flexion motion of the ankle joint was considered during their design. A state of the art review which follows evolution of active orthoses can be found in [43].

After a review of the related literature, it has been found that, though the wearable robotics is an emerging research area; only a few prototypes have been proposed by researchers. Since its inception in 1974, only a small number of robot designs have been developed to be used in clinics. A possible reason could be the multidisciplinary nature of the research issues involved in the development of wearable robots. Potential design, actuation and control related challenges in the development of wearable robots have also been discussed in Chapter 1.

2.2 Contemporary Ankle Rehabilitation robots

For ankle rehabilitation, typical devices such as elastic bands, wobbles boards and foam rollers are being used, but they allow only basic and functional rehabilitation exercises. On the other hand, commercially available rehabilitation units such as, Multi Joint System and Stability system are expensive, multi-functional and are not specific to ankle joint treatments [44]. Therefore to address the requirement of an affordable, versatile and automatic solution, robotic devices for ankle joint rehabilitation have been developed [5, 16-20]. Owing to higher stiffness and relatively small range of motion requirements for the ankle joint, these robot designs are based on parallel mechanisms. These robots can be used for rehabilitation treatments to reinstate initial range of ankle motions and advanced muscle strength training. Range of motion exercises of the ankle joint are basically of three kinds such as passive, assist-as-needed and active exercises [45]. During passive mode patients are expected to remain inactive and let the robot move the ankle and foot on a prescribed trajectory. When certain degree of improvement in the ankle joint motion is achieved, the robot switches to an assist-as-needed mode and patients are encouraged to move foot and ankle joint on prescribed trajectories. When the ankle motions are fully recovered, the robot becomes passive allowing patients to move their ankle and foot actively. Finally, upon regaining partial muscle strength, patients are put into further challenging exercises wherein role of the robot is to resist the motion with increasing degrees of resistance. Contemporary robot designs have been studied critically to provide inputs to the present research.
In one of the earliest works, Girone et al. proposed the Rutgers Ankle that used a Stewart platform, capable of providing six degrees of freedom (dof) to the ankle joint [46]. Double acting pneumatic cylinders are used as actuators to move the end-effector or the moving platform (MP) of the robot. Patient’s foot is required to be fixed firmly on top of this platform and assistive or resistive moments are applied for the passive and active modes of exercises respectively. This platform is further interfaced with the game-like virtual environments [47] to make the exercises more interesting and entertaining for patients. Apart from sprained ankle treatment the Rutgers Ankle is also used in the clinical trials for post-stroke rehabilitation [48]. Although Rutgers Ankle has been well developed and is being used for scientific experiments, its redundant actuation is a serious drawback i.e. for three dof ankle motions; Stewart platform with six dof is being used. The position of the ankle joint in the robot does not remain stationary, causing inconvenience to patients and difficulty in designing a suitable controller. In order to reduce the redundancy of the above Stewart platform, Dai et al. [5, 49] proposed a parallel robot for sprained ankle treatments using a three and four dof parallel mechanisms with a central strut. Kinematics and stiffness analysis have been carried out for their proposed mechanism. Using different types of central struts, authors have analyzed three different types of parallel robots in the domains of stiffness. However, while redundancy of the robot actuators is reduced, patient’s inconvenience due to change in the position of the ankle joint could not be addressed.

Another instance of parallel robot used for ankle joint rehabilitation is found in [20, 50] where a single platform-based reconfigurable robot mechanism has been proposed. Apart from the ankle joint motion, this robot design considers the metatarsophalangeal (MTP) joint, which exists between the fore and rear of the foot, and the robot still has less than six dof motions. Since it is a reconfigurable robot, the same platform can be used for range of motion (ROM) controlling, muscle strengthening and proprioception training. An impedance controller was proposed for this robot and impedance parameters were varied to accommodate different exercise modes. A 3-RSS/S parallel mechanism is proposed by [17] and the kinematic design of its prototype is validated using simulations. Syrseloudis and Emiris [19] have finally proposed a tripod based parallel robot actuated by electric motor, after evaluating several serial and parallel robot solutions for the ankle rehabilitation robot.

So far all the platform type robots require patient’s foot to be placed on top of a platform which is actuated from the bottom. These robots have a fixed base and thus are not portable. Apart from the problems related to the fixed base, non-compliant actuations and higher costs, there are certain other issues with such configuration, firstly, when the end-effector
containing the patient’s foot (fixed on top of the platform) is moved, the position of the ankle joint and the shinbone keeps changing with respect to the ground (Figure 2.1e). This instability in the position of the ankle joint leads to control errors which are difficult to comprehend.

![Contemporary Ankle Rehabilitation Robots](image)

**Figure 2.1:** Existing ankle rehabilitation robots. (a) The *Rutgers Ankle* [51]; (b) The reconfigurable ankle rehabilitation robot [20]; (c) & (d) Parallel robots with central strut [5, 49] (e) Ankle rehabilitation interface developed in [17]. (Images reproduced from the sources mentioned above).

Such designs are not truly biological and are not kinematically compatible with the ankle joint. The illustration (Figure 2.1e) shows the obvious discomfort caused to patients due to large shifts in the foot-ankle-leg positions. With such a configuration, measurement errors between the robot end-effector orientations and ankle joint displacements cannot be avoided. These errors finally result into lower precision of the trajectory following motions and limit the use of robot as a measurement tool [52].

It is important to mention here that the dynamic model of the robot in such cases should include the dynamic inertia of the patient’s shinbone which is difficult to estimate. In the absence of an accurate dynamic model, large trajectory errors are expected. In [47] authors have proposed an *Inside Track-3D* tracker to measure the position and the orientation of
patient’s shinbone to avoid trajectory errors and prevent ankle movements beyond specified ROM. However, movements of the shinbone relative to the ankle joint definitely causes discomfort to patients as they are required to change sitting position intermittently as also shown in Figure 2.2a.

![Figure 2.2: (a) Changing foot and ankle positions in platform type parallel robots (b) Anatomically correct arrangement of actuators to maintain ankle joint stationary. Actuators are shown by red lines.](image)

New solutions need to be developed to effectively address the design concern mentioned above. One of the possibilities is to place actuators in an anatomically correct configuration as shown in Figure 2.2b, which is close to the musculoskeletal arrangement in the human leg. With such arrangement of actuators, position of the ankle joint and the leg shall remains stationary with respect to the foot during different exercise trajectories and modes.

As can be seen, the existing ankle robots designs are not anatomically compatible to human ankle joint. Apart from their incorrect design, they are heavy, immobile and due to the use of large electromagnetic actuators they have an intimidating appearance for the uninformed patient. The actuators used in these robots are heavy, non-compliant, non-backdrivable and rigid. Wearable robots shall be discussed in this thesis for their possible use in the ankle rehabilitation to improve the design, actuation and control aspects of contemporary ankle robots.

### 2.3 Forward Kinematics of Parallel Robots

Parallel robots normally have two platforms, a fixed platform (FP) and a moving platform (MP) or an end-effector. These platforms are joined together with appropriate links to develop the robot structure. Lengths of these links are controlled to obtain required orientation and position of the end-effector. While developing control algorithms to achieve position control of robots, two types of mathematical models, a kinematic model and a
dynamic model are often required. The kinematic model describes the relationship between actuator link length/displacement and end-effector pose/position. Similarly, the dynamic model describes the dynamic behaviour of robot and is required for gravity and dynamic compensation in control schemes such as computed torque control and impedance control [53]. During the kinematic analysis a correlation is found between joint (links) and Cartesian coordinates (end-effector) with reference to the stationary FP. This correlation is defined in two ways i.e. Forward Kinematics (FK) and Inverse Kinematics (IK). The FK model is constructed to compute the position and orientation of the end-effector as a function of kinematic parameters of its joints/links. Likewise, the IK model is developed to find the set of joint/link parameters that would bring the end-effector to a desired location/orientation in the workspace. It has been established that IK solution for serial robots is difficult to obtain in closed form since more than one feasible solution exists. Conversely for parallel robots, the FK analysis involves a set of highly coupled nonlinear equations for which closed form and unique solution cannot be found using conventional tools [54]. The FK problem of a 6-6 Stewart platform can be formulated as explained below.

If \( a_i \) and \( b_i \) are the connection points at the end-effector and FP respectively then following kinematic equation (2.1) is required to solve while carrying out FK. Here \( P \) is the translational vector and \( R \) is the rotational transformation matrix of the end-effector with respect to FP for given vector of link lengths \( L_i \).

\[
\|P + Ra_i - b_i\|^2 = L_i^2 \quad i = 1, ..., 6. \tag{2.1}
\]

In fact, a number of possibilities of end-effector poses exist for a given set of link lengths which are all mathematically correct. However, out of these poses, only one solution is the desired solution for real time control applications. Alternatively speaking, the valid solution for the end-effector pose is the one which is reachable from the initial configuration without having to pass through singular points [22]. Forward Kinematics problem of parallel robots due to its intricacy has always attracted researchers and quite a few approaches have been proposed over the last decade. These approaches can be categorized into the following four groups namely, closed-form solutions for special configurations; analytical approaches; numerical methods and artificial intelligence based methods.

A 6-6 Stewart platform has as many as 64 possible end-effector poses for a given set of input joint vectors [55]. The number of solutions can be greatly reduced if the connection points at either or both the platforms are merged together to construct some special structures.
2.3 Forward Kinematics of Parallel Robots

It is then possible to solve FK for such modified designs in *closed form* and the earliest work known to the author in this regard was proposed by Griffis and Duffy [56] for a 3-3 Stewart platform. In another approach, [57, 58] loci of the connection points at the platforms were determined by considering that the platform is momentarily disconnected. A novel approach was proposed by [59] wherein a part of the parallel mechanism was converted to an equivalent serial mechanism. The constraints imposed by the remaining part of the mechanism were utilized to find constraints on the joint angles of the first part to further solve the FK of the equivalent serial mechanism and thereby for the entire parallel mechanism. In a pioneering work [60], comprehensive survey of almost all kinds of (35 different types) Stewart platforms has been presented. Maximum number of possible solutions for each design have been worked out and listed. Geometrical approaches [61-63] to obtain closed form solutions to the FK have also been proposed for variety of parallel manipulators. Use of tetrahedron theorem to find a unique closed form solution has been a representative and pioneering work. An algebraic method [64] was presented to derive a 20th order univariate polynomial for a 6-6 Stewart platform with planar FP and end-effector. A 3-dof medical parallel robot was proposed in [65, 66] however due to the constrained motion of its end-effector, the FK and IK analysis became very trivial. Recently an algebraic method, in continuation of the previous work in [64], has been proposed in [67] which uses reduced Grobner bases and lexicographic ordering to obtain a univariate equation of higher order.

While using *analytical approaches*, the platform orientation is represented by the orthogonal direction cosine matrix, instead of conventional Euler angles, facilitates linearization of the system kinematic equations (2.1). Following this argument, FK problem of a 3-dof in-parallel actuated manipulator has been successfully solved in [68] adopting analytical approach. Similar approach has been found in various other works [55, 69] wherein the degree of univariate polynomial has been reduced to lower the computational complexity. Geometrical arguments [70-74] have been helpful in determining the maximum number of the platform poses and definite formulation of FK problem. Forward kinematics analysis of a 3-dof parallel robot which is a part of 5-dof reconfigurable robot has been presented in [75] using analytical approach.

Numerical methods which are used to solve non-linear equations are good candidates for the FK solutions and can be employed in all the real time applications where only one real solution is required. However the prerequisite of having a good starting solution is a major drawback of these approaches. Non-linear equation solving algorithms such as Powell’s method [76] and Newton-Raphson method [77, 78] have been used to solve FK problem of
2.3 Forward Kinematics of Parallel Robots

parallel robots. Taking a different approach, Innocenti and Parenti-Castelli [79] have proposed to reduce the dimension of the problem using geometrical methods to 3 equations in 3 unknowns from 6 equations in 6 unknowns of the Stewart platform. To obtain all the real solutions, an efficient 3-dimensional search strategy has also been used in [80] which utilizes only geometrical considerations. A pioneering work in this regard has been done by Raghwan [81], who used a continuation method to solve a system of equations and found 40 solutions in the complex domain, tracking 960 paths from the start point. A novel numerical method has been proposed in [82] which uses the instantaneous velocity direction of the mobile platform to find its new position. In order to obtain a unique solution to FK problem, Wang and Oen [83] have used non-linear programming and Newton-Raphson methods in succession to solve FK of a fully parallel robot. Sadjadian and Taghirad [21] have presented a detailed comparison of three numerical methods namely, Neural network estimation, the quasi-closed solution and the Taylor series approximation. An iterative routine has been presented in [84] to solve FK problem of a macro-micro parallel manipulator. The proposed algorithm was originally given by T. Dekker wherein a combination of bisection, secant, and inverse quadratic interpolation methods were used [85].

Neural networks (NN) and genetic algorithms (GA) have also been used by researchers to solve the FK problem for a variety of parallel robots. These are model based approaches and do not require a solution of complex kinematic equations. Using inverse kinematics (IK) analysis, one can create a database of end-effector positions, orientations and corresponding joint variables. Subsequently, this database can be used to identify the parameters of the inference model being used. The model so developed can be used later to predict end-effector poses for a given set of joint variables. The earliest discussion on FK analysis of a Stewart platform using NN is found in [86]. A cascaded CMAC (Cerebella Model Arithmetic Computer) based NN has been used and it is shown that this algorithm is faster and more precise compared to the popular back propagation algorithm. However, a modified feed-forward network [87] was later found to be more accurate which used a mapping offset adjustment method. A linear estimator [88] has been proposed for FK problem using Luemberger’s observer. The estimation gains are learned by NN employing back propagation algorithm. A novel loop method [89] has been proposed to further fine tune NN solutions and achieve higher accuracies. A hyper-chaotic neural network [90] has been proposed to generate local initial points during mathematical programming to find all the solutions of non-linear FK equations. FK solution of a 6-dof HEXA robot has been presented using NN in [91]. A comparison of NN with various numerical methods to solve FK has been provided by
[21]. Back Propagation Neural Network (BPNN) has been used by [92] where they have employed Particle Swarm Optimization (PSO) to train NN to achieve accuracy of the order of 0.001 radians. A floating point GA using IK analysis has been proposed by [87] to solve FK problem, formulating it as an optimization problem.

Although close form solution approaches and Analytical approaches were able to determine maximum number of solutions for a given parallel mechanism, a unique solution of FK problem could not be delivered using these approaches and hence they cannot be adopted in the real time control problem of wearable rehabilitation robots.

Numerical methods are accurate and can provide unique solution to the FK problem. However, these methods are not efficient in terms of computational time, since multiple iterations are required to achieve convergence. Selection of the initial estimation of robot task space configurations is also critical in allowing the convergence to accurate solutions [87]. While the latter problem can be somewhat solved by using previous results as the new starting point in the subsequent computation, the former issue of computational complexity can only be alleviated through efficient implementation of computation algorithms and more powerful computing hardware. Comprehensive review of various artificial intelligence based approaches reveals that the accuracy and the computation time capabilities of these algorithms also need further improvement. GA based approaches are time consuming [87] and lacks in the accuracy [93] whereas NN algorithms requires large database for training and lacks interpretability. Consequently, it is desired to develop a computational approach (to solve FK problem) which is accurate and fast enough to be used in online control applications.

2.4 Parallel Robot Design Analysis

While designing wearable robots based on parallel mechanisms, it is recommended that their design be optimized to reduce the adverse effects of the simultaneous and parallel action of its actuators [94]. Possible adverse effects are reduced workspace, increased singularity, higher actuator force requirements etc. It has been revealed from the literature that to exploit the comprehensive potential of the parallel robots, researchers in past have worked on its design optimization [95]. To address the issues of the workspace and singularity, trial and error or exhaustive search methods were used during previous research. Design optimization using exhaustive search minimization method has been used in [96] wherein a performance index called space utilization is proposed to evaluate the optimal kinematic design of a linear
Delta robot. Several non-dimensional indices, related to the link length are chosen as design parameters to optimize mobility, workspace and manipulability. Trial and error search methods normally work on rigorous experiments or simulation runs and intuitive judgments on the results thereafter. The main drawback of these approaches is that, with an increase in the number of tunable parameters, the required number of simulation runs increases exponentially. Moreover tuning of all the design parameters simultaneously is difficult and time consuming.

Researchers have also tried to perform the robot design optimization using numerical methods. Several performance indices such as manipulability, isotropy, dexterity index, conditioning index, global conditioning index (GCI) and global isotropy index have been defined by different researchers and details of these indices are provided in [97]. Though these indices have been defined in the perspective of serial robots, the same can be used to evaluate parallel robot designs. Geometrical optimization of a 2-dof planar parallel robot has been performed to maximize a global mechanical advantage matrix. An exhaustive search has been performed using Monte Carlo method for the optimization [98]. Design of a robotic arm exoskeleton developed on the basis of a parallel mechanism has been optimized to achieve high force using low mass of the exoskeleton [99]. Authors have used Matlab optimization toolbox to implement the linear programming optimization problem in discretely optimizing the range of motions, inertia and the force generating capacity of the robot. Workspace and motion/force transmissibility of a spherical parallel robot has been analysed and optimized using geometrical approach [100]. This robot is intended to be used as a minimally invasive surgical robot wherein the transmission index plays an important role and thus requires optimization. A new geometric method called performance atlas is introduced in [101] to optimize the transmission index and workspace of a 3-dof parallel robot. To minimum actuator forces in a cable based parallel robot, its design has been optimized in [102]. Authors have used Dykstra’s projection method to optimally distribute the actuator forces among the cables and the redundant limbs. Although the force distribution among links has successfully been optimized to provide minimum norm solution of the force vector, the geometrical parameters of the robot have not been altered to minimize actuator forces and improve the condition number and other design performance indices. A novel kinematic design method has been implemented in [17] and various performance indices such as global conditioning workspace, global conditioning index and global stiffness index have been used to obtain optimal link lengths for the subject robot.
Workspace and stiffness of modular parallel robots have been studied and their design is optimized in [94]. Modular robots can adapt their geometry to perform different tasks. The author has proposed a branch and prune type algorithm to optimize the robot design for a specific task by changing the actuator connection points on the base platform along a straight line while keeping the actuator connection points at the moving platform fixed. Architectural optimization of a 3-UPU parallel robot has been performed in [103] to maximize the global conditioning index.

Although, numerical methods have been persistently used in robot design optimization, there is always an apprehension that, in the absence of a near optimal initial estimation, the algorithm may converge into a local optimal solution. Numerical techniques also become less efficient when the solution search space is large and is finely discretized. It has been observed that when the objective function does not change over certain points in succession, numerical methods become less effective [104].

Recently, GA has also been used by researchers, to optimize the design of parallel robots [105, 106]. GA works with population of points and processes them simultaneously hence is more likely to provide a global solution. To maximize the workspace of a 2-dof parallel robot, a mono-objective function describing the reachable workspace has been formulated in reference [105]. Optimal robot link lengths have been obtained by maximizing the objective function using GA. Design optimization of a Stewart platform has been performed by Stan et al. in [107] wherein GA has been used with constraints defined as penalty functions to minimize the sum of the condition numbers calculated on a pre-defined trajectory. However, since the condition number has not been analyzed all over the workspace, the robot design may not be optimal for multi-tasking. To validate the algorithm, results obtained from GA have been compared with those obtained from Quasi-Newton method.

Most of the previous work in design optimization of parallel robots has treated design objectives either in isolation or by combining them into one objective. Optimization in either case has been performed using single objective optimization approaches. However, looking to the multi-objective formulation of the design optimization problem, wherein the objectives are conflicting and are incomparable, the single objective optimization approaches have limitations.

Multi-objective optimization approach has also been used in parallel robot design optimization. Preceding research work, related to the multi-objective optimization is being discussed in the following sub-section.
2.5 Multi-Objective Optimization

Significance of optimization in the realm of multi-objective problems differs from its usual context of maximization or minimization. Real world problems set forth challenges wherein a best compromised solution is normally accepted. For problems having multiple conflicting goals which are incomparable, the optimal solution is normally derived in the form of a non-singular set of equitable solutions [13]. Owing to the varied perception of human end user, it is desired to provide a wide set of equitable solutions obtained from different grades and blends of trade-offs established between objectives [108]. The process of obtaining such set of solutions is cumbersome using classical optimization methods. Consequently in the past two decades, evolutionary algorithms (EA) have emerged as a plausible alternative to the classical approaches, where set of equitable solutions can be obtained in a single simulation run [109]. Contrary to the classical approaches, EA works with population and its inherent mechanism of evolution, emulating the natural evolution, facilitates simultaneous exploration of various trade-off solutions with different grades and blends. EA does not require derivatives of objective functions and has robust operators such as reproduction and regeneration to avoid convergence to local optima. Applications ranging from engineering design, groundwater monitoring, and autonomous vehicle navigation to polymer extrusion and city planning have been benefited significantly by use of EA [110]. The representative work citing applications of EA in the parallel robot design optimization problem is discussed below.

Stan et al. [106, 111] have proposed a multi-objective design optimization using GA for a 3-dof and 2-dof parallel robot. In their contributions, the authors have emphasized the importance of a global design solution and hence justified the use of GA against the numerical methods. The crossover and the mutation operators have been performed with 0.08 and 0.005 probability. Though there is no explanation given for the selection of these numbers, it is difficult to comprehend as why such low crossover probability has been used. Moreover, authors have not verified whether the design solution obtained from GA is a global solution. Large variation (0.1 to 0.8) of the Transmission Quality Index (which is similar to the condition index of Jacobian matrix) in the workspace also indicates the possibility of further design improvement. Strength pareto evolutionary algorithm (SPEA) approach has been used in [112], employing a modified GA to optimize the design of parallel robots. Performance criteria such as workspace, transmission behaviour, stiffness and rigidity have been simultaneously optimized.
2.6 System Modelling and Control of PMA

The research work, discussed above, do not consider all the three design aspects of parallel robots namely, kinematic, actuation and structural aspects, concurrently [106, 113]. Therefore the previous design optimization work cannot be taken as complete in all respects. To establish a perfect trade-off between design objectives, it is important to consider all the design aspects simultaneously during optimization.

2.6 System Modelling and Control of PMA

PMA have been commonly used in developing wearable robots due to their effectiveness in achieving of wearability, compactness, light weight and portability [12]. However despite many advantages, it has been observed that PMA have not been extensively used in the past. This is due to the use of latex in their construction which exhibit highly non-linear dynamic characteristics. The Latex bladder used in PMA has significant hysteresis and hence the actuator show different characteristics during inflating and deflating. This hysteresis is more pronounced at higher frequency actuation. Moreover the dynamic characteristics of PMA are also affected from the temperature variation arising out of friction while in operation. Thus the actuator behaviour changes with time. The driving force in the PMA is obtained from pressurized air/gas, and the inherent compliance of air/gas is difficult to comprehend which also poses control related problems [114].

These drawbacks can be effectively addressed if an accurate dynamic model of PMA is constructed. Several attempts have been made in the past to model the dynamics of these muscles and relevant control schemes have been proposed [25, 33]. Before selecting an approach to model PMA behaviour, various models proposed in the literature are studied. These modelling approaches can be categorized as analytical modelling, numerical modelling and artificial intelligence based modelling.

2.6.1 Analytical modelling

The axial length of a pneumatic muscle is shortened when it is inflated. The muscle exerts a force on its free end while contracting which can be used to perform useful work. Some portion of the input work also gets stored in the rubber bladder in the form of the changed strain energy density ($\Delta W$). Thus for a small change in PMA length, the input work done can be accounted as below [115]:

$$P\Delta V = F\Delta L + V_e \Delta W$$  \hspace{1cm} (2.2)
Here initial volume and the inflation pressure of the muscle are \( V_e \) and \( P \) respectively. Change in muscle volume is indicated by \( \Delta V \) whereas force and axial displacements are shown as \( F \) and \( \Delta L \). Rearranging and writing above equation in the differential form following can be derived:

\[
F = P \frac{dV}{dL} - V_e \frac{dW}{dL}
\] (2.3)

A linearized model relating extension and force of PMA has been derived at constant pressure by Chow et al. [116]. They have also provided a useful comparison of PMA with skeletal muscles to evaluate the use of PMA in biologically inspired robots. Subsequently, Tsagarakis and Caldwell [117] improved upon this model by introducing an end correction term to the actual force considering the conical shape of the actuator at the ends. Similarly, Tondu and Lopez [118] also suggested inclusion of a parameter ‘k’ (k\( \leq \)1) which amplifies the contraction ratio and provides end correction. Colbrunn et al. [115] proposed a model for actuator stiffness and also proposed a correction factor called effectiveness (a function of pressure) to account for hysteresis in the muscles.

Problems of large modelling errors and associated low operating bandwidth of PMA have been pointed out by [119] suggesting modifications in the pneumatic muscle design. A three element model (mass-spring-damper) describing the dynamic characteristic of PMA has been proposed by Reynolds et al. [120]. Stiffness of PMA, which is a function of muscle pressure and length, has been modelled in [121]. In another interesting research, increase in load carrying capabilities of PMA has been reported by [122] when water is used as the working medium in place of air or gas. The changed strain energy density was considered for the first time, by [123] and non-linear Mooney-Rivlin materials model was used to account the stored elastic energy. Similar work has been proposed by [124] wherein large modelling errors have been reported when the modelling results are compared with experimental outcomes.

Quantifiable amount of work has been done to analytically model the PMA; however, these models do not predict the time dependent behaviour of PMA. Low prediction accuracy in modelling the non-linear behaviour is also of serious concern. The main reason for this inaccuracy is the lack of knowledge for the PMA behaviour owing to its conical ends, friction between the inner tube and outer sheath and the use of latex inner tube. Modelling of valves and fluid flow characteristics is also difficult to obtain and the large hysteresis present in these actuators poses a challenge in the construction of an accurate dynamic model.
Consequently, several numerical models, discussed below, are proposed by researchers to enhance accuracy.

2.6.2 Numerical modelling

In order to use finite element method to model PMA, Bertetto and Ruggiu [123] have discretized its latex tube by a mesh of hyper elastic incompressible elements with eight nodes having two degrees of freedom each. Mooney-Rivlin’s energy function with two model parameters has been considered. A position control scheme of PMA has been proposed by Schindele and Aschemann [125] wherein the non-linear force characteristics is approximated into a polynomial using static measurements. A numerical technique called Non-linear Model Prediction Control (NMPC) has been used to minimize a tracking error between predicted and desired states at each time step (defined as prediction horizon \( T_p \)). As this work has been done for position control, change in the force on the actuators is not considered.

2.6.3 Artificial intelligence modelling

A number of artificial intelligence techniques such as Artificial Neural Network (ANN), Fuzzy Inference, GA and their variants can effectively model, systems showing non-linearity and uncertainties. From the discussion in the preceding sections it is evident that the conventional tools cannot fully comprehend the non-linear and time dependent PMA characteristics. Therefore these actuators are good candidates for the application of AI techniques and considerable work has been done in this direction. Recurrent neuro-fuzzy model has been proposed by Chang and Lilly [126]. A comprehensive Neuro-Fuzzy modelling of PMA has been done by Ahn and Ahn [25, 127, 128]. A low order linear approximation of PMA has been done using a ARX (Auto Regressive with Exogenous Input) model [127]. The parametric values of the ARX model have been optimized using modified GA (MGA). A recurrent NN structure has been used with ARX model [128]. Recently, a MGA based non-linear ARX (NARX) fuzzy model [25] has been proposed to model PMA behaviour. Superiority of MGA based NARX fuzzy model has been established and MSE from this model (when compared with empirical outputs) was found to be 10%.

It can be asserted from the above discussion that there are at least two vital issues that need to be properly addressed. Firstly, majority of the research on PMA modelling has been done for constant loading or in other words, PMA has not been subjected to varying loads. It is shown later in the Chapter 8, that the pressure-length characteristics of PMA greatly depend on the applied loads which cannot be taken as invariable. Secondly, the prediction
accuracy from the previous models also needs to be improved in view of the requirements of precise kinematic conformity of wearable robots.

2.6.4 Pneumatic muscle actuator control

The previous research work carried out in the field of PMA control can be divided broadly into two classes. First set of control approaches is the one which made use of a linear control method in conjunction with higher level of heuristics such as adaptive or fuzzy logic control. Variable structure control in [118] and generalized variable structure model reference adaptive controller proposed by Nouri et al. in [129] are the representative work done in this area. Other approaches such as the adaptive pole-placement method by [130], the fuzzy controller with PD+I approach [131], gain scheduling approach by [132] fall in this set of control methods.

Another set of control methods comprises of nonlinear control methods such as a second order three-element actuator model used in [133-135] to develop a robust back-stepping controller, an adaptive back-stepping controller and a sliding mode controller respectively. The cascaded sliding mode controller proposed by [136] and the sliding mode control approach of Cai and Dai [137] are typical examples of nonlinear control approaches.

Another interesting work to develop a rehabilitation robot for upper limb therapy has been introduced in [138] wherein pressure has been considered as a control variable to implement a position based impedance control strategy in a static mode.

All of the above mentioned work has been done either for serial robots or an antagonistic arrangement of PMA. The only application wherein a parallel robot using PMA has been controlled is proposed by Zhu et al. [139]. An adaptive robust controller has been implemented which includes models of PMA, pressure valves and robot dynamics. The robot design, however, has serious flaws; with all the actuators being parallel it appears to be a singular design. The authors have claimed to have achieved 3-dof from this robot which is difficult to understand since only three PMA are used. When using flexible actuators such as PMA which cannot produce a compression force, (n+1) actuators are required to achieve n-dof [140]. The results shown in [139] also do not show errors about the vertical axis of the robot, and PMA have not been modelled for the external force.
2.7 Discussion

The existing rehabilitation robots proposed for the treatment of ankle injuries and related motor disorders were surveyed and their designs were analyzed. A gradual evolution in the previous designs was observed which showed that there have been efforts to reduce the redundancy in the robot designs. However, it was found that the platform type designs are not kinematically compatible with the biological structure of the ankle joint. Moreover, previous robot designs are largely inspired from the industrial robots and hence cannot serve as wearable robots.

The related research work on FK was reviewed and classified among four categories such as, closed form solutions, analytical approaches, numerical approaches and artificial intelligence (AI) based approaches. It was found that a unique solution of FK can only be obtained using numerical and AI approaches. Numerical approaches are time consuming and do not guarantee a correct solution since they require a good starting solution. AI approaches, such as neural network, on the other hand have inherent problem of poor interpretability and require a large database for training to identify system parameters. Therefore, there is a need to develop a computational model, to provide FK solutions, which is computationally more efficient and accurate.

The wearable robot design is expected to be compact, compliant, lightweight and at the same time it is required to have large workspace (commensurate to the ankle rotations), stiff and powerful. In view of the fact that these requirements were conflicting, it is necessary to carry out optimization of the robot design to establish a trade-off between the design objectives. Consequent upon a brief review of the available literature, it was found that the design optimization process normally has three vital components such as, problem formulation, selection of the key design parameters and the optimization methodology. It was realized that there exist opportunities of improvement in all these areas. So far, the design optimization of the parallel robots had been defined separately, and the performance indices, such as condition number, singularity and actuator forces, stiffness etc., had been optimized in isolation. Therefore it is required that the optimization problem be formulated by considering all the important performance criteria so that they are simultaneously optimized. Further, during earlier design optimizations, all the key design parameters of the robot (such as locations of the actuator connection points and distance between platforms) were not altered collectively which is another research opportunity for the present work. In previously published research work, while selecting an optimization scheme for multi-objective
optimization, NSGA II, a variant of evolutionary algorithms (EA) was preferred [141] over other numerical methods. NSGA II was chosen due to the fact that it could provide a set of Pareto optimal solutions which were globally optimal. It was revealed from the study of the literature that while the variants of EA have been successfully used in last two decades to solve multi-objective optimization problems (MOP) in variety of applications, there are certain issues which need close attention. Evolutionary algorithms use the concept of non-dominance ranking while selecting better solutions. However, while dealing with large number of objectives in a continuous solution space the concept of non-dominance loses its significance [109]. As a result more and more solutions become non-dominant and a distinct Pareto front cannot be obtained. Consequently, during selection, since most of the solutions carry the same non-dominant rank, a crowding distance operator (which is a time consuming algorithm) is called in quite often to help in breaking a tie between solutions [109]. Apparently, EA becomes inefficient to solve higher dimension MOP [13, 142]. A number of other concerns regarding EA will be further mentioned in the thesis while discussing the design optimization of the wearable robot. Thus it is proposed in the present work that for applications involving higher dimension MOPs the conventional selection approaches of EA need to be revisited and alternatives be explored.

Design of a suitable controller to simultaneously control the lengths of four actuators to achieve a desired orientation at the end-effector is a difficult task. The non-linear and time dependent behaviour of the actuators further adds intricacies to the control difficulties. This calls for an accurate dynamic model of the PMA. Although, the static and dynamic modelling of the PMA had been widely reported in the literature, it was noted that the accuracy of the predictions in the presence of large hysteresis and the external force were still the unanswered concerns.

2.8 Chapter Summary

This Chapter presented a comprehensive review on the research issues which were relevant to the development of wearable robots for rehabilitation. The focus of the review was placed on the following aspects including, contemporary ankle rehabilitation robot designs; FK solutions of parallel robots; design analysis of parallel robots; multi-objective optimization and PMA modelling and their control.

Research opportunities were explored while developing the intended wearable robot and it was found that above mentioned research aspects needed further investigation. While a new
optimal design was crucial for a wearable robot, fast and accurate FK solution, design optimization, actuator modelling and controls were equally important aspects for a wearable rehabilitation robot. The Chapter concluded with a discussion on the earlier work to draw motivations for the present research.
Chapter 3 Wearable Ankle Robot: Development and System Modelling

Contemporary ankle rehabilitation robots lack kinematic compliance which causes discomfort to patients and difficulties in controls. Owing to their rigid and heavy structures, earlier robots are not portable and ambulatory. Actuators used in these robots are bulky, non-compliant, stiff and non-backdrivable. Therefore these robot designs are not suitable to be used as rehabilitation robots and interact closely with human actor. Higher costs and safety apprehensions are some of the concerns, limiting their common use.

This Chapter presents an overview of the requirements from a wearable ankle robot and discusses design concepts and construction details. A brief description of the ankle joint anatomy is provided to create acquaintance and understanding of the ankle motions. Study of the ankle joint anatomy, possible ankle disorders and contemporary rehabilitation practices is also important while investigating robot design specifications and rehabilitation trajectories and strategies. A discussion on design specifications is provided, followed by a Section on the conceptual design and development of the wearable ankle robot. Details of robot hardware and user interface are also provided to complete the discussion. Symbolic kinematic modelling of the feasible design is performed and difficulty in obtaining FK solution of the wearable robot is explained. Geometrical and dynamic modelling of the wearable robot are also carried out in this Chapter.

3.1 Human Ankle, Potential Disorders and Physiotherapy

3.1.1 Ankle Complex

Human ankle joint is a very complex bony structure in the human skeleton [143] and is fundamentally a combination of two joints (Figure 3.1). The first joint is called the ankle joint which is made up of three bones: the lower end of the tibia (shinbone), the fibula (the small bone of the lower leg) and the talus (the bone that fits into the socket formed by the tibia and the fibula). The talus sits on top of the calcaneus (the heel bone) and moves mainly in one direction. The ankle joint works like a hinge, to allow up, (dorsiflexion) and down (plantar flexion) foot motions. The second joint is the subtalar joint, also known as the talocalcaneal
3.1 Human Ankle, Potential Disorders and Physiotherapy

A joint which is a joint of the foot. It occurs at the meeting point of the talus and the calcaneus. This joint is responsible for the inversion and eversion of the foot, but plays no role in dorsiflexion or plantarflexion motion. However it is very much a part of the ankle joint and thus is important.

![Figure 3.1: (a) The Ankle joint anatomy and (b) ankle motion trajectories for right foot (±θ inversion and eversion, ±ψ plantarflexion and dorsiflexion, ±φ abduction and adduction).](image)

There is one more joint (normally not considered as part of the ankle joint) called MTP (metatarsophalangeal) joint connecting fore and the rear with Calcaneus, Cuboid and Navicular bones as shown in Figure 3.1a. The raising and lowering motions of the toes and the heel are achieved about this joint. In the present study the ankle and the subtalar joints have been collectively considered as one joint providing three rotational degrees of freedom and are called ankle joint henceforth for simplicity. Since this study is limited to the ankle joint motions and not the fore foot motions, the MTP joint and related motions are not considered. The ankle joint can have rotations in all three planes namely, sagittal, frontal and transverse planes. The sagittal plane is contained by the x and the z-axes and ankle movements in this plane occur about the y-axis (Figure 3.1b). Ankle motions in the sagittal plane are termed as plantarflexion and dorsiflexion as shown in Figure 3.2. The transverse plane is defined by the x and y axes and movements in this plane occur about the z-axis. Motions in the transverse plane are defined as adduction (when the right foot toes are moved towards the left foot) and abduction (when the right foot toes are moved away from the left foot). The frontal plane is formed by the y and the z-axes and the ankle motions in this plane occur about the x-axis. Frontal plane motions of the ankle joint are termed as inversion (when the inner side of the foot also known as medial side is lifted up) and eversion (when the inner side of the foot is pushed down) (Figure 3.2). Various ankle movements suggested by Siegler
et al. [144] and maximum passive moment requirements [145-147] are summarized in Figure 3.2 and Table 3.1.

**Table 3.1: Range of motion and moment requirement at the human ankle**

<table>
<thead>
<tr>
<th>Ankle motions</th>
<th>Range of motion</th>
<th>Maximum passive moment (Nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+θ Inversion</td>
<td>14.5° to 22°</td>
<td>33.1 ± 16.5</td>
</tr>
<tr>
<td>-θ Eversion</td>
<td>10° to 17°</td>
<td>40.1 ± 9.2</td>
</tr>
<tr>
<td>+ψ Dorsiflexion</td>
<td>20.3° to 29.8°</td>
<td>34.1 ± 14.5</td>
</tr>
<tr>
<td>-ψ Plantarflexion</td>
<td>37.6° to 45.75°</td>
<td>48.1 ± 12.2</td>
</tr>
<tr>
<td>+φ Adduction</td>
<td>22° to 36°</td>
<td>NA</td>
</tr>
<tr>
<td>-φ Abduction</td>
<td>15.4° to 25.9°</td>
<td>NA</td>
</tr>
</tbody>
</table>

NA: Not Available

Information on maximum torque requirement for adduction/abduction motions is not available from the literature. However in the present research, these values are assumed to be equivalent to the inversion-aversion torques.

### 3.1.2 Ankle Injuries and Physiotherapy

Ankle injuries [143] are one of the most common injuries in sports and daily life. For example, youngsters are subjected to ankle injuries from sports and whilst carrying excessive load, and children and the elderly are injured from walking on uneven surfaces and bone weakness. Non-functionality of ankle joint is also quite common for stroke surviving patients.

Common ankle injuries are sprain, strain and fracture. An overstretched muscle or tendon can often cause strain which is a mild injury. However if a ligament is overstretched it causes
more serious injury called sprain which results in pain and joint non-functionality. Sometimes when a ligament is overstretched and broken it may pull off a piece of bone causing a fracture [148].

Ankle joint is predominately subjected to the sprain injuries resulting from overstretched ligaments. International studies report that [149] ankle sprains contribute to 15-20% of all sports related injuries. Referring to the study conducted by the Effective Practice Institute, University of Auckland, ankle sprains cause a significant cost to ACC (Accident Compensation Corporation), New Zealand [149]. In the year 2000/01, apart from the 17,200 ongoing claims, 82,000 new claims, amounting over $19 million were received by ACC. Expenses on part of new and ongoing ankle claims, ranked fourth largest cost to ACC after the lower back, neck and shoulder injuries. During a study conducted in the United States for epidemiology of ankle sprain, it was found that 2.15 incidents of ankle sprain per 1000 persons, were recorded during 2002-2006 [150].

Primary treatment for ankle injuries [5] includes, rest, ice, compression and elevation (RICE) of the affected foot-ankle entity. Application of ice after rest is used to reduce swelling, compression stockings are used to firmly support the foot-ankle body and elevation helps to minimize further swelling. The primary treatment should be followed by stretching and motion therapy along with partial weight bearing to maintain mobility in the ankle joint. Motion therapy is recommended to start within 72 hrs of the injury, to prevent muscular atrophy which may lead to a reduced range of motion (ROM). Motion therapy also stimulates healing of the impaired ligaments [151]. Once the ROM is achieved, strengthening of weakened muscles is essential for rapid recovery and is a preventive measure against further injury. As patients achieve full weight bearing capability without pain, proprioceptive exercise is initiated for the recovery of balance and postural control using wobble boards. Finally, advanced exercises using uneven surface wobble board should be performed to regain functions specific to normal activities.

The ankle joint is an important joint in human skeleton since it is responsible to carry the body weight and maintain balance during gait. It is subjected to high impact forces which may be as high as several times of the body weight (e.g. while jumping). It is a very strong joint with stiffer muscles and hence offers large moments as shown in Table 3.1. In light of the above facts, it can be concluded that apart from the ROM capabilities, wearable robotic device for ankle rehabilitation should have higher payload capacity to provide required passive and resistive moments at the ankle joint.
3.4 Symbolic Kinematics Modelling

Several ankle rehabilitation robots have been proposed in the literature [5, 16-20]. Most of the previous robot designs are based on parallel mechanisms due to higher stiffness and large force requirements for the ankle joint treatment. All the previous designs are relatively similar and thus have common drawbacks as discussed earlier.

3.2 Design Specifications

To develop a cost-effective, safe, compliant and wearable solution for an ankle rehabilitation robot, it is important to study the requirements and specifications which formed the basis of the proposed ankle robot design.

To begin with, robot’s links and joints should be designed to provide sufficient workspace for the end-effector motions which are required to implement basic range of motion treatments of the ankle joint. Ankle joint is very stiff joint, a property which helps it to bear large impact forces which may be several times of the body weight. Large active moments of the order of 100 Nm can be applied by the ankle joint [152] therefore the ankle robot construction should have sufficiently high stiffness to avoid possible deformations and associated positional errors. The robot design should be right-left symmetric so that left and right foot-ankle joints can be treated using the same robot. Moreover since subjects of different ethnicity, physical appearances and age groups may possibly use this robot, its design features should be adaptable. Safety is a major concern and it should be considered at every stage of robot development.

There are certain requirements which come from the wearable aspect of the robot. These are portability, compact size, light weight, friendly appearance and ease to put on and remove etc. The robot is expected to have low impedance to allow backdrivability which will further help the robot in minimizing trajectory errors during assistive motions.

With intentions to develop a pragmatic and feasible design for the ankle robot, meetings were arranged with physician in the School of population health at The University of Auckland, Auckland. Regular meetings with practicing therapists were important for a practical design solution

3.3 Conceptual Design and Construction

Analyzing through the above specifications, it is realized that higher stiffness requirements associated with relatively small orientation workspace of the ankle joint can be addressed using a parallel kinematic structure for the ankle robot. Subsequently various
concepts are perceived and analyzed for their feasibility in the present application. A design which is finally chosen is shown in Figure 3.3.

This robot design is able to provide three rotational degrees of freedom to the ankle joint for the necessary range of motion (ROM) and muscle strengthening exercises [47]. It employs two parallel platforms; a ‘U’ shaped FP built-in with a leg support structure and a moving platform (MP) or end-effector designed to accommodate the foot and the ankle of patients. Pneumatic muscle actuators [138] are considered over electrical motors owing to their light weight and skeletal muscle like behaviour. Electrical motors for the present application would be very heavy (one motor would weigh more than 3 Kg) and the robot will not be a wearable one. Excess weight of actuators shall also cause significant undesired slip between human limbs and the robot. Pneumatic muscle actuators are mounted on the leg support with their actuating ends connected to the end-effector through flexible steel cables. Use of flexible cables along with PMA helps in achieving kinematic compatibility. Basically the PMA have a simple construction wherein a rubber tube is covered with braided plastic mesh shell. The inner rubber tube shortens approximately 30% in length when inflated with compressed air at low pressure (2 to 4 bars).

Pneumatic muscle actuators are chosen due to their very light weight and compliant action. These actuators can deliver power-to-weight ratio as high as 400:1 compared to the pneumatic cylinders and DC motors which can deliver only 16 times of their weight. The actuation of PMA, in general, can be characterised equivalent to the human skeletal muscles. The actuators used in the proposed prototype are 30 cm in length, weigh only 80 gm, and can exert more than 600 Newton pull force when operated at 4 bar pressure. A pressurized air supply is given to the four PMA through a special pneumatic control unit for their sequential/simultaneous actuation.

It is known that robots using cables along with actuators will lose controllability if during operations one of the cables becomes slack or is not in tension [153]. Thus, it is important to ensure that all the cables remain under tension during robot motions. Further since the PMA can only pull and cannot push, to maintain the tension in all the cables during operations, it is important to have redundant actuation. In fact redundant actuation is a prerequisite for all the cable based parallel robots which means that ‘(n+1)’ actuators are required to achieve ‘n’ degrees of freedom (dof) motion of the manipulator [153]. Therefore, in the above prototype, four sets of PMA in series with cables are used to obtain 3-dof from the robot. Coordinated and antagonistic actuations of PMA will ensure the desired changes in the cable lengths to achieve a required pose of the moving platform for a range of ankle exercises.
3.4 Symbolic Kinematic Modelling

Design capability analysis

The final design of the ankle robot, selected from many concepts, was analyzed for two important requirements i.e., range of motion and moment exerting capacities. Careful measurements were carried out from the readings of position and force sensors and results obtained from the above analysis were analyzed. It was observed that the ankle robot design is able to provide the estimated range of motion, with a maximum plantarflexion of 46°, maximum dorsiflexion of 35° and maximum inversion-eversion of 25°. The abduction and adduction motion available from the robot remains at about 70°.

Moment exerting capacity of the final design was also evaluated by considering the maximum forces available from the actuators. First of all, the maximum moments that can be applied at all end effector orientations were found while the robot was moved to trace the entire workspace. It is important to note here that this moment analysis was carried out on each of the X, Y and Z directions by using a maximum actuator force output of 700N. The maximum achievable end effector orientations and the moment capacity of the final design are summarised in Table 3.2. It can be seen that following movements and moments achievable by the ankle robot are similar to what is required for the X, Y and Z directions.
3.4 Symbolic Kinematic Modelling

(Table 3.1), which shows the capability of the proposed robot to perform the required course of rehabilitation treatment.

Table 3.2: Typical range of motion at the human ankle and corresponding robot capabilities

<table>
<thead>
<tr>
<th>Type of motion</th>
<th>Robot capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maximum motion</td>
</tr>
<tr>
<td>Dorsiflexion</td>
<td>35°</td>
</tr>
<tr>
<td>Plantarfexion</td>
<td>46°</td>
</tr>
<tr>
<td>Inversion</td>
<td>26°</td>
</tr>
<tr>
<td>Eversion</td>
<td>26°</td>
</tr>
<tr>
<td>Internal rotation</td>
<td>52°</td>
</tr>
<tr>
<td>External rotation</td>
<td>52°</td>
</tr>
</tbody>
</table>

3.4 Symbolic Kinematics Modelling

Spatial motion of the robot end-effector about a fixed reference frame can be described with the help of a kinematic model. During kinematic modelling, two types of analyses, namely, Inverse Kinematics (IK) and Forward Kinematics (FK) are carried out to study the motion of robotic manipulator in time domain. The desired trajectories of the end-effector are devised from the prescribed rehabilitation treatment and specified in terms of end-effector’s position and orientation in the workspace.

The joint/link variables to accomplish this task are found using IK analysis. Using FK analysis, sensor based measurements of these joint variables are used to compute the instantaneous position and orientation of the end-effector to find the error in the desired and actual displacements. Therefore, both types of kinematic models are normally required to study end-effector differential motion, its statics and to implement a control scheme for the end-effector. The kinematic analysis is also required to estimate the feasible workspace of the robot and to perform singularity analysis. Kinematic modelling of the proposed robot is briefly discussed in the following section.

3.4.1 Inverse Kinematics

The inverse kinematics of the proposed wearable ankle robot is relatively simple and provides a unique solution of cable lengths for given end-effector pose. In the following discussion the wire lengths have been determined in terms of the pose of the moving platform.
3.4 Symbolic Kinematic Modelling

The diagram presented in Figure 3.4 shows the position vectors of the cables in the proposed wearable ankle robot. The connection points on end-effector and FP are denoted by $a_i$’s and $b_i$’s respectively. The connection points on the FP are all in the same plane ($Z_o = 0$) and are placed at a radial distance ‘$r^b$’ from the coordinate system which is located at O. The position vectors ($b_i^o$) of point $b_i$’s on the FP are defined by (3.1).

![Figure 3.4: Line sketch of an actuator and its position vectors on FP and MP](image)

Here, $r^b_i$ is the radial distance of the fixed connection point from the fixed coordinate frame. The angular position of the connection point $b_i$ on FP is denoted by $\beta_i$. Similarly, connection points on the end-effector are located at ($r^a_i$, $\alpha_i$) in the polar coordinate frame attached to $O_e$. The position vectors ($a_i^e$) of the four connection points on the end-effector in global coordinates can be given as follows:

$$a_i^e = \begin{bmatrix} r^a_i \cos \alpha_i \\ r^a_i \sin \alpha_i \\ -120 \end{bmatrix}$$

The variable $\alpha_i$ is the angular position of point $a_i$ on the end-effector with respect to their respective axes and end-effector is located 120 mm below the FP. The position vectors of the actuator lengths in terms of end poses can be expressed as a system of four equations described below:
\[ L_i^o = P_e^o + R_e^o a_i^e - b_i^o \quad i = 1, \ldots, A. \] (3.3)

Here \( P_e^o \) represents the position vector of point \( O_e \) with respect to \( O \). \( R_e^o \) is the rotational transformation matrix of end-effector with respect to \( FP \) using a fixed axis rotation sequence of \( \theta_1, \theta_2, \) and \( \theta_3 \) about \( X_o, Y_o \) and \( Z_o \) axes, respectively, and can be written as below.

\[
R_e^o = \begin{bmatrix}
C\theta_3 C\theta_2 & -S\theta_3 C\theta_1 + C\theta_3 S\theta_2 S\theta_1 & S\theta_3 S\theta_1 + C\theta_3 S\theta_2 C\theta_1 \\
S\theta_3 C\theta_2 & C\theta_2 C\theta_1 + S\theta_3 S\theta_2 S\theta_1 & C\theta_2 S\theta_1 - S\theta_3 S\theta_2 C\theta_1 \\
-S\theta_2 & S\theta_2 C\theta_1 & C\theta_2 C\theta_1
\end{bmatrix} \quad (3.4)
\]

Here \( S \theta = \sin \theta \) and \( C \theta = \cos \theta \)

After the rotational matrix is calculated for given orientation of the end-effector, actuator displacements or joint actuation can be obtained by finding the Euclidean norm of the actuator length vector in (3.3).

### 3.4.2 Forward Kinematics

In a similar manner FK solution can also be obtained using (3.3), wherein Euler angles shall be required to be determined for a given set of link lengths. To solve FK, system of equations represented by (3.3) is solved for three unknowns with four available equations which are highly coupled. Apparently (3.3) shall provide more than one solution of Euler angles, making this difficult to select the correct solution. This is a very interesting problem and quite a few studies have already been done by previous researchers. During present research a new computational model shall be proposed to enhance accuracy and reduce the computational complexity of the FK solutions. Due to distinct and large extent of the subject, a detailed description of this model shall be provided in the next Chapter.

### 3.5 Geometric Modelling

The kinematic models establish the correlation between the joint displacements and the position and orientation of end-effector of the robot. This correlation can only be used for the static control of manipulator in the workspace. For the proposed robot the final desired angular pose of the manipulator is important and at the same time the angular velocity by which it has traversed to reach to the final location is also equally important. Thus, it is essential to obtain a mapping between joint velocities and end-effector velocity. This mapping can be defined by a matrix, which is called the robot Jacobian matrix. The Jacobian matrix depends on robot configuration and linearly maps the Cartesian velocity in to joint velocities. It is interesting to note that the Jacobian matrix defined for the parallel robots
corresponds to the inverse Jacobian of the serial robots. To determine the Jacobian matrix of the parallel robots two approaches, namely geometric approach and analytical approach can be used. The geometric approach, which has been used in the present work, is discussed below.

To determine geometric Jacobian matrix using robot geometry, initially a relation between cable lengths and end-effector pose is formulated. For the subject robot this relation is given by (3.3). Subsequently, the magnitude of the cable lengths can be calculated for a given set of end-effector orientation. The magnitude of each $L_i^o$ vector can be given by $l_i^o$ as shown below.

$$(l_i^o)^2 = (L_i^o)^T L_i^o \quad (3.5)$$

Taking the time derivative of the above kinematic constraint equations and using (3.3) following equation can be obtained.

$$2l_i^o \dot{l_i^o} = (\dot{P}_e^o + \dot{R}_e^o a_t^e)^T (P_e^o + R_e^o a_t^e - b_i^o) + (P_e^o + R_e^o a_t^e - b_i^o)^T (\dot{P}_e^o + \dot{R}_e^o a_t^e) \quad (3.6)$$

Using linear algebra identity, if $x$ and $y$ are column vectors, following holds true,

$$x^T y + y^T x = 2x^T y = 2xy^T \quad (3.7)$$

This further implies that the (3.6) can be rewritten as below.

$$2l_i^o \dot{l_i^o} = 2(P_e^o + R_e^o a_t^e - b_i^o)^T (\dot{P}_e^o + \dot{R}_e^o a_t^e) \quad (3.8)$$

Since the end-effector is constrained to only rotational motion and there is no translation motion between the platforms, the time derivative of $P_e^o$ should be zero. Setting $\dot{P}_e^o = 0$ and further simplifying (3.8), one gets,

$$l_i^o \dot{l_i^o} = (L_i^o)^T (\dot{R}_e^o a_t^e) \quad (3.9)$$

but since

$$\dot{R}_e^o = \omega_e^o \times R_e^o \quad (3.10)$$

$$l_i^o \dot{l_i^o} = (L_i^o)^T (\omega_e^o \times R_e^o a_t^e) \quad (3.11)$$

Here $\omega_e^o$ is the angular velocity vector of the end platform with respect to the coordinate system of the fixed platform. Further since $a_t^e$ and $a_t^o$ are related as shown in (3.11), (3.12) can be rewritten as (3.13). Rearranging the variables, (3.13) can be presented as (3.14 &3.15).
If \( \dot{q} \) is the vector of link velocities and \( t \) is the twist vector of the end platform, the Jacobian matrix \( J(q) \) of the robot can be defined as,

\[
\dot{q} = J(q) t
\]  \hspace{1cm} (3.16)

Here \( t \) is a vector of angular velocities of the end platform and is given by following.

\[
t = \begin{bmatrix} 0 \\ \omega_c^o \end{bmatrix}
\]  \hspace{1cm} (3.17)

Now (3.15) can be rearranged and compared with following matrix equation (3.18) to find the Jacobian matrix as given by ( & ).

\[
X \dot{q} = Y t
\]  \hspace{1cm} (3.18)

\[
J = X^{-1} \times Y
\]  \hspace{1cm} (3.19)

Finally the Jacobian matrix of the proposed cable based robot can be written such that its \( i^{th} \) row is given by following equation.

\[
J_i = \left[a_i^o \times \frac{L_i^o}{l_i^o}\right]^T
\]  \hspace{1cm} Where \( i = 1, \ldots, 4 \).  \hspace{1cm} (3.20)

The robot Jacobian matrix is an important parameter and is extensively used for the kinematics and dynamic analysis of the robot. This matrix shall be further used in the workspace analysis and in the actuator force analysis in Chapter 5.

### 3.6 Dynamic Modelling

Dynamic modeling of the robot is essential while designing a controller for the robot. It is also a vital component for the study of the structural design parameters of the wearable robot such as, stiffness and Eigen values. Presently, to evaluate the potential of the wearable robot design in the ankle joint rehabilitation, only robot system is modeled and the environment, which is the patient, is not considered at this stage. The end-effector and the links are initially modeled separately for convenience and later combined to form one model using robot’s
3.6 Dynamic Modelling

Jacobian matrix. The dynamic model of parallel robots due to the coupled motion of their links is difficult compared to the serial robots. Nevertheless, the final dynamic model is still similar to the standard dynamic model of serial robots, which can be written as below:

\[ M \ddot{\Theta} + C(\dot{\Theta}, \Theta) + G \Theta = J^{-T}F + M_{ext} \]  

(3.21)

Here \( M \) is the inertia matrix, \( C \) is the term used for Coriolis and centripetal forces, \( G \) is the gravitational term whereas \( F \) and \( M_{ext} \) are actuator force vector and vector of external moments applied to the robot’s end-effector. Friction is not considered at this point for simplicity. Owing to complexities of the mechanical system of robot, Lagrange approach is chosen for the dynamic modeling. Lagrange’s equation for the end-effector can be given by following (3.22).

\[
\frac{d}{dt} \left( \frac{\partial L_{MP}}{\partial \dot{\theta}} \right) - \frac{\partial L_{MP}}{\partial \theta} = \sum_{i=1}^{n} \left( R_{xi} \frac{\partial x_i}{\partial \theta} + R_{yi} \frac{\partial y_i}{\partial \theta} + R_{zi} \frac{\partial z_i}{\partial \theta} \right) \]

(3.22)

Lagrangian of the end-effector is defined as \( L_{MP} \) and \( R_{xi}, R_{yi} \) and \( R_{zi} \) are the magnitudes of the Cartesian components of the reaction force while \( n \) is the total number of links between the two platforms. Further Lagrange’s equation for the links or the actuators (considering them as pure source of force) can also be expressed as below:

\[
\frac{d}{dt} \left( \frac{\partial L_i}{\partial \dot{l}_i} \right) - \frac{\partial L_i}{\partial l_i} = -R_{xi} \frac{\partial x_i}{\partial l_i} - R_{yi} \frac{\partial y_i}{\partial l_i} - R_{zi} \frac{\partial z_i}{\partial l_i} + \begin{bmatrix} 0 & 0 & F_i \end{bmatrix} \]

(3.23)

In this equation, \( L_i \) represents the Lagrangian of the individual link, \( l_i \) is the link length vector and \( F_i \) represents the forces along the actuator. Actuator force can be found by resolving the external moment into forces along links using robot’s Jacobian matrix. The reaction forces are of opposite sign compared to the ones used in equation (3.22). Further, the relation in the Cartesian and the joint space coordinates can be explained as below:

\[
\frac{\partial x_i}{\partial \theta} = \begin{bmatrix} \frac{\partial l_i}{\partial (\theta, \phi, \psi)} \end{bmatrix}^T \frac{\partial x_i}{\partial l_i} \]

(3.24)

After arrangements the final dynamic equation of robot takes the form shown below.

\[
\frac{d}{dt} \left( \frac{\partial L_{MP}}{\partial \dot{\theta}} \right) - \frac{\partial L_{MP}}{\partial \theta} + \sum_{i=1}^{n} \left( \frac{\partial l_i}{\partial (\theta, \phi, \psi)} \right)^T \left( \frac{d}{dt} \left( \frac{\partial l_i}{\partial l_i} \right) - \frac{\partial L_i}{\partial l_i} \right) = J^{-T}F \]

(3.25)

Actuator used in the present work exhibit non-linear and time dependent behaviour and hence their dynamics is difficult to carryout using the conventional approach.
Nevertheless, a fuzzy logic based dynamic model has been developed for actuator dynamics and details of its development and implementation are discussed in Chapter 8.

3.7 Chapter Summary

This Chapter discussed ankle joint anatomy, possible ankle disorders and process of conventional therapeutic treatment. Useful inputs were drawn from this discussion to conceptualize a wearable ankle rehabilitation robot design. The proposed design was kinematically compatible with the human ankle joint and had an arrangement of actuators which allows natural foot-ankle motions while keeping the ankle joint position stationary. Emphasis was given to the wearability aspect of the design so that the resulting ankle robot design is portable, ambulatory, and light weight. Actuators used in this robot were back-drivable, exhibit skeletal muscle like behaviour and provide compliant actuation. Construction of the wearable ankle robot was explained with a brief note on its design specifications. Symbolic kinematic modelling was performed analysing the inverse kinematics solution. Geometrical modelling of the wearable robot was carried out to obtain a Jacobian matrix. It was emphasized that the Jacobian matrix analysis is important in view of the wearable robot design evaluation and better controllability. System modelling was concluded with a discussion on dynamic modelling of the robot using Lagrangian approach.
Chapter 4 Forward Kinematics Modelling Using Modified Fuzzy Inference

This Chapter deals with Forward Kinematics (FK) mapping of the wearable ankle robot. Owing to the parallel arrangement of actuators the wearable robot exhibit highly coupled nonlinear motions hence conventionally a unique closed form solution of their FK cannot be obtained. However, since FK is a key module in closed loop position and force control, its accurate and fast solution is indispensable. Quite a few approaches have been mentioned in the literature to solve the FK of parallel robots; however given the online use of FK, there’s a need to develop an algorithm which can provide better accuracy with improved computational efficiency. To solve the FK problem, a modified FIS is being proposed in this Chapter which is time efficient and becomes very accurate when system parameters are optimized. Subsequently, FIS has been optimized using three approaches namely; gradient descent (GD), GA and MGA. The MGA, which has been proposed for the first time, has been designed to improve local search capabilities of GA for enhanced accuracy. The FIS, optimized by MGA has been found to be more accurate compared to the GD and GA optimized FIS. Performance of the MGA based fuzzy system has also been found better both in terms of accuracy and computation time, when compared with Newton-Raphson iterative method.

4.1 Forward Kinematics

While the robot joint displacements can be obtained from position sensors located on the PMA, it is generally difficult to measure the exact position and orientation of the robot end-effector. Inclinometers using internal gyroscopes can be used to determine the orientation but there are problem associated with noisy signals and undesired weight increment of the wearable robot. The configuration of the end-effector in task space is therefore often obtained through the use of FK.

In a position controlled system, the desired trajectory is normally specified in the task space and inverse kinematics is used to resolve the required joint displacements to realize the corresponding task space configuration. However, in more complex control problems such as encountered while controlling proposed ankle robot, the robot is operating within an
uncertain environment and the behaviour of the robot is required to be controlled depending on the task space configuration of the end-effector. In other words, the control law is given in the task space coordinates. The ankle robot interacts physically with patient’s foot and hence is desired to change its interaction force level according to the motion error in task space. In such applications, FK should be computed in real time since it is difficult, if not impossible, to pre-plan the robot’s joint displacements to the small perturbations in the end-effector, imposed on it by the uncertain environment.

During kinematic calibration, the end-effector configuration is measured and its gradient with respect to the kinematic parameters is calibrated to adjust the robot kinematic parameters to reduce errors between the actual and estimated task space configuration. Since this gradient is to be computed numerically, the FK routine will have to be run multiple times with slightly different robot kinematic parameters. Accuracy of the FK algorithm will thus be important to ensure correct computation of the previously discussed gradient. When the FK routine is used in the overall control loop, the time required for its execution should be as short as possible to improve real time control performance. Numerical (Newton-Raphson method) and AI based approach (Neural network based methods) can provide unique solution to the FK problem [21]. Though these two approaches can provide good accuracy, they are computationally expensive and hence are not recommended in the present case. Instead, alternate approaches, to solve FK problem, are investigated which can offer reduced computational complexity while providing improved accuracy. As the outcome of this research a fuzzy logic based computational model shall be proposed to solve the FK problem for the wearable ankle robot. It will be shown that the fuzzy model is fast and provides accurate solutions when compared with the conventional Newton-Raphson method.

In the following Sections, the Newton-Raphson iterative method and the proposed Fuzzy logic based approach will be introduced, implemented and compared for their accuracy and computation time.

4.2 Forward Kinematics Solutions

4.2.1 Newton-Raphson Method

The Newton-Raphson algorithm is a numerical method commonly used to solve system of nonlinear equations (3.3) which are difficult to solve analytically. The kinematic model of the parallel robot in consideration can be represented as (4.1), with \( l \) being the vector of
actuator displacements and $\Theta$ being the vector of the ZYX Euler angles used to describe the pose of the end-effector of the ankle rehabilitation robot.

The Newton-Raphson algorithm is used to solve the optimization problem which aims to minimize the discrepancies between the estimated actuator displacements based on an estimated end-effector pose and the measured actuator displacements. The estimated actuator displacements can be found using the kinematic model as shown in (4.2), where the estimated quantities are denoted with the operator $\hat{\cdot}$. The cost function of the optimization problem is given in equation (4.3) and the optimization problem can be represented as (4.4). The iterative algorithm is then given in equation (4.5), where $J$ is the Jacobian matrix which contains information regarding the gradient of the actuator displacements with respect to changes in the end-effector pose. Terms in row $i$ of the Jacobian matrix, are given in equations (4.7–4.9). The Newton-Raphson algorithm is iterated until the change in cost function falls below a threshold value set as $10^{-4}$ in the present case

$$l_i^2 = (P_e^o + R_e^o a_i^e - b_i^o)^T (P_e^o + R_e^o a_i^e - b_i^o)$$

$$l = f(\theta)$$

$$\theta = [\theta_1 \ \theta_2 \ \theta_3]^T \text{ and } l = [l_1 \ldots l_4]^T$$

$$\hat{l} = f(\hat{\theta})$$

$$C = (l - \hat{l})^T (l - \hat{l})$$

$$\min_{\hat{\theta}} C(l, \hat{\theta})$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + (J^T J)^{-1} J^T (l - \hat{l})$$

$$J = \begin{bmatrix}
\frac{\partial l_1}{\partial \theta_1} & \frac{\partial l_1}{\partial \theta_2} & \frac{\partial l_1}{\partial \theta_3} \\
\frac{\partial l_2}{\partial \theta_1} & \frac{\partial l_2}{\partial \theta_2} & \frac{\partial l_2}{\partial \theta_3} \\
\vdots & \vdots & \vdots \\
\frac{\partial l_4}{\partial \theta_1} & \frac{\partial l_4}{\partial \theta_2} & \frac{\partial l_4}{\partial \theta_3}
\end{bmatrix}$$

$$\frac{\partial l_i}{\partial \theta_x} = \frac{1}{l_i} \left( \frac{\partial R_e^o}{\partial \theta_x} a_i^e \right)^T (P_e^o - b_i^o)$$
4.2 Forward Kinematics Solutions

\[
\frac{\partial l_i}{\partial \theta_y} = \frac{1}{l_i} \left( \frac{\partial R_e^0}{\partial \theta_2} \alpha_i^e \right)^T (P_e^o - b_i^o)
\]

\[
\frac{\partial l_i}{\partial \theta_z} = \frac{1}{l_i} \left( \frac{\partial R_e^0}{\partial \theta_3} \alpha_i^e \right)^T (P_e^o - b_i^o)
\]

Figure 4.1: Actuator lengths (a), measured and estimated pitch angles (b) and corresponding estimation error (c) for a set of data obtained from actual robot platform using Newton-Raphson Iterative method.

The above iterative numerical approach can be accurate, but due to its iterative nature, the algorithm takes more than 140 ms for 100 computations. Actuator lengths, corresponding pitch angles and the estimation error for a testing database obtained from actual robot platform using the Newton-Raphson iterative method is illustrated in Figure 4.1. A lag in the computed values of pitch angles was observed due to signal noise in measurements resulting in the large error (>0.1 radians) deviations. The MSE in the measured and estimated pitch angles from the testing data was found to be 0.0021 radians. These results are further discussed while compared with other approaches in the Section 4.6.
4.3 Fuzzy Inference Approach

Fuzzy inference approach is employed next to develop a model which in turn can map the joint variables with the Cartesian variables of the proposed robot. Out of the two types of fuzzy modelling available, namely, the Mamdani based fuzzy model and the Takagi-Sugeno based fuzzy model, the Takagi-Sugeno based fuzzy inference is used for the reason explained in the following Section [154].

4.3 Fuzzy Inference Approach

The proposed Takagi-Sugeno (TS) fuzzy system was first introduced by Takagi and Sugeno [155] and since then it has been extensively used as an inference system. A FIS has three main building blocks. They are the fuzzification of antecedent and consequent variables, construction of a rule-base describing the relationship between these variables and formulation of an inference mechanism to provide crisp consequent variables. There are two basic approaches of fuzzy system modelling in the literature i.e. linguistic fuzzy modelling and precise fuzzy modelling [154]. Linguistic fuzzy modelling, also termed as Mamdani Approach has high interpretability but lacks accuracy. On the other hand, precise fuzzy modelling such as Takagi and Sugeno’s Approach exhibits high accuracy at the cost of interpretability. Interpretability is a measure of understanding of the system behaviour and expressing it through a model. The accuracy of a fuzzy model indicates how closely it can represent the system being modelled. Mamdani fuzzy system requires small input-output database for tuning and can interpret system behaviour between the discrete data. Conversely, the TS fuzzy system requires large database with high resolution to tune as it lacks interpretability.

Figure 4.2: Block diagram illustrating design of the fuzzy inference system
In the Mamdani approach, both input and output variables are fuzzy variable and hence a defuzzification method is required to convert fuzzy outputs to crisp outputs to be able to use it. Conversely, the Sugeno approach uses fuzzy input variables and produces crisp output variables and therefore a simple weighted average of all the rule outputs is taken as the aggregated output for a given set of input variables. In the present application since the accuracy is more emphasized, the Sugeno inference engine has been chosen for the proposed fuzzy model which has been shown using a block diagram in Figure 4.2. As can be seen, inputs to the fuzzy system are passed through a three step process to finally obtain the outputs. These steps have been defined using three building blocks of the fuzzy inference mechanism. These blocks, in the context of present problem are briefly explained below.

### 4.3.1 Fuzzification

In the present problem of finding FK solution for ankle robot, input variables are a set of four link lengths whereas outputs are orientations of the robot end-effector about three axes. During fuzzification, the antecedent variables (or the input variables) of the system are converted into fuzzy variables using fuzzy sets. A Gaussian Activation Function (AF) is chosen to describe the type of input variable’s fuzziness. It was revealed from the simulations that the Gaussian functions performed better in the present case, compared to the triangular or the trapezoidal activation functions. Further, to initialize fuzzy system design, a decision is made on the number, shape and the position parameters of the AFs. To begin with, the minimum fuzziness points (e.g. point q1 in Figure 4.3) of the fuzzy AFs are placed equally dividing the universe of discourse (operating range) of the variable. Standard deviation of the Gaussian AF’s (q2) is assumed to be equal to the universe of discourse.

![Figure 4.3: The fuzzy Activation Function (low, medium and high) and the tuning parameters q1 & q2](image-url)
4.3 Fuzzy Inference Approach

Conventionally, the consequent variables are defined using a set of real numbers or first degree polynomial functions of the input variables [156]. Thus a non-linear consequent variable is converted or decoupled into sum of several linear variables. Based on the type of consequent variables (whether zero order or first order polynomial functions of inputs), the TS fuzzy system is termed as the zero order TS fuzzy system or the first order TS fuzzy system. Higher order fuzzy systems (second and higher order) normally are not considered due to their non-linearity and exponentially increased complexity. For simplicity, the zero order and the first order TS fuzzy systems shall be defined henceforth as constant and linear TS fuzzy systems. Consequent variables are identified using the pseudo-inverse operator in the inference mechanism block as explained later in this Section.

4.3.2 Rule-base

Rule-base of a TS fuzzy system relates the fuzzy antecedent and crisp consequent variables using if-then statements and has the following structure.

\[
\text{If } l_1 \text{ is } A_{i_1} \text{ and, } \ldots \ldots \text{ and } l_4 \text{ is } A_{i_4} \text{ then } \theta_{i_1} \text{ is } y_{i_1}, \theta_{i_2} \text{ is } y_{i_2} \text{ and } \theta_{i_3} \text{ is } y_{i_3}
\]

Here \( i, (i = 1, \ldots, n) \) is the rule number, \( j, (j = 1, 2, 3) \) is the output number, \( A_{i_1}, \ldots, A_{i_4} \) are the membership or activation functions in the antecedent part, \( l_1, \ldots, l_4 \) are input link lengths of the parallel robot. Further, \( \theta_{i_1} \ldots \theta_{i_3} \) are the output orientations of the end-effector about three axes and \( y_{i_j} \) is a linearly parameterized consequent part of the rule-base. Ideally the total number of rules is derived from the number of AF’s and consequent variables using following relation [156].

\[
N_r = \prod_{k=1}^{n} m_k
\] (4.10)

Here \( N_r \) is the total number of rules; \( n \) is number of input variables and \( m_k \) is the number of linguistic terms of \( k^{th} \) input variable. For example if there are four inputs (link lengths) and each input has three AF’s, then the total number of rules shall be \( 3^4 \), which is 81. As discussed previously, the order of the TS fuzzy system will affect the parameters used to define \( y_{i_j} \). In general terms, the consequent part can be represented in the form shown in (4.11). Where \( v_{i_j} \) is an input dependent vector, \( c_{ij} \) is a vector of real numbers and \( \rho_{ij} \) is the parameter vector. These vectors for the constant and linear TS fuzzy systems are given in Table 4.1.
4.3 Fuzzy Inference Approach

\[ y_{ij} = v_{ij}^T \rho_{ij} \]  \hspace{1cm} (4.11)

### Table 4.1 Order of TS fuzzy system and consequent parameters

<table>
<thead>
<tr>
<th>TS fuzzy model type</th>
<th>( v_{ij} )</th>
<th>( \rho_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>( l )</td>
<td>( c_{ij} )</td>
</tr>
<tr>
<td>Linear</td>
<td>([l_1 l_2 l_3 l_4]^T)</td>
<td>([\rho_{ij0} \rho_{ij1} \rho_{ij2} \rho_{ij3} \rho_{ij4}]^T)</td>
</tr>
</tbody>
</table>

4.3.3 Inference Mechanism

Output of the TS fuzzy model for a given set of input variables, is evaluated using its inference engine, which uses a two step procedure [156]. Fundamentally, it computes weighted average of the outputs from all the rules for given input values. First of all, the degrees of fulfilment \( \mu_{ij} \) of input values \( \xi_{ij} \), in all the rules are calculated using a product operation using (4.12). Subsequently the final output \( \theta \) is computed using weighted average of the individual rule fulfilments (4.13).

\[ w_i = \prod_{k=1}^{4} P_{ik} A_k(l_k) \] \hspace{1cm} (4.12)

\[ A_k(l_k) = [A_{k1} \; \ldots \; A_{kM}]^T \quad A_{km}(l_k) = a e^{-\frac{(l_k - \bar{l}_{km})^2}{2\sigma_{km}^2}} \quad m = 1, \ldots, M \]

Here \( P_{ik} \) is a selection vector which chooses the activation function to be used in input \( k \) and rule \( i \).

\[ \theta_j = \frac{\sum_{i=1}^{N_r} (w_i \cdot y_{ij})}{\sum_{i=1}^{N_r} w_i} = \frac{\sum_{i=1}^{N_r} (w_i \cdot v_{ij}^T \rho_{ij})}{\sum_{i=1}^{N_r} w_i} \] \hspace{1cm} (4.13)

Here \( w_i \) is the Gaussian activation value of \( i^{th} \) inference rule and \( y_{ij} \) as explained in the previous subsection, is a set of real numbers, which is the consequent part of the rule-base. It is useful to note that (4.13) can be further rewritten as (4.14), with \( x \) being a regressor vector and \( \Gamma_j \) is a vector formed by augmenting all the parameter vectors \( \rho_{ij} \) from \( i = 1 \) to \( i = N_r \).

\[ \theta_j = x^T \Gamma_j \] \hspace{1cm} (4.14)

When multiple observations are available, these observations can be combined together to give a system of equations of the form shown in (4.15), where \( \Theta_j = [\theta_{j1}^j \ldots \theta_{jp}^j \ldots \theta_{jN}^j]^T \) is the vector of desired output and \( X = [x^1 \ldots x^p \ldots x^N]^T \) is the regressor matrix. Here, \( p \) denotes the pattern/observation number and can range from 1 to \( N \).
4.4 Optimization of Fuzzy Inference System

\[ \theta_j = X I_j \]  \hspace{1cm} (4.15)

Fuzzy parameters are tuned or optimized using a training database obtained from the IK analysis. Thus for a given set of input and output data where the number of observations is greater than the total number of parameters, the parameters for the consequent part of the fuzzy system can be obtained by solving the least squares problem using the pseudo-inverse operation as shown in (4.16).

\[ I_j = \text{pinv} (X) \theta_j \]  \hspace{1cm} (4.16)

Initially, the fuzzy system parameters such as number, location and spread of the fuzzy AF’s are decided intuitively. As a result such a system is not very accurate and to achieve higher accuracy these parameters are required to be tuned. There exist optimal values of these fuzzy parameters for which the model can emulate the system with finer accuracy. Hence, to minimize the inference error from the fuzzy model, it is required that these parameters be optimized within their limiting values. Thus, it is a constrained optimization problem involving multiple variables which are non-linear in nature.

Optimization of fuzzy inference systems has been researched rigorously during the last decade and few works have been presented in the literature. A comprehensive review illustrating the application of various non-linear optimization schemes can be found in [157]. Some of the representative algorithms used by researchers are least square method, genetic-fuzzy approach, neural-fuzzy approach, ant colony optimization, tabu search and simulated annealing optimization algorithm. Though all these algorithms have been successfully implemented in a variety of applications, neural-fuzzy approach (which is based on a GD approach) and genetic-fuzzy method are found to be more accurate [157]. In the present work, both of these methods have been used along with a MGA approach to optimize the fuzzy model of the system. In the subsequent section, the methodologies, implementation and the results obtained from these approaches have been analyzed and compared.

4.4 Optimization of Fuzzy Inference System

Optimization of FIS is a non-linear problem. It is therefore a good candidate for the use of metaheuristics such as GA. In the absence of a near optimal initial solution the gradient based methods cannot provide a globally optimized solution. On the other hand GA works with population of solutions and can guarantee a globally optimal solution. It has been observed
that the GA and the gradient methods are complementary to each other [154]. GA can provide a solution that is close to the global optimum quickly but fails to further fine tune it because its local search capabilities are not equally good. On the contrary, gradient based methods have good local search capabilities but normally get trapped in a local optimal solution in the absence of a correct initial guess.

During this research six types of fuzzy models have been designed comprising of constant and linear TS fuzzy systems models of two, three and four AF’s each (Figure 4.4). The Gaussian AF’S have been defined linguistically as low (L), medium (M), high (H) and very high (VH). The interpretability of fuzzy system increases when more AF’s are used and their overlaps are reduced [156].

As can be seen from Figure 4.4, individual AF’s encompass the whole allowable range of actuator lengths, thus every length measurement has some activation value in all the AF’s. For instance, a length measure of 0.4m has activation values as (0.54/L, 0.32/H), (0.59/L, 0.95/M, 0.27/H) and (0.136/L, 1.0/M, 0.36/H, 0.045/VH) respectively for 2, 3 and 4 AF choices. Obviously, the activation value of 0.4m measurement (which eventually is medium) is more explicitly defined when four AF’s are used.

Further, these six fuzzy models have been optimized using three different approaches namely, GD method, GA and modified GA. Thus a total of eighteen models have been developed and their accuracies and computational complexities are compared. Detailed analysis of different types of fuzzy models and optimization methods has been performed to provide a variety of choice for broader future applications. This study can help future researchers in establishing a trade-off between accuracy and computation time in solving FK.

Fuzzy logic toolbox called *anfisedit* is available in Matlab software which facilitates the development of a TS fuzzy inference system and optimization of fuzzy parameters using the gradient based approach. However this toolbox cannot be used in the present problem since it only works for systems having a single output and in the present problem there are three outputs for the three Euler angle of the end-effector. Therefore, relevant codes required to develop TS fuzzy system and implementation of optimization algorithms namely, GD, GA and MGA have been written in Matlab codes for implementation and simulations purpose.

A table (Table 4.2) listing various fuzzy models with numbers of their tunable parameters (obtained from (4.11) and Table 4.1), has been provided for quick reference. It is to be noted here that these parameters have been given to determine only one Euler angle.
4.4 Optimization of Fuzzy Inference System

Figure 4.4: Plot of 2, 3 and 4 Activation Functions for fuzzy actuator lengths

4.4.1 Problem Formulation

When a specific relation between inputs and outputs of a system is desired to be established and some example data are available, the fuzzy parameters can be optimized to produce the desired system model. The example or training data for the present problem has been obtained using (3.3). The tuning of fuzzy parameters can be achieved by using an optimization algorithm which minimizes an objective error function described below.

Minimize

$$
H = E(H_1, H_2, H_3)
$$

Here $H_1$, $H_2$ and $H_3$ are the expected values of the squared errors in the magnitude of end-effector pose in three orientations, each described by a relationship of the form shown in (4.17).

Table 4.2 Types of fuzzy models studied and their tunable parameters

<table>
<thead>
<tr>
<th></th>
<th>TS fuzzy model type</th>
<th>Number of non-linear parameters</th>
<th>Number of linear parameters</th>
<th>Total number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2-AF constant</td>
<td>16</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>2-AF linear</td>
<td>16</td>
<td>80</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>3-AF constant</td>
<td>24</td>
<td>81</td>
<td>105</td>
</tr>
<tr>
<td>4</td>
<td>3-AF linear</td>
<td>24</td>
<td>405</td>
<td>429</td>
</tr>
<tr>
<td>5</td>
<td>4-AF constant</td>
<td>32</td>
<td>256</td>
<td>288</td>
</tr>
<tr>
<td>6</td>
<td>4-AF linear</td>
<td>32</td>
<td>1280</td>
<td>1312</td>
</tr>
</tbody>
</table>
4.4 Optimization of Fuzzy Inference System

\[
H_j = \frac{1}{N} \sum_{p=1}^{N} \left[ \frac{1}{2} \left( |\theta_j^p| - |\theta_j^{rp}| \right)^2 \right] 
\]  

(4.17)

Here \( \theta_j^p \) is the outputs from the fuzzy model and are calculated by taking weighted average of the truth values of all the fuzzy rules as shown below.

\[
\theta_j^p = \frac{\sum_{i=1}^{n} \left( \prod_{k=1}^{4} a e^{-\frac{(t_k^p - \tilde{l}_{ik})^2}{2\sigma_{ik}}} \right) y_{ij}^p}{\sum_{i=1}^{n} \left( \prod_{k=1}^{4} a e^{-\frac{(t_k^p - \tilde{l}_{ik})^2}{2\sigma_{ik}}} \right)} 
\]

(4.18)

Subjected to

\[ a > 0, \alpha_{k1} < \tilde{l}_{ik} < \beta_{k1} \text{ and } \alpha_{k2} < \sigma_{ik} < \beta_{k2} \]

The objective error function is the mean of orientation errors recorded about the three axes. Further \( |\theta_j^{rp}| \) are the absolute measures of roll, pitch and yaw orientations of the end-effector, obtained from inverse kinematics and is the desired output value for the \( p^{th} \) input data. Similarly, \( |\theta_x^p| \) etc. are the outputs from the fuzzy model for the same \( p^{th} \) input data.

Output of the fuzzy model can be calculated using (4.18), where ‘i’, \( i = 1, \ldots, n \) is the number of inference rules, \( k, k = 1, \ldots, 4 \) is the number of input variables to the fuzzy model, \( \tilde{l}_{ik} \) is the position of the centre of the peak and \( \sigma_{ik} \) is the width of the Gaussian activation function for \( i\)-th rule and \( k\)-th input. As defined previously, \( y_{ij} \) is the consequent part of the fuzzy rule-base, \( a \) is a positive constant and \( e \) is the Euler’s number. The constraints mentioned above are the ranges of two parameters (shown as q1 & q2 in Figure 4.3) of the Gaussian AF. These are the position of minimum fuzziness point \( (\tilde{l}_{ik}) \) and the standard deviation \( (\sigma_{ik}) \) of the Gaussian function. Both of these parameters have different limits for different input variable and hence these limiting values have been shown as positive constants \( \alpha_{k1}, \alpha_{k2}, \beta_{k1} \) and \( \beta_{k2} \).

The numbers of tunable parameters for the fuzzy models considered have been provided in Table 4.2 and can be explained with an example. The constant TS fuzzy model has four inputs and if each of these is described by three AF’s, twelve activation functions are obtained in all. Since a Gaussian activation function can be defined by two parameters \( (\tilde{l}, \sigma) \) the total number of non-linear parameters to be tuned becomes 24. Further, as discussed in Section 4.3, the fuzzy system will have 81 rules and have equal number of consequent \( (r_i) \) or
4.4 Optimization of Fuzzy Inference System

output variables. Thus to optimize this fuzzy model it is required to optimize 24 non-linear parameters and 81 linear parameters resulting a total of 105 variables to optimize. Number of parameters for other fuzzy models has been computed in a similar fashion. In the following sections three approaches namely GD, GA & MGA, employed to optimize these parameters have been discussed.

4.4.2 Gradient Descent Optimization

A gradient-descent method can be also used to optimize the antecedent and the consequent parts of the FIS rule-base. The numerical method can update the parameters in order to minimize an objective error function \( H_i \) described in the previous section. The update rule for various parameters is stated as below.

\[
p_{kmj}(t + 1) = p_{kmj}(t) - \alpha \sum_{p=1}^{N} \frac{\partial H_j^p}{\partial p_{kmj}} (4.19)
\]

Where \( p_{km} \) is a vector of non-linear parameters of antecedent fuzzy activation functions for input \( k \), activation function \( m \) and output \( j \), \( t \) is the number of epochs, constant \( \alpha \) is the learning rate, which decides the quantum of change in the parameters after each iteration and \( N \) is the number of training patterns. Consequent variables, as explained before, are not tuned using GD approach but they are identified using pseudo-inverse. As an alternate to the pseudo-inverse method, other regression methods such as Least square regression, LU decomposition, Quantile regression etc. could also be used. However, these methods sometimes result in the over fitting, of the function being approximated, and therefore not considered. The derivatives for the parametric updates shown in (4.19) are calculated using following chain rule [154].

\[
\frac{\partial H_j}{\partial p_{kmj}} = \sum_{p=1}^{N} \left( \frac{\partial H_j^p}{\partial \theta_j^p} \cdot \frac{\partial \theta_j^p}{\partial w_{ij}^p} \cdot \frac{\partial w_{ij}^p}{\partial A_{kmj}^p} \cdot \frac{\partial A_{kmj}^p}{\partial p_{kmj}} \right) (4.20)
\]

The partial derivatives are further derived using relations explained below.

\[
H_j = \frac{1}{2} (|\theta_j^p| - |\theta_j^{pp}|)^2 \quad \text{or} \quad \frac{\partial H_j}{\partial \theta_j^p} = (|\theta_j^p| - |\theta_j^{pp}|) (4.21)
\]

\[
\theta_j^p = \frac{\Sigma_{i=1}^{N_r} (w_{ij}^p \cdot y_{ij}^p)}{\Sigma_{k=1}^{N_r} w_{ij}^p} \quad \text{or} \quad \frac{\partial \theta_j^p}{\partial w_{ij}^p} = \frac{(y_{ij}^p - \theta_j^p)}{\Sigma_{k=1}^{n} w_{ij}^p} (4.22)
\]
4.4 Optimization of Fuzzy Inference System

\[ w^p_{ij} = \prod_{k=1}^{4} p_{ik} A^p_{kj}(l^p_k) \quad \text{or} \quad \frac{\partial w^p_{ij}}{\partial A^p_{kmj}} = p_{ik} \frac{\partial A^p_{kj}}{\partial A^p_{kmj}} \prod_{k \neq k} p_{ik} A^p_{Kj}(l^p_k) \quad (4.23) \]

Here \( A^p_{kj} = [A_{k1j} \ldots A_{kMj}]^T \) and \( p_{ik} \) is a vector which selects the corresponding activation function to be used for input \( k \) in rule \( i \).

Finally, to compute \( \frac{\partial A^p_{kmj}}{\partial \sigma_{kmj}} \) from \( A^p_{kmj}(l^p_k) = ae^{-\frac{(l^p_k-\bar{l}_{ik})^2}{2\sigma_{kmj}^2}} \), partial differentiation of \( A_{ik} \) with respect to its parameters \((\bar{l}_{ik}, \sigma_{ik})\) is calculated taking care that the activation values should always be greater than zero. Later, all these partial derivatives are substituted in the chain rule mentioned in (4.20) and then the update equation (4.19) is used to revise the parameters in order to reduce the error function. Consequent variables of the fuzzy model are identified using the method previously discussed.

Thus the gradient approach works iteratively in two stages, first using the pseudo-inverse operation to solve the consequent variables for some initial assumed antecedent parameters and then the antecedent parameters are updated using the error signal by GD approach. The resulting fuzzy models (obtained after the GD based optimization) were evaluated using validation data (Section 4.5) and their performances in terms of mean squared errors and computation time are displayed in Genetic Algorithm Based Optimization.

Since the objective error function (4.17) is multi-modal and not convex, the gradient method may converge in a local optimum and provide a fuzzy model which is less accurate. This observation is well supported from the results displayed in Table 4.3. These results are further discussed in Section 4.6.

Genetic algorithm on the other hand is able to skip local optimum, owing to two basic reasons. Firstly, GA discretizes the search space and evaluates a population rather than a single point which helps it in avoiding convergence to a local suboptimal solution. Secondly, it has probabilistic operators such as mutation and crossover which are capable of taking the solution out of local optima. Fundamentally, GA is a population based optimization technique and is governed by the natural genetics and Darwin’s principle of natural evolution. Essentially, it is an iterative search method which works on the concepts of probability. In the present work GA has been used to optimize the shape and position of the AFs of the antecedent variables. Basically, the location and the spread of the activation functions are changed to find their optimal values in order to minimize the error between fuzzy inference output and the desired output (obtained using IK). It has been mentioned in [154] that to
4.4 Optimization of Fuzzy Inference System

improve the accuracy from a fuzzy system, optimization of its antecedent AF’s is more
decisive compared to the consequent AFs. While developing TS fuzzy systems, consequent
variables are not optimized, rather they are identified using pseudo-inverse operation as
shown in (4.16). The fitness evaluation and other genetic operators are discussed here and for
a detailed reading on the working of GA, [104] can be referred.

Initially a population of 100 binary coded strings was generated using Knuth’s random
number generator [104]. Non-linear parameters for antecedent variables which are 16, 24 and
32 respectively for three types of TS fuzzy models, were represented by binary strings
wherein 16 bits were assigned for each variable. The solution accuracy of the order of $10^{-5}$
or higher can be ascertained with 16 bits.

\[
\begin{align*}
q_1 & = 1101\ldots1 \\
q_2 & = 1010\ldots0 \\
q_3 & = 1011\ldots1 \\
& \ldots \ldots \\
q_n & = 1011\ldots1
\end{align*}
\]

The 16 bit long binary segments of individual fuzzy parameter were converted into
decimal numbers and the fuzzy model was constructed accordingly. Thus each binary string
solution represents an independent fuzzy model. Later, each of the fuzzy models was
evaluated for the training input data and their outputs were compared with the training output
data. The expected values of the squared error for three Euler angles were calculated and their
average is found to evaluate the fitness function (4.24) which has been derived from the
objective function mentioned in (4.17). Since GA can only be used to maximize a function,
for present case of error minimization, an altered error fitness function has been used.

\[
F = \frac{1}{1 + H} \quad (4.24)
\]

At a later stage of GA, better binary solutions were selected based on their fitness values
and their multiple copies in proportion to their fitness were saved in the mating pool. The
Roulette-Wheel Selection method [104] has been used for reproduction. An eight point
crossover operator with 0.95 probability and mutation operator with 0.01 probability have
been performed on the selected strings as a usual practice [104]. Generally crossover
probability is kept close to one so that all the parent strings may get a chance to crossover.
Mutation helps in fine tuning the global optimum solution, but to avoid a random search its
probability is kept low.

Performance of all six types of GA optimized TS fuzzy models, in solving FK problem,
has been presented in Table 4.3 and discussed later in Section 4.6
4.4.3 Modified Genetic Algorithm Method

It has been observed [154] that GA quickly converges to a near global optimal solution but then it lacks in fine tuning the global optima. Therefore, two modifications in the GA have been suggested in the present research which would enhance its performance and local search capabilities. Firstly to ensure that the best solution from a particular generation does not become extinct in the evolution process, an elitist approach [109] (normally employed in multi-objective optimization) has been used. In this scheme, the best solution of the previous generation is stored separately and is considered during fitness evaluation of the succeeding generation. Secondly, to enhance the local search capabilities of GA, a two step selection process has been used so that when GA, using Roulette Wheel Selection method, stops to reduce the error further, the selection method shifts to a gradient-based selection which has been proposed for the first time. In the gradient-based selection method, first of all, the best fit solution of the iteration is picked up. While performing local optimization it is desired that the derivatives/gradient of solutions be as small as possible. Therefore the stopping criterion for the algorithm is checked by finding the first derivative of the best fit solution (4.25). Central difference method is used to calculate the numerical derivative [104]. If the termination criterion is not satisfied, all other solutions are updated and moved in the direction of the best fit solution based on their parametric gradients towards the best fit solution. The update rule for the individual parameters of a solution can be given as (4.26), where $x^{best}$ represent parameters from the best solution. This modification enables the algorithm to search in the vicinity of the near global solution and fine tune the global optimal point. The steps involved in the modified GA are explained below.

\[
\text{If } |f'(x^{best})| < \epsilon, \quad \text{Terminate} \tag{4.25}
\]

\[
\text{Else } x^{(t+1)} = x^t + \frac{(x^{best} - x^t)}{x^{best}} \tag{4.26}
\]

Step1. Select a termination criteria based on either the number of epochs or the accuracy ($\epsilon$) required. In the present case the algorithm terminates either if the number of epochs ($t$) is 50 or if the error is less than $10^{-5}$. These numbers were decided after several simulation trials.

Step2. Initialize a random population of 100 binary strings with 16 bits for each variable. Also Initialize the maximum fitness value as $F_{max} = 0$.

Step3. Convert binary values of fuzzy system parameters to their decimal equivalent considering their universe of discourse into account as shown below:
for \( i = 1, \ldots, 18 \).

\[
I_{ik} = \alpha_{ki} + \frac{\beta_{ki} - \alpha_{ki}}{2^{b_i} - 1} \times (\text{decoded value of the binary string}) \tag{4.27}
\]

Here \( \alpha_{ki} \) and \( \beta_{ki} \) are lower and upper limits of the design variable, \( b_i \) is the length of the binary segment provided for a single variable.

Step 4. Calculate the fitness function for each binary solution from the decimal values of the parameters obtained in Step 3. Append the maximum fitness \( (F_{\text{max}}) \) binary solution obtained from the preceding iteration into the present generation of binaries. Find out the maximum fitness value of the combined population and compare it with the termination criterion. Continue if the number of epochs is less than 50, terminate otherwise.

Step 5. Save a copy of the best fit solution \( (F_{\text{max}}) \) separately and compare it with the previous best fit solution. Select the roulette wheel selection method for the next step of GA if the best fit solution is not the same as the previous best fit solutions. Otherwise use the new gradient based selection method as explained.


Step 7. Perform crossover operator on randomly selected parents and later perform mutation operator on the crossed-over binaries.

Step 8. Return to Step 3.

Optimized fuzzy models, obtained as a result of the three optimization approaches namely, GD, GA, and MGA, have been illustrated in Appendix A. Antecedent fuzzy variables which were obtained after their optimization, have been plotted for the eighteen fuzzy models for reference.

### 4.5 Training Data and Results

A training database consisting of 68921 sets of four link lengths and three end-effector orientations was created using inverse kinematics. To obtain this database, end-effector of the robot was rotated keeping its centre of gravity fixed in space. Incremental rotations were given to the end-effector within the ranges of \( \pm 20^\circ \), \( \pm 40^\circ \) and \( \pm 20^\circ \) respectively for Y, X and Z axes in accordance with the required rotations from the robot to perform ankle rehabilitation treatments [144].
4.5 Training Data and Results

Table 4.3 Prediction accuracies (average of MSE in three Euler angles) obtained using various fuzzy models and three kinds of optimization methods

<table>
<thead>
<tr>
<th>Type of TS Fuzzy Model</th>
<th>Optimization</th>
<th>Time for 100 computations (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GD</td>
<td>GA</td>
</tr>
<tr>
<td>2AF_Constant</td>
<td>3.361e-04</td>
<td>3.472e-04</td>
</tr>
<tr>
<td>2AF_Linear</td>
<td>3.417e-06</td>
<td>2.900e-08</td>
</tr>
<tr>
<td>3AF_Constant</td>
<td>1.191e-06</td>
<td>1.625e-07</td>
</tr>
<tr>
<td>3AF_Linear</td>
<td>3.561e-07</td>
<td>1.069e-11</td>
</tr>
<tr>
<td>4AF_Constant</td>
<td>3.870e-07</td>
<td>6.630e-11</td>
</tr>
<tr>
<td>4AF_Linear</td>
<td>2.151e-07</td>
<td>1.177e-13</td>
</tr>
</tbody>
</table>

The step size of one degree was considered for rotations about Y and Z axes whereas for X axis a two degree step size was considered. Inverse Kinematics (3.3) was used to calculate the associated four link lengths of the robot for above mentioned rotations. Later, this database was used to optimize six types of TS fuzzy models (Table 4.2) using three approaches namely, GD, GA and MGA.

Validation databases consisting of 9261 sets of robot link lengths and end-effector orientations were created with the help of IK (3.3), using \(21^3\) random combination of end-effector orientations. To evaluate a fuzzy model, simulation were run with the validation data and the mean squared error (4.17) from all the six fuzzy models was recorded. Comprehensive results of the average of MSE in three Euler angles (X, Y and Z) for six types of fuzzy models using validation data have been provided in Table 4.3. Computation time required for 100 sets of FK calculations have also been provided for these models. It is important to mention here that the mean squared error from the fuzzy models was found to be of the same order for different validation databases.

A Pareto chart exhibiting trade off between accuracy and the computation time for various fuzzy models has been provided in Figure 4.5 to help in selection of a particular fuzzy model depending on the requirements. Apparently, four out of eighteen models stretch out on the Pareto line, which are obviously better choices.
4.5 Training Data and Results

Figure 4.5: Pareto optimal curve realizing a trade off between accuracy and computation time (listed in Table 3) for eighteen fuzzy models, optimized using GD, GA and MGA.

To illustrate the performance enhancement of the fuzzy models by using MGA approach over simple GA approach, results from these two optimization methods have been compared and presented in Figure 4.6. Six fuzzy models each optimized by GA and MGA have been considered and due to lack of space their names have been abbreviated as explained below. The acronym uses number of activation functions as the first numeral followed by an alphabet ‘C’ or ‘L’ for constant or linear type of TS fuzzy model. The remaining part of the acronym suggests different methods of optimization in abbreviation e.g. GD, GA and MGA.

<table>
<thead>
<tr>
<th>Number</th>
<th>Acronym</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2CGA</td>
</tr>
<tr>
<td>2</td>
<td>2CMGA</td>
</tr>
<tr>
<td>3</td>
<td>3CGA</td>
</tr>
<tr>
<td>4</td>
<td>3CMGA</td>
</tr>
<tr>
<td>5</td>
<td>2LGAM</td>
</tr>
<tr>
<td>6</td>
<td>2LMGA</td>
</tr>
<tr>
<td>7</td>
<td>4CGA</td>
</tr>
<tr>
<td>8</td>
<td>4CMGA</td>
</tr>
<tr>
<td>9</td>
<td>3LGA</td>
</tr>
<tr>
<td>10</td>
<td>3LMGA</td>
</tr>
<tr>
<td>11</td>
<td>4LGA</td>
</tr>
<tr>
<td>12</td>
<td>4LMGA</td>
</tr>
</tbody>
</table>
Thus an acronym ‘4LMGA’ means a linear TS fuzzy model with four activation functions for antecedent variables which has been optimized by modified GA. Average of MSE in three Euler angles using validation database have been plotted for six types of fuzzy models each optimized by GA and MGA in Figure 4.7. Fuzzy models have been compared on varying scales due to large difference in their error values.

The testing database was obtained from the real robot platform using online sensor data. Robot platform was moved through desired ankle trajectories and while instantaneous link lengths were recorded using linear potentiometers, an inclinometer was used to measure and register the end-effector orientations. After duly validating the models, the most accurate model which is ‘4LMGA’ was employed on the testing database. Actuator lengths, measured and estimated pitch angles and corresponding estimation error for pitch angles have been displayed in Figure 4.8. Estimation errors for other orientations were found to be of the same order. Due to presence of noise in the sensor data, the model accuracy was not as good as it was with training and the validating databases. While the spread of MSE for validation database using ‘4LMGA’ fuzzy model was of the order of ±1e-07 radians, for testing database it was observed to be ±0.02 radians. MSE in pitch angles using testing database was found to be of 1e-06 order which is an improvement over the Newton-Raphson method (Figure 4.1).

To provide a quick glance through the advancements in the area of FK solution of parallel robots using artificial intelligence approaches, past research outcomes are arranged chronologically in Table 4.4. However, the comparison in terms of computation time shall not be fair since the previous research work mentioned here has been performed on machines of varying capabilities. Apparently, the proposed MGA based fuzzy approach performs better as accuracy of the order of 1.3358e-14 can be obtained using 4LMGA fuzzy model during simulations. Further owing to the less computational complexity of fuzzy approach, the fuzzy model is time efficient and can perform 100 computations in less than 3.5 ms.

4.6 Discussion

The mainstay of the work carried out, has been to reduce the computation time and simultaneously achieve better accuracy in solving the FK problem of the wearable parallel robot. Numerical and artificial intelligence based approaches were studied and applied since they can provide a unique solution. Newton-Raphson method has been the predominantly used numerical method in solving FK problem in the past and hence the same was used in the
present problem. Similarly FIS was preferred over purely GA and neural network (NN) approaches. This is because GA is time consuming [87] and not very accurate [93] at the same time NN cannot solve for the datasets which are of lower resolution than the training database and hence requires large database with low resolution for training.

![Graph showing error distribution between GA and MGA optimized fuzzy models.](image)

**Figure 4.7:** Comparison of error distribution between GA and MGA optimized twelve fuzzy models.

**Table 4.4 Summary of previous work on forward kinematics of parallel robots using Artificial Intelligence (AI) approaches**

<table>
<thead>
<tr>
<th>Authors</th>
<th>AI approach</th>
<th>Year</th>
<th>Accuracy MSE/SSE (sq. radians)</th>
<th>CPU Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geng and Haynes</td>
<td>Neural Network</td>
<td>1991</td>
<td>0.01(1%)</td>
<td>NM</td>
</tr>
<tr>
<td>Boudreau et al.</td>
<td>Genetic Algorithm</td>
<td>1996</td>
<td>1e-6</td>
<td>90-130</td>
</tr>
<tr>
<td>Yee and Lim</td>
<td>Neural Network</td>
<td>1997</td>
<td>2.9e-4</td>
<td>200</td>
</tr>
<tr>
<td>Sang and Han</td>
<td>Neural Network</td>
<td>1999</td>
<td>3.4e-4</td>
<td>NM</td>
</tr>
<tr>
<td>Sadjadian et al.</td>
<td>Neural Network</td>
<td>2005</td>
<td>8.1e-7</td>
<td>NM</td>
</tr>
<tr>
<td>Yurt et al.</td>
<td>Neural Network</td>
<td>2007</td>
<td>1.15e-4</td>
<td>1</td>
</tr>
<tr>
<td>Li, Zhu and Xu</td>
<td>PSO &amp; NN</td>
<td>2007</td>
<td>5e-4</td>
<td>100</td>
</tr>
</tbody>
</table>

NM: Not Mentioned

The Fuzzy systems have two types of inference mechanisms namely, Takagi-Sugeno and Mamdani fuzzy systems. TS fuzzy model was chosen over Mamdani model due to its
enhanced accuracy. TS fuzzy models are further classified as constant and linear TS fuzzy models. During present research both types of TS fuzzy models have been used with varying number of activation functions. It can be seen from the results (Figure 4.5) that by increasing the number of AF’s, accuracy from a fuzzy model can be increased but at the same time the computation time also increases. This fact is also evident from the results displayed in Table 4.3, wherein computation time for one hundred FK calculations have been provided for fuzzy models with two, three and four AF’s. While the computation time for a fuzzy model with two activation functions is just 0.54ms, it becomes seven folds when four activation functions are used. The accuracy on the other hand increases 1010 times when instead of two, four AF’s are used.

It has been also observed that while constant TS fuzzy models are faster, the linear TS fuzzy models are more accurate. Therefore, a trade off should be established before selecting a particular types of TS fuzzy model and number of AF’s. To provide further insight to the user, a Pareto chart (Figure 4.5) has been provided, which shows that out of eighteen fuzzy models (Table 4.3) four models lay on the Pareto line. When looking for a fastest algorithm, ‘2CMGA’ model is the best choice. However, when the accuracy of FK solution is emphasized, ‘4LMGA’ model can provide solutions with MSE of the order of $10^{-14}$. Further
two moderate choices could be ‘2LMGA’ (MSE as $10^{-8}$, 0.647ms) and ‘3LMGA’ (MSE as $10^{-12}$, 1.3ms).

![Figure 4.9: Actuator lengths (a), measured and estimated pitch angles (b) and corresponding estimation error (c) for a set of data obtained from actual robot platform.](image)

Effect of the modification suggested in the GA (Section 4.4.3) can be seen more clearly in the line chart shown in Figure 4.6, wherein the fuzzy models optimized by GA and MGA have been plotted for comparison. The MGA based fuzzy models are more accurate when all six types of fuzzy models are compared, nevertheless the improvement is more discernible for linear TS fuzzy models when three or four AF’s are used.

It is apparent from Figure 4.7 that, apart from decline in MSE, the range or the spread of MSE also gets reduced when MGA based TS fuzzy model is used. For instance, range of errors using ‘2CGA’ model was ±0.085 radians, whereas it was found to be ±0.05 radians when ‘2CMGA’ was used. Similarly, it was also observed that the linear type TS fuzzy models are more accurate than the constant types since error range for ‘4CMGA’ was ±1.05e-05 and same for ‘4LMGA’ was found to be 1e-06. Results from the testing data obtained from the real system (Figure 4.9) are also satisfactory, considering the noise present in the sensor measurements. Small amount of lag can be observed (Figure 4.1) when Newton-Raphson method is applied to the same database.
4.7 Chapter Summary

The FK problem of the wearable ankle robot was addressed in the present Chapter using fuzzy inference system. From the study of previous literature it was revealed that a new approach is required so that the FK computations can be performed faster without compromising on accuracy. Fuzzy inferencing requires less function evaluations hence it is time efficient. Moreover, when fuzzy inference systems are employed with optimized parameters, better accuracy can also be ensured.

To begin with, a frequently used numerical approach, known as Newton-Raphson iterative method was used to solve FK problem of the wearable ankle robot and it was found that though the algorithm can provide good accuracy, it is not time efficient and takes more than 140 ms for one hundred FK calculations.

To improve on the computation time of FK solution, alternate algorithms were studied and consequently an algorithm based on fuzzy logic was proposed. TS fuzzy systems were discussed and two of its variants namely, constant and linear TS fuzzy models were developed with two, three and four antecedent AF’s. These TS fuzzy models were then optimized using three approaches such as GD, GA and MGA. Thus a total of eighteen different fuzzy models were designed and optimized using a training database.

Subsequently, the fuzzy models were used to solve FK problem using a validation database and results obtained were analyzed. It was found that while linear TS fuzzy models were more accurate, constant TS fuzzy models were more time efficient (Table 4.3) Later, computation time and accuracy obtained from these eighteen fuzzy models were plotted and a Pareto optimal curve (Figure 4.5) was drawn. It was observed that four models (out of eighteen) were non-dominated i.e. all of these are equally better in terms of both accuracy and computation time. The Pareto analysis was performed to provide a wider choice of models to select, for different preferences of the time and the accuracy.

Improvement in GA, using the modifications suggested in this work, was emphasized by plotting MSE’s from all the fuzzy models (optimized by GA and MGA) together on a chart (Figure 4.6). It was demonstrated that for all six types of fuzzy systems, MGA based fuzzy models were more accurate than the GA based fuzzy models. With an increase in number of AF’s and subsequent optimization of fuzzy parameters, reduction in both, the MSE and the spread or range of error was recorded (Figure 4.7).

Finally the ‘4LMGA’ TS fuzzy model (Linear TS fuzzy model employing four activation functions and optimized by MGA) was selected for the present use and its performance was
evaluated using the testing database obtained from real system (Figure 4.8). The MSE in measured and estimated pitch angles was found to be of the order of 1e-06 which was better compared to the Newton-Raphson method (Figure 4.1). Similarly, it was found that the computation time for 100 FK computations, using 4LMGA fuzzy model was recorded to be 3.5ms which was far better compared to the Newton-Raphson method which takes 140ms.
Chapter 5 Design Analysis of the Wearable Ankle Robot

The wearable ankle robot, designed during the course of this research, set forth challenges with regards to its wearability requirement, use of a parallel mechanism, cable actuation and clinical requirements for ankle joint rehabilitation treatments. Wearability can be further explained in terms of requirements such as light weight, compact design, comfortable in use, safety and portability. While using parallel mechanisms for the robot, issues such as, smaller workspace and singularity, were needed to be addressed. Cable actuation required that the robot motion be achieved through positive actuator forces and the stiffness of the robot be analyzed in context to its rigidity. Finally, the robot’s application in the ankle joint rehabilitation stipulated higher actuator force requirement and set forth design constraints arising from its use by subjects of varying physical abilities. In the light of these challenges, it was desired that the robot design be analysed from various aspects and some performance indices be identified in order to find a trade-off between above mentioned limitations and achieve an optimum design.

![Design stages of the ankle parallel robot and related performance indices.](image)

Following an initial analysis of the literature [22, 158], seven important performance indices (PIs) were identified encompassing three main aspect of the robot design namely, kinematic design, actuation design and structural design. Singularity and condition number of the robot’s Jacobian matrix along with the isotropic workspace of the robot were considered to investigate its kinematic design aspect.
5.1 Kinematic Design

It is important to note here that the robot workspace determination is governed by two aspects of the robot design namely, kinematic and actuation design. Therefore, while investigating the actuation design aspect, apart from the actuator forces and the tensionability requirement, workspace of the robot was also carefully studied. Finally the structural design aspect was examined in context to the stiffness and rigidity of the robot. The relation between these PIs and their respective design aspects have been shown in Figure 5.1. The PIs are not mutually independent rather they are inter-reliant while some of them even conflicting. The interdependence of PIs and their dependence on condition number is shown in Figure 5.2. It has been established and shall be verified later in this work that the geometrical parameters of a parallel robot affect the above mentioned PIs and subsequently the three design aspects [94]. This Chapter formulates the mathematical representation of these PIs and investigates their dependence on the geometrical parameters, and how they affect the robot performance.

![Figure 5.2: Interdependency of the performance indices](image)

5.1 Kinematic Design

The wearable robot is made up of two parallel platforms which are connected together with four links. Thus the kinematic structure of the robot has four closed kinematic pairs and the robot motions are achieved through simultaneous motion of these kinematic pairs. While designing the wearable robot, its geometrical parameters, which define positions of these kinematic pairs, have been carefully selected in order to avoid the robot configuration becoming singular. The condition number of the robot’s Jacobian matrix, which is also a
5.1 Kinematic Design

measure of singularity, provides a relation between changes in the joint space and task space kinematic variables. Condition number is an important robot design parameter and solely depends on robot’s physical construction. Condition number and singularity aspect of the wearable robot are further discussed in view of their overall significance and dependence on robot’s geometrical construction.

5.1.1 Condition Number

Jacobian matrix of the robot \((J)\) maps its joint velocities to its Cartesian velocities. The condition number of this Jacobian matrix has important physical significance, which is apparent from the following relation (5.1) [87, 159, 160]. This relation shows that the mapping between joint force vector \(F\) and the task space moment vector \(M_{ext}\) can be obtained using the condition number \(k(J)\). Similarly other characteristics such as stiffness, velocities and acceleration can also be transformed from joint space to the Cartesian space using condition number properties.

\[
\frac{||\delta F||}{||F||} \leq k(J) \frac{||\delta M_{ext}||}{||M_{ext}||} \tag{5.1}
\]

A robot design with near unity condition number is desirable [97] since it minimizes the error in the end-effector torque due to input error in joint wrench. The condition number can also be used to evaluate the workspace singularities. It reveals how far the robot is from its present configuration to the nearest singular configuration. Eventually, the condition number is a vital design parameter and the robot configuration should be optimally designed to acquire a condition number close to unity. The Jacobian matrix of the proposed robot was calculated while performing geometrical modelling (Section 3.5) and it was found to be a \(4 \times 3\) non-square matrix due to the redundant actuation. To compute the condition number of this Jacobian matrix, its singular values are required which can be defined as the square root of the Eigen values of \(J^TJ\) and \(JJ^T\). Further, to find these singular values, \(J\) can be factorized with the help of the singular value decomposition rule as shown below.

\[
[J]_{4 \times 3} = [X^T]_{4 \times 4}[\Sigma]_{4 \times 3}[Y]_{3 \times 3} \tag{5.2}
\]

Where \(X\) and \(Y\) are orthogonal matrices and \(\Sigma\) is a diagonal matrix of three singular values, related as \(\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0\). The condition number \(k(J)\) is defined as the ratio of the largest singular value \(\sigma_1\) to the smallest singular value \(\sigma_3\) for a fixed orientation of the end-effector.
5.1 Kinematic Design

The range of condition number is described as below.

$$ 1 \leq k(J) \leq \infty $$ (5.4)

When the condition number approaches unity, the matrix $J$ is said to be well conditioned or far from singularities. On the contrary, if the condition number is higher, the matrix is said to be ill conditioned. An ill conditioned Jacobian matrix will further magnify the kinematic or dynamic error present in the robot joint motions and hence should be avoided.

To evaluate the robot design, condition number is generally obtained at different workspace points on the specified robot trajectory with assumed resolution. Though condition number at different end-effector orientation is useful information, to get a comprehensive view of its distribution in the entire workspace volume, a Global Condition Number (GCN) given by (5.5) is normally used [140, 161, 162].

$$ GCN = \frac{\int_{w} (k(J)) dw}{\int_{w} dw} $$ (5.5)

Here $w$ is the reachable workspace with the restricted stroke lengths of the actuators. Since it is difficult to calculate the exact solution to the integrals mentioned above, GCN has been discretely defined in the present work and expressed as below.

$$ GCN = \frac{\sum_{i=1}^{n} (k)}{n} $$ (5.6)

Here $n$ is the total number of discrete feasible points constituting the workspace and the numerator is the sum of condition numbers obtained at these points in the feasible workspace volume grid. Similar to the condition number, GCN is bounded by the range as given by (5.7).

$$ 1 \leq GCN \leq \infty $$ (5.7)

Once again, when GCN is a large number, the entire workspace tends to be ill conditioned and when the GCN is near unity the entire workspace is said to be well conditioned.

It is apparent from discussion in the last Chapter that the Jacobian matrix largely depends on the robot configuration, which is defined by the arrangement of connection points at both the platforms and the link lengths. Thus it can be concluded that the GCN, which is a global measure of the condition number of this Jacobian matrix, is governed by the robot’s
geometrical design. There exists an optimum robot configuration which will provide a near unity GCN and enhanced robot performance thereof [163]. However, since GCN is an average value over the robot’s workspace, there is an apprehension of overlooking certain undesirable peak values of the condition number during optimization process. To ensure that all the points in the workspace provide a condition number within certain range; the maximum value of the condition number in the entire workspace can be obtained and minimized. Once the GCN and maximum GCN are minimized, it can be ensured that,

1. The final GCN represents the average behaviour of the condition number over the feasible workspace.
2. The condition number all over the feasible workspace is always less than or equal to the optimized GCN value.

5.1.2 Singularity Analysis

During the course of its motion, the wearable robot which works on parallel mechanism, sometimes enters into a configuration wherein, instantaneously, it gains or losses extra degrees of freedom [22]. This configuration of the robot is referred to a singular configuration and as a result robot loses its stiffness and become uncontrollable. This phenomenon can be best explained using robot Jacobian matrix as below.

Equation (3.18) is rewritten here for quick reference.

\[ X \dot{q} = Y t \] (5.8)

Here \( \dot{q} \) is the vector of link velocities and \( t \) is the twist vector of the end platform, the Jacobian matrix \( J(q) \) of the robot is further defined as below.

\[ J = X^{-1}Y \] (5.9)

\[ \dot{q} = J(q)t \] (5.10)

A close inspection of (3.13&3.18) reveals that \( X^{-1} \) is a square matrix and always has a solution, on the other hand matrix \( Y \) is a \( 4 \times 3 \) matrix and can be rank deficient. Therefore, rank of Jacobian matrix \( J \) is decided by the rank of matrix \( Y \). In other words, matrix \( J \) will be singular when matrix \( Y \) is also singular. Subsequently, matrix \( Y \) was analyzed to deduce inferences regarding the configurations and geometries of the robot where it will enter into singularity. Diao et al. have presented an analytical approach in [164] to check the instances when matrix \( Y \) becomes rank deficient. The analysis has been presented for a simple planer
5.1 Kinematic Design

cable driven robot and the same is difficult to perform for the present case where the wearable ankle robot is an instance of a spatial cable driven parallel robot. Nevertheless a different perspective has been used here wherein the rank behavior of matrix $Y$ is analyzed with regards to its invertibility and the outcomes are supported by simulation results.

Referring to equations (3.20 & 5.9), $Y$ can be deduced as (5.11) to further explain its rank analysis.

$$Y = \begin{bmatrix} (a_i^p \times L_i^p)^T \\ \vdots \end{bmatrix} \quad \text{(5.11)}$$

The Link vector $L_i^o$ can be expressed in terms of position vector of the ankle joint and link connection points on the two platforms as (5.12). It can be easily shown that when $R_e^o = I$ and $a_i = \mu b_i$ i.e. when the two platforms have same orientation and proportional dimensions, all four lengths of the robot would be equal.

$$L_i^o = (p_e^o + R_e^o a_i^e - b_i^o) \quad \text{(5.12)}$$

It is important to note here that the full rank of matrix $Y$ (which is a $4\times3$ matrix) is three when its columns are independent. However during instances when all the lengths are equal, only two columns of $Y$ remains independent and its rank become two. Inverse of matrix $Y$ in such case does not exist as it has become rank deficient. As a result, Jacobian matrix (5.2) also loses rank and becomes rank deficient. Interestingly, the minimum singular value in this case becomes zero and the condition number of the Jacobian matrix, which is the ratio of largest singular value to the smallest singular value, in such instances becomes infinite. This inference can be further validated from the simulation results shown in Figure 5.3, where condition number distribution is shown at various robot orientations. The geometrical parameters of the robot are chosen so that the condition $a_i = \mu b_i$ holds true. Apparently, condition number values become infinite at the beginning of each cycle of robot motions (orientations 1st, 12th, 23rd) when the two platforms are aligned or $R_e^o = I$. Condition number values at other orientations are also very high which are not acceptable. Therefore, in the light of above findings, it is recommended that following arrangements of robot geometry should be avoided while designing the parallel robot.

Case I: The two platforms are of same size and have same orientations.

Case II: The two platforms are of slightly different size but have same orientations.
Case III: The two platforms are of different size and have zero separation or are coincident.

Case IV: The two platforms are of different size but are in the same plane.

5.2 Actuation Design

It has been emphasized in the previous Chapter that the wearable robot is actuated using PMA in series with cables. The PMA are flexible actuators which can provide pull/tension forces but fail to supply compressive forces. Thus use of PMA and cables for actuation of the robot required that during its course, the robot end-effector orientation be achieved with positive forces or tensions in the links, failing which the robot loses controllability. This requirement has been named as tensionability and is further explained in the following subsection. Appropriate placement of the actuators is important in achieving tensionability.

Comprehensive workspace of the wearable robot is a subset of the constituent workspaces from individual kinematic pair and hence is governed by robot’s geometrical design which includes parameters such as lengths and placements of its links and actuators. While working out actuator forces, it is explicable that, to realize required moments at the end-effector, the actuator forces largely depend on the placement of actuator connection points on the two platforms which defines robot design. Apparently when the actuators are connected close to the robot’s centre of rotation, the forces to realize certain moment at the end-effector are large compared to when the actuators are placed farther. The force closure, workspace and the actuator forces are further analyzed under the actuation design aspect of the robot and discussed in the following subsections.
5.2 Actuation Design

5.2.1 Force Closure and Tensionability

The ankle rehabilitation robot has three rotational degrees of freedom and during rehabilitation treatments, the end-effector of the robot is required to be rotated about a point where the ankle joint is theoretically positioned. The requisite moments by which the end-effector is moved are realized by providing suitable actuator forces. Apparently when the force closure is solved for certain end-effector moment, a vector, which has combination of positive and negative forces, is obtained for the actuator wrench (5.13). Here, a major constraint called tensionability is encountered as discussed below.

As discussed, cables used in conjunction with the PMA in the ankle robot have unidirectional properties i.e. they cannot provide compressive forces. Additionally, the flexible PMA can also provide only a positive or tension force. Thus the force closure obtained from (5.13) is not a feasible solution and some other means are to be explored to obtain a positive force vector. Further investigations revealed that, the redundant actuation in the present case provides an extra degree of freedom in the null space solution of the robot’s Jacobian matrix \( J \). This makes it possible that the desired orientations and moments at the end-effector are achieved through with positive (or tension) forces in the cables.

To carry out the force closure analysis, the joint space forces (5.14) which can produce a particular task space moment vector can be obtained using (5.13). Further, since the task space has three degrees of freedom, realized by using four actuators, the resulting end-effector Jacobian matrix is a non-square matrix. It can be easily verified that there will be an infinite number of actuator force vectors (5.14) which can provide the requisite task space moment (5.13). The extra component of null space forces can be arbitrarily selected without influencing the actual task space torque. Therefore, the extra degree of freedom in the null space of robot’s Jacobian matrix can be utilised to meet the tensionability constraint posed by cable actuation.

\[
\begin{align*}
M_{\text{ext}} &= J^T F \\
F &= J M_{\text{ext}}
\end{align*}
\]  
(5.13)  
(5.14)

Where \( \bar{J} = J(J^T J)^{-1} \) is the pseudo inverse of \( J^T \).

Next, at each point of the desired trajectory of the end-effector, forces in each cable are calculated using equation (5.13). Since the pushing force is not possible from the actuators, a positive force vector for the cables can be obtained using a quadratic minimization algorithm. The formulation of optimization problem is explained below:
Without loss of generality, the moment provided by the resulting positive force vector can be written as (5.14). Comparing (5.13) with (5.15) then yield (5.17, 5.18), where $V_1$ and $U$ are matrices containing the input and output basis vectors corresponding to non zero singular values of $J^T$ and $V_0$ is the null vector of $J^T$. Additionally, $\eta$ is a vector specifying the components of $\Delta F$ along the column vectors of $V_1$ and $\varepsilon$ is a scalar defining the component of $\Delta F$ along $V_0$.

$$M_{res} = J^T F_{res}$$  \hspace{1cm} (5.15)

$$\Delta M_{ext} = M_{res} - M_{ext}$$  \hspace{1cm} (5.16)

$$\Delta F = F_{res} - F = V_1 \eta + V_0 \varepsilon$$  \hspace{1cm} (5.17)

$$J^T = U \Sigma \begin{bmatrix} V_1^T \\ V_0^T \end{bmatrix}$$  \hspace{1cm} (5.18)

The relationship between (5.16) and (5.17) can be written as:

$$\Delta M_{ext} = J^T \Delta F$$

$$\Delta M_{ext} = U \Sigma \begin{bmatrix} V_1^T \\ V_0^T \end{bmatrix} [V_1 V_0] [\eta] \begin{bmatrix} \varepsilon \end{bmatrix}$$  \hspace{1cm} (5.19)

$$\Delta M_{ext} = U [diag(\sigma_1, \sigma_2, \sigma_3)] \eta$$

From above analysis, following objective function can be deduced.

Minimize

$$\Delta M_{ext}^T \Delta M_{ext} = \eta^T [diag(\sigma_1^2, \sigma_2^2, \sigma_3^2)] \eta$$

Subjected to

$$\mu_l \leq F_{res} \leq \mu_u$$

$$\Rightarrow \mu_l \leq [(J^+)^T M_{ext} + \varepsilon V_0 + V_1 \eta] \leq \mu_u$$  \hspace{1cm} (5.20)

Thus during the optimization a force vector is obtained which lies in the constrained limits given by (5.20) and is able to provide the desired moments at the end-effector with minimum error. In the present work, lower bound ($\mu_l$) of the actuator forces is consider to be zero and the upper bound ($\mu_u$) is defined as force threshold, another objective function which is later minimized. Since the links in the robot have been provided some pretension, the zero force in the links added with the pretension actually provides a positive actuator force.

### 5.2.2 Robot Workspace Analysis

The workspace of the proposed cable driven robot is difficult to analyze for three major reasons. Firstly, the orientation workspace is achieved through coupled motion of its links or cables which is difficult to evaluate independently. Consequently, the workspace has been
defined simply as the space where the inverse and forward kinematic solutions exist [112]. Secondly, workspace for the cable driven robots [153], is defined as the conglomeration of points where sets of positive cable tensions is attainable, a condition which has been discussed in the preceding Section. Finally, the compact design requirement of the robot poses a constraint on the actuator lengths which in turn constraints the reachable workspace. Since the length of the wearable ankle robot is governed by the length of its actuators, a compact design requires that the actuator lengths should be kept short to keep the total length of the robot close to the size of patient’s shinbone. The PMA, upon inflating can only expand to 30% of its normal length or in other words it can provide an actuation close to one third of its length. Thus, there is a limit on the stroke length and on the reachable workspace provided by the group of muscles.

The feasible workspace has been computed by carrying out a singularity analysis for the robot’s Jacobian matrix, checking the tensionability condition (discussed in the preceding Section) and observing the actuation constraint of the PMA at discrete workspace points. Mathematically, the feasible workspace index (I) is defined as below.

\[
I = \frac{\varphi_f}{\varphi_T} \tag{5.21}
\]

\[
\varphi_T = (\theta_{\text{max}} - \theta_{\text{min}})(\phi_{\text{max}} - \phi_{\text{min}})(\psi_{\text{max}} - \psi_{\text{min}}) \tag{5.22}
\]

Here \(\varphi_f\) is the feasible workspace which has been obtained after satisfying all above mentioned constraints and \(\varphi_T\) is the total orientation workspace with limiting values for Euler angles as \(\theta^{+25}, \phi^{+40} \text{ and } \psi^{+30}\).

### 5.2.3 Robot Actuator Forces

Due to higher stiffness of ankle joint, higher joint space forces are required to realize the necessary moment at the task space or the end-effector. As discussed in the previous sub-section, it is desired to keep the length of the robot and its actuators small for compactness of the robot structure. Apart from actuation limits, the capacity of air muscle to exert force also proportionally depends on its length i.e. longer PMA are required to realize higher actuator forces during ankle joint motions. Eventually to minimize the lengths of actuators the actuator force requirements should be reduced. Yet another motivation to minimize actuator forces is that the higher actuator forces may cause the cables to break or may also produce
undesired elongation in the cable and the flexible PMA, adversely affecting the positional accuracy.

Apparently, the actuator force is a function of robot’s geometrical parameters. By selecting actuator connection points on the robot platforms farther from the axis of rotation, the actuator forces can be greatly reduced. Further, to minimize the actuators force vector it is necessary to present the values of force vector using a single number.

Vector norms are generally used to represent vectors in a single value. Three types of vector norms are generally used namely, $l$-norm, $2$-norm or $\infty$-norm. Within these three norms, $2$-norm or Euclidean norm is more preferred [102] owing to its sensitivity towards changes in the larger force components. The $l$-norm is equally sensitive to all the force components whereas $\infty$-norm is only sensitive to the changes in the largest force component. In the present study, $2$-norm of the actuator forces has been considered which is given by the minimum norm solution of (5.13) and can be written as the left pseudo inverse (5.14).

The limits of the actuator forces can be determined using singular value decomposition theorem and are given as following.

$$
\frac{\|M_{\text{ext}}\|}{\sigma_1} \leq \|F\| \leq \frac{\|M_{\text{ext}}\|}{\sigma_3}
$$

Referring to (5.3&5.23), it is interesting to note that the magnitude of the upper limit of the actuator forces is governed by the minimum singular value $\sigma_3$ and the forces along the actuators can be reduced by maximizing this value. However, when the condition number is minimized, it is possible that the minimum singular value also gets increased and as a consequence the actuator forces may reduce. However this is an assumption which can only be verified from experiments. This aspect is further discussed in the Chapter 5 while analyzing experimental results.

In the light of above discussion, Euclidean norm of the four actuator forces, averaged over the workspace points is considered as one of the objective functions to minimize.

5.2.4 Maximum Robot Actuator Forces

Even if the norm of forces is small, there’s an apprehension that an individual actuator force may exceed the permissible limits posed by PMA. Thus it is desired that the maximum actuator force be also minimized along with the norm of actuator forces. The maximum actuator force in an individual link is given by (5.24).
5.3 Structural Design

Stiffness was a major concern while designing the wearable robot, which is being actuated by flexible and compliant PMA. Though compliance is a desirable feature for the wearable robot, stiffness is essential for positional accuracy and stability. It is discussed in the following subsections that the robot’s stiffness which is governed by actuator stiffnesses, is a function of its geometry defined by the placement of links and actuators. Owing to the inherent flexibility in the actuators and cables, it was also desired to evaluate the rigidity of the robot which can be defined using Eigen frequencies of the robot stiffness matrix. Robot’s stiffness matrix and its rigidity aspects are further elaborated in the next subsections.

5.3.1 Robot Stiffness Conditioning Index

A robot actuated by cables is said to be stabilizable if the stiffness matrix of the robot is positive definite under any arbitrary external wrench [165]. Stabilizability is a condition which guarantees the stability of the robot in any circumstances under sufficient antagonistic forces. The stabilizability is necessary to be investigated since antagonistic forces from PMA are used in the proposed robot to actuate the end-effector. This has been reported in the literature that cable driven robots exhibit two types of stiffness namely, active stiffness and structural stiffness [140]. The active stiffness is produced by the internal forces of the cables and the structural stiffness comes from the elasticity and the stiffness of the actuation system. Apparently, the effect of internal forces on the stiffness can be ignored since the elongation of the cables due to the internal forces is insignificant. Thus in the present application, the structural stiffness matrix consisting of two components i.e. cables, and PMA has been analyzed.

\[ K = J^T SJ \]

Where \[ S = \frac{dF}{dl} = diag(k_1, k_2, k_3, k_4) \]

The overall stiffness matrix of the robot, from its actuator stiffness’s, is computed using the robot’s Jacobian matrix (J) as shown above. Here \( K \) is the total stiffness of the robot, \( F \) is the cable force vector, \( l \) is the vector of link lengths consisting of cables and PMA and \( ‘k_1.. k_4’ \) are the stiffnesses of the individual links.
5.4 Chapter Summary

Link stiffness here represents two stiffnesses in series, the elastic stiffness of the cable and the stiffness of the PMA. Since both are in series, the resultant stiffness is mostly dominated by the less stiff member which is PMA.

\[ k = \frac{3P_g L}{2\pi n^2} \]  

(5.26)

This relation can be further written in terms of actuator force as:

\[ k = \frac{6F}{3L - \frac{b^2}{L}} \]  

(5.27)

Here \( P_g \) is the internal gauge pressure, \( L \) is the length of the PMA, \( n \) is the number of turns for a single thread of the mesh of PMA, and \( b \) is the thread angle of the PMA mesh. This is an approximate model and has been used in the present research; however, value of \( b \) and therefore stiffness may vary depending upon the type of constituent elements of the PMA. The overall stiffness matrix \( K \) can be resolved in three matrices using singular value decomposition as shown in (5.28).

\[
[K]_{3 \times 3} = [X^T]_{3 \times 3} [\Sigma]_{3 \times 3} [Y]_{3 \times 3}
\]  

(5.28)

\[ K_{\text{min}} = \min \left( \text{diag}(\Sigma_1, \Sigma_2, \Sigma_3) \right) \]  

(5.29)

Once again \( X \) and \( Y \) are orthogonal matrices and \( \Sigma \) is a diagonal matrix of three singular values as \( \left( \Sigma_1, \Sigma_2, \Sigma_3 \right) \). The minimum of these diagonal values is the minimum value of actuator stiffness (5.29) which has been considered as an objective to be maximized in the present work.

Apart from being stiff, the robot structure should be sufficiently rigid to carry out various ankle rehabilitation exercises. The robot should be able to respond quickly to the real time input from the system dynamics and the interaction dynamics. Apparently, the rigidity is closely related to the open-loop natural frequencies of the robot. The natural frequencies \( (\omega) \) are obtained by solving the generalized Eigen value problem considering the mass matrix of the robot and the stiffness matrix (5.30).

\[ \omega = \sqrt{\text{diag}(K(M^{-1}))} \]  

(5.30)

5.4 Chapter Summary

Performance indices (PIs), which are important in context of the wearable robot design, were defined in this Chapter. Dependence of these PIs on condition number and robot
geometry was explained and illustrated. It was mentioned that all these PIs can be controlled by selecting appropriate geometrical parameters for the robot. These PIs were studied by classifying them into three design aspects namely, kinematic design, actuation design and structural design to evaluate controllability and other desired attributes of the wearable robot. A brief description of each of the PI was provided along with its mathematical formulation.
Chapter 6 Single Objective Design Optimization of the Wearable Ankle Robot

Investigation of the PIs, identified for wearable robot design evaluation, revealed that all the PIs are connected to the condition number of the robot’s Jacobian matrix and thus condition number is an important objective to be optimized. It is expected that by minimizing the condition number, the Jacobian singularities can be reduced which will improve the feasible workspace. Actuator forces are likely to decrease and stiffness of the robot is expected to increase as a consequence of condition number optimization. Therefore, the design optimization was performed by considering condition number as the sole objective or performance criterion. To begin with, Robot’s geometrical parameters, which define robot design, have been discussed in this Chapter along with the requisite kinematic constraints. The MGA methodology employed for the optimization has been explained with regards to the implementation. Results obtained from single objective optimization strategy have been analyzed to draw motivations for implementing a multi-objective optimization in the present design optimization problem.

6.1 Robot Geometrical Parameters

The study of the robot design in the present work has been fundamentally the analysis of its geometry, which is mainly defined by the configuration of actuator connection points on the two platforms and the separation between the platforms. Effective shapes and sizes of the platforms are decided by these connection points and the height of robot is governed by the distance between the platforms. It has been theoretically substantiated (3.20) that the robot’s constructional configuration affects its Jacobian matrix which in turn influences its condition number and related performance indices [94].

The geometrical design parameters of the robot have been considered as the optimization variables and are illustrated in Figure 6.1. Different robot designs can be obtained by varying the polar coordinates of the connection points on both the platforms. The objective is to find the best design from the infinite number of possible constellations of actuator connection points. The feasible area on the platforms is the region which is available after applying the requisite constraints given by (6.1&6.2).
It is apparent that the solution space, after applying the constraints, is a continuous region which can be investigated by changing the polar coordinates. The limiting values of parameters $q_1, \ldots, q_8$ (6.1) have been carefully decided by leaving sufficient space for patients to conveniently place their foot on the end-effector. A total of eight design parameters have been selected consisting of the polar coordinates of actuator connection points (6.1) on both the platforms (Figure 6.1). Distance between the platforms has not been changed and is kept as 130 mm, which is the expected distance of ankle joint from the foot base. Owing to the limitations arising from the use of ankle robot by variety of subjects, following geometrical constraints have been considered. The limits on geometrical constraints are in millimetres and radians (6.1&6.2).

$$\frac{\pi}{12} \leq q_1, q_3 \leq \frac{\pi}{3} \quad \text{and} \quad \frac{\pi}{12} \leq q_5, q_7 \leq \frac{4\pi}{9}$$

$$80 \leq q_2, q_6, q_8 \leq 120 \quad \text{and} \quad 120 \leq q_4 \leq 160$$

Additionally, actuator stroke length constraint (6.2) for the PMA has also been considered.

$$\max(l_p^0) - \min(l_p^0) \leq 110 \quad \text{for} \quad p = 1, \ldots, 4.$$  \hspace{1cm} (6.2)

Here, $l_p^0$ is the length of $p^{th}$ cable of the robot and $q_i$'s are the geometrical parameters shown in Figure 6.1.
6.2 Optimization of Global Condition Number (GCN)

This has been emphasized and illustrated with simulation results in the previous Chapter (Section 5.1.2) that the Jacobian matrix of the robot, which relates the joint and Cartesian rates, is required to be well conditioned. If the Jacobian matrix is ill-conditioned, small changes in the joint variables will result in very large changes in the Cartesian variables. This will further lead to difficulty in end-effector control and build-up of large trajectory following errors. Nevertheless, by choosing optimal geometric parameters of the robot, the condition number of robot’s Jacobian matrix can be improved and the design can be made robust to the small parametric variations.

It has also been substantiated in the last Chapter that the GCN is an important criterion for the design optimization of the proposed robot. Optimization of GCN shall yield a robot workspace which is free from singularities and shall also reduce the actuator force requirements (Section 5.2.3).

While considering optimization of GCN, it is desired to obtain an optimal configuration of the actuator-attachment points on both fixed and moving platforms. The proposed robot has four actuators and thus has eight connection points on the platforms. Since all the connection points on a platform are coplanar, each of them can be defined by two parameters, yielding 16 parameters in all. The robot is to be used for both right and left foot ankle treatments hence it should have left-right symmetry. In other words the actuator-attachment points on the right half of the platform should be a mirror replica of the connection points on the left half of the platform. Consequently, the number of independent parameters reduces to half i.e. instead of 16 only 8 parameters need to be considered.

After deciding for the design parameters a suitable optimization algorithm is investigated. Previously, gradient based or numerical optimization methods have been used for robot design optimization [97, 166]. These methods require a near optimal initial solution and in the absence of a good initial solution, the algorithm may converge to a local sub-optimal solution. Moreover, these methods become less efficient when the search space is large and is finely discretized or continuous. It has been observed that when the objective function does not change significantly in the close vicinity of solution space, numerical methods become less effective [104].

In view of above assertions, application of GA has been recommended in the present optimization problem because GA works with population of solutions and processes them simultaneously hence is more likely to give a global optimal solution. However, GA is a
discrete optimization algorithm and there is an apprehension of loss of information once the workspace is discretized. Nevertheless, the simulation results (Figure 6.2-Figure 6.4) for the condition number distribution in the workspace clearly reveal that the values of condition number are quite similar in the neighbourhood of any given point and the variation is very smooth. Thus discretizing the workspace does not lead to any information loss and this fact further supports the choice of GA. While carrying out the optimization of TS fuzzy models using GA in Section 4.4.3, it was emphasized that the local search capabilities of GA are not as good as its global search capabilities. Consequently, MGA methodology was proposed in order to improve performance of simple GA. The MGA has been once again used in the optimization of GCN. Mathematical formulation of the fitness function for the MGA has been discussed next followed by the detailed description of the optimization methodology.

6.2.1 Mathematical Formulation

Diverse design concepts have been obtained by varying the geometrical parameters of the wearable robot. For each instance of robot design, a GCN is computed which is an average feature of the condition number in the entire robot workspace. The main objective is to obtain a robot design which exhibits near unity value for GCN. However, to ensure that there are no undesirable peak values of the condition number in the robot workspace; the maximum value of the condition number in the entire workspace can also be obtained and minimized. This maximum value of condition number \( P \) has been included in the objective function as a penalty term (6.3). The inherent mechanism of the GA maximizes a given objective function hence for present case of minimization of GCN, a special fitness function (6.3), converting the problem into a maximization goal, has been used.

The optimization problem is formulated as explained below.

\[
Maximize
\]
\[
Fitness \ (i) = \frac{1}{1 + f(q_1, q_2, \ldots, q_8)}
\]
\[
Where
f(q_1, q_2, \ldots, q_8) = GCN(i) + P(i)
\]

Mathematically \( GCN \) and \( P \) are further defined as
\[
GCN(i) = \frac{\sum_{j=1}^{n} k(i, j)}{n} \quad for \quad i = 1, \ldots, m
\]
6.2 Optimization of Global Condition Number (GCN)

Here $k(i, j)$ is the condition number of $i^{th}$ solution (since GA works with a population of solutions) at $j^{th}$ workspace point and $P(i)$ is a penalty term described as below. Here $i$ is the number of binary string in GA representing number of robot design solutions and $j \in W$ or is a workspace point. As defined earlier, the condition number is the ratio of maximum singular value $\sigma_{\text{max}}(J(i,j))$ to the minimum singular value $\sigma_{\text{min}}(J(i,j))$ of the Jacobian matrix (5.3).

$$P(i) = \max \left( \frac{\sigma_{\text{max}}(J(i,j))}{\sigma_{\text{min}}(J(i,j))} \right) \quad \text{for} \quad j = 1, ..., n$$ \hfill (6.5)

The optimization is carried out abiding by the geometrical constraints mentioned above (6.1 & 6.2). Limits on the geometrical parameters of moving platform have been chosen to allow patient’s foot on it without interfering with the cables. Distance between the two parallel platforms (which is the height of ankle joint above moving platform), is kept as 130 mm and has not been varied. However the link lengths are not constant as they are altered when the connection point positions are changed on the platforms.

6.2.2 Modified Genetic Algorithm Methodology

Modified Genetic Algorithm has been explained previously in the Section 4.4.3 of Chapter 4, while solving the forward kinematic problem. Nevertheless the implementation details of MGA i.e. the fitness function evaluation and details of genetic operators namely, generation, selection and crossover have been discussed here in context to the design optimization.

**Generation**

Initial population of 100 binary coded strings of 96 bits was generated using Knuth’s random number generator [167]. The binary string of 96 bits has 12 bits assigned for each of the eight parameters, thus the solution space can be discretized with accuracies of the order of $10^{-4}$. An example binary string representing a robot conceptual design is shown below.

\[ q_1^{1101.1} \quad q_2^{1010.0} \quad q_3^{0110.1} \quad q_4^{1011.0} \quad q_5^{1010.1} \quad q_6^{0101.0} \quad q_7^{1010.0} \quad q_8^{1001.0} \]

**Fitness Evaluation**

In order to evaluate each string in the population, its fitness must be calculated using (6.3). The binary strings are decoded to real numbers to obtain the robot design parameters and the values of GCN and the penalty term for each string (each string corresponds to a robot design) is evaluated from the feasible workspace to further determine the values of the fitness function.
6.2 Optimization of Global Condition Number (GCN)

Reproduction

The population initially selected may not have all the good strings and to select good strings to form the mating pool, two kinds of reproduction operators have been used. Initially all 100 binary strings are evaluated for their fitness values. If the maximum fitness value is less than an assumed value of 0.10, a roulette-wheel selection method [104] is used and multiple copies of strings in proportion to their individual fitness are selected in the mating pool. When the maximum fitness is found to be more than 0.10, gradient based selection method is used to get the best solution in the generation stored in the mating pool. The gradient values of the binary strings are calculated using the central difference method [104] as explained in the Section 4.4.3. The limiting number 0.10 has been chosen, based on the results from several simulation runs. Simple GA working on the roulette-wheel selection scheme stops to further increase the fitness function after achieving a value of 0.10. This number corresponds to a combined value of GCN and maximum condition number as 9.

Crossover and Mutation

A four point crossover with probability of 0.95 has been used and mutation has been performed with 0.01 probabilities. The crossover operator of GA is responsible for the local search and its probability is normally kept high. Whereas the mutation operator prevents the algorithm from converging into a local optimum solution and its probability is generally kept low. Higher mutation probability results in an undesired random search in the solution space [154] and hence is avoided. Various steps used in the algorithm are explained below.

Step1. Select a termination criteria based on either the number of epochs or the accuracy obtained. In the present case a distance function has been used which is defined as 

\[ d = (0.33 - \text{Fitness}) \]

Ideally both GCN and \( P \) should be equal to unity and the fitness function (6.3) should be equal to 0.33. The distance function indicates, as how far the condition number is from its ideal value. The algorithm terminates if the distance function is 0.15 (this would mean that the sum of GCN and the maximum condition number value is 4.56) or the number of iteration is 100.

Step2. Initialize a random population of 100 binary strings with 96 bits in each binary. There are eight design parameters in the present problem and each of them are represented by twelve bits thus every binary corresponds to an individual robot design.

Step3. Convert binary values of eight design parameters to the decimal values taking their universe of discourse into account as shown below.

\[ \text{for } i = 1 \ldots B \]
6.3 Results and Discussion

\[ q_i = q_i^l + \frac{q_i^u - q_i^l}{2i - 1} \times (\text{decoded value of } i^{\text{th}} \text{eighth part of the binary string}) \]

Here \( q_i^u \) and \( q_i^l \) are the upper and the lower limits of the design variables, \( l_i \) is the length of the eighth part of the binary string which is 12 in the present case.

Step 4. Calculate the fitness function for each robot design (6.3) from the decimal values of the parameters obtained in step 3. Find out the maximum fitness value and compare it with a predefined value of 0.10 to make a decision on the type of selection operator to be used.

Step 6. Use roulette wheel selection method when the maximum fitness value is less than 0.10, otherwise switch to the gradient based selection approach. Create multiple copies of binaries in the mating pool, based on their individual fitness or gradient.

Step 7. Perform crossover operator on randomly selected parents and later perform mutation operator with specified probability on each bit of the crossed-over binary.

Step 8. Check for the termination criteria, if not met, go to Step 3 or else terminate.

6.3 Results and Discussion

The orientation workspace was created by rotating the moving platform along \( x, y \) and \( z \)-axes through incremental rotations within the ranges of ±25°, ±40° and ±30° respectively.

Initially, the robot design was conceptualized as shown in Figure 6.5 and its performance indices were evaluated. Global condition number was found to be 14.99 which is a very high value. Similarly, the actuator force norm, which has been averaged over the entire workspace and is shown in the Table 6.1, was also found to be undesirable. These results from an intuitive design also justify the need for the design optimization of the wearable robot.

To begin with, the design optimization was carried out using simple GA wherein the roulette-wheel selection method was used. Several runs of the algorithm were found to converge to the same minimum GCN value of 3.16, further reduction in the GCN however, was not observed. Apparently this optimum GCN value was quite an improvement compared to the GCN value of 14.99 from the initial design. Number of iterations to achieve the optimum value of GCN varied between the experiments and the minimum number of iterations was found to be 22. The average actuator force norm in the entire workspace for this design was calculated to be 669.09 N, in order to produce 30 Nm external torque on the end-effector.
6.3 Results and Discussion

Afterwards, using modified GA, wherein the gradient based selection scheme is employed, the GCN value further reduced to 2.06 which is a visible improvement and indicates that the simple GA was not capable of fine tuning results to the global optimal solution. The three design solutions i.e. the initial intuitive design, GA-based design and the modified GA-based design have been summarised in Table 6.1. Other performance indices have also been provided for the comparison of design alternatives.
6.3 Results and Discussion

While using the modified GA, owing to its probabilistic nature, the switching between the two selection operators occurred at different number of iterations for different experiments. The results for the actuator forces displayed in Table 6.1 also support the assertion presented in Section 5.2.3 regarding the actuator forces minimization. The average actuator force norm has been drastically decreased from 920.29 N in the initial design to 645.19 N in the final design after optimization of the GCN using MGA. Clearly, with the condition number, the actuator force requirements are also lowered.

![Figure 6.4: Condition number versus end-effector orientations (radians) at $\psi = 20^\circ$](image)

![Figure 6.5: Initial geometrical parameters of the moving platform (a) and the FP (b)](image)
6.3 Results and Discussion

Figure 6.6: Condition number (K) distribution in the robot workspace after GA-based optimization

Figure 6.7: Condition number (K) distribution in the robot workspace obtained using modified GA
The average actuator force norm of 645.19 N can be achieved from 30cm long PMA used in the wearable robot prototype. The feasible workspace of the optimum robot configuration was once again checked for condition number analysis. Analyzing the results from the final design, it was found that the range of condition number variation in the entire feasible workspace was from 1.55 to 2.77, which is within the permissible limits [97]. This also indicates that the robot configuration obtained is stable and has better manipulability in its feasible workspace for given range of orientations. The condition number is further analyzed at different workspace points (represented by a set of three Euler angles) and the results are plotted for three fixed yaw orientations ($\psi$), as shown in Figure 6.2-Figure 6.4. These results show that the condition number varies smoothly in its vicinity and does not have sudden peak values. Further, to provide a more comprehensive picture of the condition number distribution in the entire workspace, results from simple GA and modified GA are provided in three dimensional plots shown in Figure 6.6 and Figure 6.7 respectively. These results once again confirm that the condition number is well within desired limits in the entire feasible workspace. However higher values of condition number can be observed in the extremities of the robot workspace. The final geometrical parameters obtained as a result of the optimization are shown in Figure 6.8. Analyzing the results for the geometrical parameters it is revealed that the actuator-attachment points on the moving platform are placed in a manner to comfortably accommodate patient’s foot.
### 6.3 Results and Discussion

Table 6.1: Comparison of robot designs obtained using single objective optimization approach (geometrical dimensions are in degrees and mm)

<table>
<thead>
<tr>
<th></th>
<th>q1</th>
<th>q2</th>
<th>q3</th>
<th>q4</th>
<th>q5</th>
<th>q6</th>
<th>q7</th>
<th>q8</th>
<th>GCN</th>
<th>Actuator Force Norm (N)</th>
<th>Workspace Utilization (%)</th>
<th>Stiffness (Nm⁻¹)</th>
<th>Error in moments (Nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Design</td>
<td>45°</td>
<td>100</td>
<td>25°</td>
<td>150</td>
<td>25°</td>
<td>100</td>
<td>45°</td>
<td>100</td>
<td>14.99</td>
<td>920.29</td>
<td>87.58</td>
<td>42</td>
<td>1.08</td>
</tr>
<tr>
<td>GA-based Design</td>
<td>46°</td>
<td>96</td>
<td>22°</td>
<td>150</td>
<td>66°</td>
<td>97</td>
<td>60°</td>
<td>100</td>
<td>3.16</td>
<td>669.09</td>
<td>94.13</td>
<td>191</td>
<td>2.31E-8</td>
</tr>
<tr>
<td>Modified GA-based Design</td>
<td>41°</td>
<td>99</td>
<td>22°</td>
<td>159</td>
<td>68°</td>
<td>98</td>
<td>58°</td>
<td>91</td>
<td>2.06</td>
<td>645.19</td>
<td>95.32</td>
<td>232</td>
<td>1.37E-14</td>
</tr>
</tbody>
</table>

$q_1, \ldots, q_8$ are the geometrical parameters of wearable robot and GCN is the Global Condition Number.
6.4 Chapter Summary

Initially, design optimization of the wearable robot was considered as a single objective optimization problem and discussed accordingly in this Chapter. Robot’s geometrical parameters were explained along with the requisite kinematic constraints on these parameters. Optimization of the GCN was emphasized in terms of its significance to other performance indices. Since GA can only maximize an objective function, the GCN and the maximum condition number indices were converted into a suitable fitness function. Subsequently, MGA was discussed in context to the present problem stating its termination criterion and other important parameters.

Results from an intuitive design, a GA-optimized design and finally a MGA-optimized design were tabulated and discussed. Measure of GCN and its distribution in the workspace was found to be satisfactory for the MGA optimized design. However higher actuator force norm, reduced feasible workspace and low stiffness were still a matter of concern. It was found that the single objective optimization approach cannot provide a best compromised solution with regard to all the performance criteria. Moreover using a single objective approach, it is difficult to establish a trade-off between criteria which are of conflicting nature, such as the actuator force norm versus workspace and tensionability versus condition number. These facts supported the decision for using of a multi-objective optimization approach in the present case. Subsequently, various multi-objective optimization approaches are reviewed and a brief discussion on their feasibility in the present case is provided in the following Chapter.
Chapter 7 Multi-objective Design Optimization of the Wearable Ankle Robot

The condition number distribution, in the workspace of wearable ankle robot, was largely improved as a consequence of single objective optimization of GCN. However, there was not much improvement in the performance indices, such as actuator force norm, the workspace utilization index and the stiffness. Consequent upon a brief analysis of the single objective optimization results it was found that criteria such as workspace and tensionability conflict with actuator forces and condition number. When objectives conflict they cannot be collectively optimized using a single objective approach. Moreover in such cases, it is difficult to find a single design solution which can offer optimal values for all the objectives. Therefore, multi-objective optimization methods need to be investigated to obtain a set of equitable solutions which may provide different grades and blends of trade-offs between the objectives. Subsequently, a decision on design selection can be made by analyzing this set of equitable solution and considering the importance of individual objectives and severity of the solutions. Therefore, multi-objective optimization methods were reviewed and to begin with, the preference based approach which is frequently used in design optimizations was implemented in the present design optimization problem. Later, to explore further improvements in the values of PIs, a multi-objective evolutionary approach called NSGA II was also employed.

It was noted that existing evolutionary algorithms (EA) become inefficient when number of objectives is large and hence a new concept of fuzzy dominance as an alternative to the existing non-dominance criterion of EA is proposed. Two selection approaches based on fuzzy dominance namely, equitable fuzzy sorting genetic algorithms (EFSGA) and biased fuzzy sorting genetic algorithms (BFSGA) have been proposed in this work to address various shortcomings of existing EA approach. Design optimization of the wearable robot was performed using existing EA and the proposed fuzzy dominance based approaches. Analyzing the results from above methods it was found that while EFSGA exhibits strong discrimination abilities and can provide optimal solutions at the desired level set by the end user, the BFSGA approach performs well in exploring the extreme zones of the Pareto front.
7.1 Multi-Objective Optimization Approach

Real world optimization problems normally involve more than one objective wherein the optimization goal is to find one or more optimal solutions. Different solutions provide diverse conflicting scenarios or tradeoffs for dissimilar objectives. A particular solution which provides best value of an objective may offer a compromised value for the other objective. The optimization goal in such case is to obtain a cluster of solutions providing various grades and blends of the objective values. By grade and blends it is meant that different weights of objectives are grouped in different manners e.g. \((0.1f_1, 0.9f_2)\) and \((0.9f_1, 0.1f_2)\). Following this approach a Pareto front can be obtained as shown in Figure 7.1 which has solutions providing different grades and combinations of the two objectives.

Figure 7.1: Pareto front solutions for two-objective optimization problem

There are basically two approaches to obtain a Pareto front namely, *preference based multi-objective optimization* and *EA based multi-objective optimization*. While using the preference based approach a large number of optimization run would be required to construct the Pareto front, whereas the EA approach is able to provide a Pareto front in a single run of the optimization. Multi-objective optimization approaches are discussed further in context to their possible implementation in the present optimization problem of wearable ankle rehabilitation robot.
7.2 Existing Approaches for Multi-Objective Optimization

7.2.1 Preference Based Multi-Objective Optimization

Conventionally, to solve multi-objective optimization problems, they are transformed into a single-objective problem. This is usually achieved by assigning a numerical preference index to each objective (performance index) and then combining the values of the preferences into a single value by either adding or multiplying all the weighted criteria [108]. That is, the value $V$ of a given candidate solution is typically given by one of the two kinds of formula as shown (7.1 & 7.2).

\[ V = w_1 \times f_1 + w_2 \times f_2 + \cdots + w_n \times f_n \]  

(7.1)

\[ V = f_1^{w_1} \times f_2^{w_2} \cdots \times f_n^{w_n} \]  

(7.2)

Here $w_n$ is the preference assigned to function $f_n$ and $n$ is the number of evaluation criteria. This approach has several inherent advantages, such as; the significance of one objective over the rest can be regulated using appropriate preferences. Owing to its simplicity in implementation, this approach is very popular among designers. However, it has certain shortcomings and worse of them is the ad-hoc selection of preferences for dissimilar objectives [110]. Normally, the selection of preferences is either based on trial and error experiments or on the perceptive judgment of the end user. Such preference selection are subjective and do not have a logical base. Moreover, dissimilar quality measures of different units and scales are added or multiplied in a single objective function which is not correct mathematically.

7.2.2 Evolutionary Algorithms (EA)

The fundamental concept of the EA is that, instead of converting a multi-objective problem into a single-objective function, all the objectives are evaluated concurrently. As has been discussed earlier, EA works with population of prospective solutions and applies genetic reproduction (selection) and regeneration operators (crossover and mutation) to provide a possible global optimal solution. Quite a few EA based Pareto approaches are proposed by researchers and a comprehensive review of these can be found in [109, 110]. Nevertheless, the Pareto approaches have also been criticized for the indiscrimination in dealing with
higher dimension problems and subjectivity in selecting the best solution from the final Pareto optimal solution set.

7.2.3 Vector Evaluated Genetic Algorithm (VEGA)

This approach works with a population of solutions and proceeds similar to EA without using the idea of Pareto optimality [168]. Despite its similarity to the simple genetic algorithms, it uses a different selection process which is adapted to perform multi-objective optimization. Offspring (next generation binary solutions) for each of the objective functions are selected independently after evaluating their individual fitness. Later, these offspring are combined and regeneration genetic operators such as crossover and mutation are performed on them. Thus this approach draws benefits over preference based approaches such that the non-commensurable criteria are processed separately with different offspring or subpopulation. Nevertheless this approach is difficult to implement compared to the preference based method. Yet another shortcoming of this approach is the exclusion of Pareto dominance concept in the selection process. Consequently a design which provides a good trade-off solution for all the objectives but is unable to provide best solution for one of them is rejected.

7.2.4 Pareto Set Pursuing (PSP) Approach

A new approach based on direct sampling to approximate the Pareto frontier has been recently proposed [169]. The algorithm initializes random samples and draws samples iteratively which are closer to the true Pareto frontier. The solutions are sampled close to the Pareto frontier or right on it provided the sampling trend continues. It has been mentioned that the approach is intended for expensive black box functions only and may not be feasible for engineering design optimization problems [169]. However since this approach is truly based on sampling it does not guarantee a globally optimal solution.

In the present work of design optimization, two of the most popular methods namely, preference based method and EA based method were implemented. Six important performance indices (PIs) were selected to evaluate the robot designs. These PIs were GCN, robot workspace, norm of actuator forces averaged over the workspace, maximum actuator force, tensionability and robot stiffness. Definition and mathematical formulation of these PIs has been discussed in Chapter 5. In the following Sections, the optimization problem formulation, implementation strategies and the optimization outcomes of the two approaches have been discussed.
7.3 Preference Based Multi-Objective Optimization

Initially the conventional approach of representing multiple objectives in a single objective using the summation approach mentioned above (7.1) was investigated. First of all, the objectives were normalized as shown in (7.3), wherein the normalization of GCN, which is one of the objectives, has been shown.

\[
GCN_{\text{normalized}} = \frac{G_{C_{N_i}} - G_{C_{N_{\text{min}}}}}{G_{C_{N_{\text{max}}}} - G_{C_{N_{\text{min}}}}} \tag{7.3}
\]

Likewise values of all the six objectives were normalized to lie in the limiting values of 0 and 1. Later, each of the objectives is assigned some weighting factor (7.1) ensuring that the sum of weights for the overall optimization objective is equal to one. Thus a fitness function similar to (7.1) is obtained which has limiting values between 0 and 1. Genetic algorithms are used to perform the optimization due to its obvious advantages over numerical methods mentioned in Section 4.4.3. Genetic algorithm due to its inherent mechanism maximizes objectives and hence all the objectives are transformed into maximization functions (by inverting objectives intended to be minimized). Apparently maximum value of the fitness function (7.1) can be one, thus it is decided to terminate the algorithm when the maximum fitness is achieved. Optimization is performed taking different sets of priorities for the objectives and the results obtained are displayed in the Table 7.1. Here case I to VI refer to instances wherein individual objectives are preferred over rest and case VII indicates results when all the objectives are given equal priorities. Results for the best objective values are shown in the Table 7.1 with bold face; these values have been obtained when the corresponding objectives were preferred. Perceptibly best individual objective values, at the cost of other objectives, are obtained when they are given more preference. These results are discussed once again while comparing with the results from the EA.

As explained above, the preference based approach is simple to implement and provides a mean to regulate preferences of objectives as per user’s requirements. However in some instances where it is difficult to decide the priorities among objectives, this approach cannot be recommended. In the present case of design optimization the objectives are interdependent and conflicting, hence it is not possible to decide the vector of weights or the preferences for individual objectives. The arbitrary chosen weight vector may not be a good choice. Therefore, a multi-objective optimization approach which evolves the entire set of objectives equally and simultaneously is explored in the next step.
7.4 Multi-Objective Evolutionary Optimization Using NSGA II

Multi-objective optimization using evolutionary algorithms (MOEA) is an approach which optimizes multiple objectives concurrently without costing on individual criterion. Many variants of MOEA, which are based on the Pareto front approach, have been proposed by researchers such as NSGA (non-dominated sorting genetic algorithm), NPGA (Niched Pareto genetic algorithm), SPEA (strength Pareto evolutionary algorithm) and MOMGA (multi-objective messy genetic algorithm) [110]. Despite their different strategies, these algorithms essentially work with population of solutions and their inherent mechanism of evolution emulates the natural evolution. The evolution mechanism further facilitates exploration of various trade-off solutions with different grades and blends of objectives. Moreover EA does not require derivatives of objective functions and has robust operators such as reproduction and regeneration to avoid convergence to local optima. Since last two decades MOEA are used successfully in variety of applications as discussed in Chapter 2 while surveying the literature concerning multi-objective optimization. The most popular of the MOEA is the non-dominated sorting algorithm (NSGA II) which is the most efficient optimization routine as maintained by researchers [170]. Consequently the NSGA II is implemented and investigated for its significance in the present research. As discussed above, NSGA II works with the randomly generated initial population of solutions and emulates the natural evolution process using operators namely, selection, crossover, mutation and crowding distance. The selection or reproduction operator facilitates competent solutions and creates their multiple copies while eliminating less competent solutions from the mating pool of solutions. Crossover operator is responsible for creation of new solutions by combining good substrings from parent populations. To further improve solutions a local search is performed using mutation operator. If the mutation is carried out at the first place of the binary solution, the mutation operator may also help in maintaining diversity in the population by changing solutions values to a large extent. Crowding distance is another operator used to maintain diversity among solutions by supporting distant solutions. The scheme of NSGA II is further explained in the following steps.

1. [Start] Select a termination criterion based on number of iterations.
2. [Initialize] Initialize a random population of Q chromosomes/binaries (suitable solutions for the problem).
3. [Fitness] Evaluate the values of various objectives for each chromosome in the population.
4. **[Rank]** Classify population into fronts using non-dominating sorting algorithm [109] and assign non-domination ranks to each solution.

5. **[Offspring]** For the first iteration, create a duplicate copy of the parent population by randomly arranging its solutions and call this offspring population.

6. **[Selection]** Combine offspring with the parent population and perform Tournament Selection [109] on the combination (2Q) to select Q solutions based on non-dominance rank. Use *crowding distance operator* for instances when two solutions carry equal non-dominance rank. Call the resulting population as parents.

7. **[Crossover]** Cross over the parents with a crossover probability to form new offspring.

8. **[Mutation]** Mutate the new offspring at each bit with a chosen mutation probability.

9. **[Rank]** Evaluate solutions for their objective values to perform non-dominated sorting and once again classify the population into fronts based on their non-domination ranks.

10. **[Check]** Check for the termination criteria and stop if this is achieved else go to Step 6.

11. **[End]** Stop and return the final population when the termination criterion is reached.

### 7.4.1 Experimental results from NSGA II

Design optimization was carried out to evaluate wearable robot design based on its important performance criteria discussed in the Section 7.2.4. NSGA II algorithm was implemented using the set of parameters shown below.

- Population size: 1000
- Crossover probability: 0.95
- Real-parameter mutation probability: 0.05
- Distribution index for crossover: 10
- Distribution index for mutation: 50

Twelve binary bits were used to represent each of the design parameters, thus the total length of a binary, representing a design solution with eight parameters was 96 bits. Standard NSGA II algorithm described in [109] was used wherein binary tournament selection operator was used to pick solutions from the combined population of parents and offspring solutions based on their non-dominating ranks.
Table 7.1: Results from preference based multi-objective optimization

<table>
<thead>
<tr>
<th>Priorities</th>
<th>Global Condition Number</th>
<th>Workspace Utilization (%)</th>
<th>Stiffness (N/m)</th>
<th>Max. Force Norm (N)</th>
<th>Normalized Moment Error (Nm)</th>
<th>Max. Actuator Force (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>w₁ = 0.5; w₂ = 0.1; w₃ = 0.5; w₄ = 0.1; w₅ = 0.1; w₆ = 0.1</td>
<td>2.12</td>
<td>83.80</td>
<td>293.00</td>
<td>386.98</td>
<td>0.83</td>
</tr>
<tr>
<td>II</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>3.71</td>
<td>95.00</td>
<td>125.60</td>
<td>431.20</td>
<td>0.36</td>
</tr>
<tr>
<td>III</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>2.17</td>
<td>82.00</td>
<td>357.14</td>
<td>360.85</td>
<td>0.365</td>
</tr>
<tr>
<td>IV</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>3.29</td>
<td>85.35</td>
<td>200.00</td>
<td>349.21</td>
<td>1.25 E-14</td>
</tr>
<tr>
<td>V</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>3.10</td>
<td>85.30</td>
<td>191.30</td>
<td>360.93</td>
<td>0.30</td>
</tr>
<tr>
<td>VI</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>3.79</td>
<td>81.62</td>
<td>155.25</td>
<td>402.69</td>
<td>1</td>
</tr>
<tr>
<td>VII</td>
<td>w₁ = 0.5; w₂ = 0.5; w₃ = 0.5; w₄ = 0.5; w₅ = 0.5; w₆ = 0.5</td>
<td>2.59</td>
<td>92.05</td>
<td>231.55</td>
<td>377.59</td>
<td>0.43</td>
</tr>
</tbody>
</table>

w₁, w₂, .... w₆ are the weights or priorities for the six objectives in their respective order of placement in the Table.
7.7 Discussion on Results

Interestingly, after three iterations all the design solutions were found to be non-dominating and shared the first non-dominant front.

Figure 7.2: Results obtained from NSGA II
7.7 Discussion on Results

Procedurally, when two solutions have the same rank, the crowding distance operator is invoked and computed, the solution with larger crowding distance is selected in such case. In the present case, where all the solutions carry equal ranks, the crowding distance operator was required to be called each time to break the tie.

Since the crowding distance is calculated by evaluating objective values of solutions, calling this operator requires further function evaluations and the overall efficiency of the algorithm decreases. Moreover since the optimization is performed solely on the basis of the crowding distance measure, the algorithm becomes another kind of single optimization routine where crowding distance is considered as a kind of non-linear weight index to carry out optimization. NSGA II results after 25 iterations for three important objectives namely; the condition number, norm of actuator forces and the stiffness are displayed in Figure 7.2. Representative results for all the six objectives while using NSGA II are presented in Table 7.2 where the design parameters are shown in millimeters and degrees (Figure 6.1). It is important to note here that despite the large variation in the values of six objectives all these solutions share the first non-dominant front.

Table 7.2: Representative design optimization results from NSGA II.

<table>
<thead>
<tr>
<th>q1</th>
<th>q2</th>
<th>q3</th>
<th>q4</th>
<th>q5</th>
<th>q6</th>
<th>q7</th>
<th>Cond. number</th>
<th>Work space</th>
<th>Stiffness (N/m)</th>
<th>Actuator Force Norm (N)</th>
<th>Moment error (Nm)</th>
<th>Max Actuator Force (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49.5</td>
<td>11.7</td>
<td>48.4</td>
<td>15.4</td>
<td>68.8</td>
<td>11.6</td>
<td>67.5</td>
<td>11.7</td>
<td>2.66</td>
<td>0.91</td>
<td>312.92</td>
<td>176.21</td>
<td>1.62</td>
</tr>
<tr>
<td>39.9</td>
<td>10.6</td>
<td>39.4</td>
<td>14.4</td>
<td>71.3</td>
<td>10.6</td>
<td>65.3</td>
<td>10.4</td>
<td>2.66</td>
<td>0.97</td>
<td>236.22</td>
<td>193.99</td>
<td>2.76</td>
</tr>
<tr>
<td>55.6</td>
<td>10.8</td>
<td>39.4</td>
<td>14.3</td>
<td>54.3</td>
<td>10.7</td>
<td>66.5</td>
<td>10.4</td>
<td>3.39</td>
<td>1</td>
<td>121.20</td>
<td>223.47</td>
<td>1.31</td>
</tr>
<tr>
<td>58</td>
<td>10.5</td>
<td>42.1</td>
<td>15.8</td>
<td>54.8</td>
<td>10.5</td>
<td>55.4</td>
<td>11</td>
<td>5.27</td>
<td>1</td>
<td>62.995</td>
<td>213.55</td>
<td>1.30E-14</td>
</tr>
<tr>
<td>39.1</td>
<td>11.7</td>
<td>42.1</td>
<td>15.8</td>
<td>54.8</td>
<td>10.5</td>
<td>55.4</td>
<td>11</td>
<td>2.55</td>
<td>1</td>
<td>184.34</td>
<td>172.92</td>
<td>1.07E-14</td>
</tr>
<tr>
<td>58</td>
<td>10.5</td>
<td>59.1</td>
<td>15.7</td>
<td>64</td>
<td>11.9</td>
<td>67.5</td>
<td>11.4</td>
<td>5.23</td>
<td>0.91</td>
<td>88.32</td>
<td>195.39</td>
<td>8.88E-15</td>
</tr>
<tr>
<td>47.7</td>
<td>10.6</td>
<td>42.1</td>
<td>15.7</td>
<td>64</td>
<td>11.8</td>
<td>67.8</td>
<td>11.4</td>
<td>2.39</td>
<td>0.91</td>
<td>360.89</td>
<td>184.91</td>
<td>1.60E-14</td>
</tr>
<tr>
<td>39.6</td>
<td>11.2</td>
<td>42.1</td>
<td>15.7</td>
<td>64</td>
<td>11.8</td>
<td>67.8</td>
<td>11.4</td>
<td>2.34</td>
<td>0.94</td>
<td>362.15</td>
<td>178.17</td>
<td>1.21E-14</td>
</tr>
<tr>
<td>38.1</td>
<td>11</td>
<td>38.9</td>
<td>15.7</td>
<td>50.3</td>
<td>10.6</td>
<td>68.1</td>
<td>11.4</td>
<td>2.19</td>
<td>1</td>
<td>293.50</td>
<td>198.56</td>
<td>9.47E-15</td>
</tr>
<tr>
<td>39.5</td>
<td>10.5</td>
<td>47.7</td>
<td>15.8</td>
<td>54.8</td>
<td>10.5</td>
<td>55.4</td>
<td>11</td>
<td>3.44</td>
<td>1</td>
<td>113.45</td>
<td>184.59</td>
<td>1.56E-14</td>
</tr>
</tbody>
</table>

7.4.2 NSGA II and its Limitations

Apparently the concept of Pareto dominance is important while optimizing mutually conflicting objectives and owing to this fact, evolutionary algorithms have been used in some of the real life MOPs [170]. The concept of non-dominated Pareto front solutions proposed by [109] is used in most of the MOP’s wherein two solutions are compared and the non-
dominated one is selected. A non-dominated solution is the one which is not worse than the other solution being compared with, in terms of all the objectives and is strictly better in at least one objective.

However, there are certain issues which limit the use of NSGA II in higher dimension MOP’s and otherwise as discussed here. Firstly, when the number of objectives is large and the solution space is not discrete then the concept of crisp non-dominance loses its significance and more often fails to discriminate vital few from useful many solutions. An example shown in the Table 7.3 below explains as how using non-dominated sorting, all the solutions in the given population become non-dominated and finally belong to the same crisp non-dominated front. However, solutions one and two in the Table are far better than all other solutions, if the goal of optimization is to minimize objectives ($f_1, ..., f_6$) and should not be placed in the same front. It is shown later in the following Section that a better discrimination can be obtained using the concept of fuzzy dominance proposed for the first time in this work.

Further, during selection stage, all these solutions due to their equal front indices will invoke the crowding distance operator [109] making the algorithm inefficient. Thus it is clear that in such cases the concept of crisp non-dominance looses significance and some other criterion for discrimination among solutions is required to be investigated.

Table 7.3: Limitation of the non-dominance concept

<table>
<thead>
<tr>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
<th>Crisp ND Fronts</th>
<th>Fuzzy dominant Fronts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.60</td>
<td>2.50</td>
<td>2.00</td>
<td>2.50</td>
<td>2.30</td>
<td>2.00</td>
<td>1.00</td>
<td>2</td>
</tr>
<tr>
<td>3.00</td>
<td>2.40</td>
<td>2.00</td>
<td>3.00</td>
<td>2.60</td>
<td>3.00</td>
<td>1.00</td>
<td>3</td>
</tr>
<tr>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>2.4</td>
<td>8.00</td>
<td>5.00</td>
<td>1.00</td>
<td>10</td>
</tr>
<tr>
<td>7.00</td>
<td>4.00</td>
<td>5.00</td>
<td>8.00</td>
<td>2.20</td>
<td>7.00</td>
<td>1.00</td>
<td>11</td>
</tr>
<tr>
<td>2.58</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>1.00</td>
<td>16</td>
</tr>
</tbody>
</table>

An additional motivation for the search for an alternate of NSGA II is the lack of clear termination criterion. An optimization algorithm should terminate when either the global optima has been reached or the values of objectives in their acceptable ranges are obtained. NSGA II has no mechanism to ascertain the global optimality of final solutions obtained. This is evident from [141] wherein ten rigorous experiments are performed using NSGA II with varied algorithm parameters such as population size, number of generations and distinct crossover and mutation parameters. Solutions obtained from these experiments are further clustered and few wide spread trade-off solutions selected and further optimized using genetic local search. Once again to validate global optimality, each of the objectives is
7.7 Discussion on Results

optimized independently treating it as a single objective and compared with the NSGA II final solutions. Eventually for numerical optimization problems which do not have clear mathematical convergence proof, NSGA II becomes inefficient and cumbersome.

In most of the real world problems pertaining to MOP’s, all the objectives are not expected to have equal priorities and user may have diverse preferences for different criteria. Consequently, user-preference has been considered by researchers in the past and two basic approaches are mentioned in the literature namely, a priori method and posteriori method \[171\]. According to the priori method, user preference is accounted in the algorithm during optimization. Conversely, in the posteriori method, the algorithm is run without preferences and the user can select preferred solutions from a wider set of solutions received after the algorithm is terminated. Apparently, the first approach is efficient since computation efforts are utilized in the preferred areas instead of searching the entire solutions space. The time cost saving increases with the increased number of objective functions. Recently, research has been successfully carried out encompassing a wide range of knowledge on priori incorporation of user preferences in the selection process of EA \[13\]. NSGA II has no provisions for including information on user preferences in the algorithm and preferred solutions are selected using a posteriori approach.

Another issue concerning MOPs reported in the literature is the selection of a final solution from the assortment of Pareto optimal solutions. Though all the final Pareto front solutions are non-dominated, end user wants a unique solution which is best among the better ones. Little work has been done by researchers in the past wherein fuzzy inferencing is preponderantly used to select the best solution from the finally obtained Pareto solutions \[108, 172\].

Until now some of the above mentioned issues have been attempted by researchers in seclusion \[13, 171-177\]. However, the issues are required to be revisited and an approach be devised whereby all these issues can be effectively addressed with one methodology.

To attempt these issues, a fuzzy based genetic algorithm has been proposed in the present work for multi-objective optimization of higher dimension MOP. Previously fuzzy logic has been applied in solving multi-objective optimization problems. Fuzzy dominance has been defined in \[178, 179\] and implemented in selection of a final solution from the lot of Pareto optimal solutions. The degree of fuzzy dominance is calculated using the simple fuzzy intersection concept \[154\]. In the present work this approach has been termed as a simple fuzzy sorting based genetic algorithm (FSGA). Apart from preference based multi-objective optimization and NSGA II, the simple FSGA has also been implemented in the present
7.7 Discussion on Results

design optimization problem. Clearly illustrating the results using these approaches, problems encountered have been explained which in turn forms the basis for the motivation of the present research. Simulation results of proposed EFSGA and BFSGA clearly show that the proposed algorithms exhibit a number of practical advantages. First of all, these facilitate inclusion of user preference on objectives at the beginning of the algorithm so that the rest of the algorithm searches solutions in the preferred areas efficiently. Secondly, by defining fuzzy dominated fronts to classify intermediate populations better classification of solutions has been achieved. While due to the increased number of objectives and continuous solution space the crisp non-dominance concept is not successful, the fuzzy dominance notion has been found to be able to clearly discriminate a vital few from useful many (Table 7.3). Thirdly, as explained later in the following Section, width of a fuzzy front is decided by the user in the proposed algorithm and the number of fuzzy fronts (which depends on the fuzzification of objectives and is discussed later) is definite. Thus the termination criterion in the proposed approach is unambiguous and definite i.e. the algorithm terminates as and when solutions in the lowest front (solution space with acceptable objective values) are obtained.

Finally the proposed algorithm has an inherent process of providing certain scores to the solutions based on the fuzzy values of their objectives. Interestingly these scores, since they are unique, can be used to select the best solution from the final Pareto front solutions. The proposed algorithm is further explained in the following Section.

7.5 New Fuzzy Based Sorting Genetic Algorithms

Fuzzy set theory was given by Prof Lotfi Zadeh in his seminal paper [180] wherein qualitative numbers can be treated vividly with conventional mathematical operators. Lately, fuzzy logic has become a popular heuristic approach in modelling non-linear, uncertain and ambiguous systems [157].

As has been discussed in the previous Section, there’s no provision in NSGA II whereby user’s preferences for various objective functions can be included. However user can select a solution of his choice and preferences from the large set of final Pareto optimal solutions. An argument can be raised in this context is that if user preferences are included at the beginning of the optimization; user shall have a larger solution set to choose from. Furthermore user would be able to ascertain the quality of the final Pareto optimal solution set in terms of its acceptability. Defining objective functions as fuzzy variables it is possible to input user
preference information. Fuzzification of objectives and notion of fuzzy dominant fronts is further explained in the subsequent Sections.

**Fuzzification of Objectives**

During fuzzification, the objective functions are converted into fuzzy variables using fuzzy sets. A Gaussian Activation Function (AF) has been chosen during the present work over triangular or trapezoidal activation functions to describe the fuzziness’s of objective functions due to its smooth shape transition between activation functions. Later, to initialize fuzzification of objective functions, a decision is made on the number of activation functions and their shape and position parameters. At this stage the information on user preferences can be included in the fuzzy variables.

![Activation functions and their arrangements for two objectives](image)

**Figure 7.3: Activation functions and their arrangements for two objectives wherein function (a) is preferred over function (b)**

To begin with, the number of activation functions and their parameters, which are minimum fuzziness points (i.e. the mean of Gaussian function) and the standard deviation are decided by the user. In an example shown in Figure 7.3, four Gaussian AFs are used to define two fuzzy objective functions. Values for both objectives are normalized and their limiting values are 2 and 8. Note that the AFs are overlapping and the transition between AFs is smooth as desired and not abrupt. Normally, the standard deviation is kept the same for all the AFs of an objective (Figure 7.3b). However to incorporate higher user-preference of an objective, different standard deviations for AFs can be used as shown in Figure 7.3(a). By constricting the standard deviation of ‘AF1’ and expanding that of ‘AF4’, solutions providing low values of the objective function are supported and those providing higher values are penalized. Clearly from the two objective functions shown in Figure 7.3, objective
7.7 Discussion on Results

represented by Figure 7.3(a) shall get more preference. In a similar fashion, user may decide the acceptable value for an objective and accordingly make a decision for the position and standard deviation of the AFs.

It will be later explained as how the position and standard deviation of these AFs affect the placement and width of fuzzy dominant fronts. It is recommended to dynamically update the universe of discourse of the objectives for fuzzification, based on their limiting values $(f_{\min}^k, f_{\max}^k)$ over successive iterations (k). Thus even if a particular solution does not change between iterations, it may not have same fuzzy front index in succeeding epochs. Number of AFs to represent an objective function is another important parameter to decide since this influences the number of fuzzy dominant fronts. Larger number of AFs used for an objective will result in the increased number of fuzzy dominant fronts and will provide more discrimination between solutions as explained in the following Section.

**Fuzzy Dominant Fronts**

The concept or the definition of non-dominance is more qualitative than quantitative and hence it is proposed to use fuzzy dominant fronts in place of conventional crisp non-dominated fronts. While comparing two solutions in NSGA II, a solution providing ‘better’ or ‘not worse’ objectives is selected. Since ‘better’ and ‘worse’ are qualitative variables it is not possible to decide upon the extent of these qualitative variables (how much better or worse) using crisp numbers. The irresolute state results in lower discrimination which is apparent from Table 7.3. Further the definition of non-dominated solutions in NSGA II is not straight since we are not looking for solutions which are dominating but we select solutions which are not dominated by other solutions in the population. In contrast to this, the proposed fuzzy dominance criterion is simple wherein dominating solutions in terms of the activation scores (explained later in the Section) of objective functions are selected and placed in the fuzzy dominant fronts. The definition of the fuzzy dominance which is similar to the crisp dominance criterion is given later in this Section.

A complete Pareto front should consist of solutions in two major regions of the solution space as shown in Figure 7.4. The first one is (A) the most preferred region where all the solutions are better in terms of all the objectives (this region is different from the weighted average approach accounting equal weights to all the objectives). The second region of interest is the extremities of Pareto front (B&C) wherein a bias among objectives exists. End user sometimes may be interested in finding trade-off solutions whereby preferring certain objectives at the expense of others.
NSGA II often is criticized for its inability of finding these extreme solutions on the Pareto front [169]. Two variants of fuzzy sorting genetic algorithms (FSGA) are proposed here namely Equitable fuzzy sorting GA (EFSGA) and Biased fuzzy sorting GA (BFSGA). The first approach aims to find a Pareto optimal front by selecting solutions which are dominating in terms of qualitative objective values. Thus the first approach mainly focuses on the most desirable solutions. On the other hand, the second approach tries to find solutions which excel in at least one objective, regardless of its fitness in the other performance criteria. Consequently the final Pareto front obtained using the Biased fuzzy sorting, solutions in the extreme regions of Pareto front can be obtained. The methodology and implementation of these approaches are further explained below.

7.5.1 Equitable Fuzzy Sorting

Following fuzzification of the objectives their numerical values can be represented by a collection of linguistic terms in some proportion. In another words, numerical value of an objective function shall have some degree of activation in all the linguistic variables [157]. Later in the process a unique score is assigned to each of the linguistic terms as shown in the Table 7.4. Here zero activation score \( AS \) is given to the lowest fuzzy dominant front and is augmented by unity for subsequent higher fronts. In general the activation score for \( m^{th} \) activation function can be given by following relation.

\[
AS(m) = m - 1
\]  

(7.4)

Shapes of activation functions and their placement in the solution space are explained with illustrations in Figure 7.5. Here two objectives having universe of discourse between 2
7.7 Discussion on Results

and 8 and represented by four AFs are displayed. This illustration helps in understanding the placement of oblique equitable fuzzy fronts wherein the degrees of fulfilments for combination of fuzzy functions are equal. Thus the first front is characterized by fuzzy function values given by AF1 i.e. where function values are less than 3.

<table>
<thead>
<tr>
<th>Table 7.4: Activation scores of linguistic variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linguistic Variables</td>
</tr>
<tr>
<td>Activation Score (AS)</td>
</tr>
</tbody>
</table>

Identification of total number of fuzzy dominant fronts is further explained with the help of Table 7.5 by taking two examples of a MOP with two objectives, fuzzified using two and three activation functions respectively. A general expression providing the number of fuzzy dominant fronts for a MOP with ‘N’ objectives and ‘m’ activation functions can be given by (7.5). Here \( M_j \) is the total number of membership functions for objective function \( J \).

\[
F_f = 1 + \sum_{j=1}^{N} (M_j - 1)
\]

Thus a MOP involving two objectives, using 2 and 3 AFs for each of the objectives shall have 3 and 5 fuzzy dominant fronts respectively. By increasing number of AFs, the total number of fronts also increases which enhances the discrimination capability of the algorithm for selection of better solutions amongst the good ones. Furthermore, since the number of fronts for a MOP is definite the end user can ascertain the quality of solutions obtained after each epoch. For instance, when solutions belonging to the first front are obtained the user can confidently make a decision on termination of the simulation. Since in the first front all the objectives shall have acceptable values defined by the user. The next step after input (objectives) fuzzification is to compute the fuzzy inference. Inference mechanism of Fuzzy systems is implemented through its rule-base which is a collection of if and then rules connecting the antecedent (input) and consequent variables (output). General structure of a rule-base is given by as shown below.

If \( f_1 \) is \( AF_{i1} \) and, \( \ldots \ldots \) and \( f_N \) is \( AF_{im} \) then \( AS_i \) is \( y_i \)

Here \( f_1 \ldots f_N \) are the objectives as inputs to the fuzzy system, \( m \) is the index for the activation function level, \( AF_{i1} \ldots AF_{im} \) are the activation functions corresponding to objectives, \( AS_i \) is the activation score for \( i^{th} \) rule and its numerical value is \( y_i \). Total number of
rules $N_R$ is derived from the number of AFs and the antecedent variables and is given by following relation [27].

$$N_R = \prod_{j=1}^{N} M_j \quad (7.6)$$

Once again here, $j$ is the index for the objective function; $N$ stands for the total number of objective functions, and $M_j$ is the total number of activation functions for objective function $j$. Thus when two objectives are represented using four activation functions (as shown in Figure 7.3), a total of $4^2$ i.e. 16 rules shall be formed. These rules are the combination of all possible arrangements of activation function of the two objectives.

As discussed before, numerical values of both objective functions shall have some degree of activation in almost all the arrangements of AFs. In other words, numerical values of objectives shall find varying degree of fulfillment in all the rules and each rule shall give an output correspondingly. The overall activation score for a pair of input objective values is the weighted average of all the rule outputs as explained below. Computation of output of the fuzzy system for a set of input objectives is a two step procedure based on a weighted average defuzzifier [154]. Degrees of fulfillment of all the rules for given objective values are computed and averaged to find the final activation score.

Upon establishment of the AS, the output of each rule i.e. $y_i$ can be computed using following equation.

$$y_i = 1 + \sum_{j=1}^{N} [\text{AS}(m_{ij})] \quad (7.7)$$

Here $i$ represent the rule index and $m_{ij}$ is the activation function which is active for objective function $j$ in the rule $i$. The weighting of objectives for each rule can then be computed by considering the product of all the applicable activation function’s degree of fulfillment. This degree of fulfillment can be computed using (7.8) and the weighting is given by (7.9). Here $f_j$ is the input objective value and $N$ stands for the number of objectives. Note that $\bar{f}_{ij}$ and $\sigma_{ij}$ are dependent on the active activation function $m_{ij}$ and are updated during successive iterations based on the limiting values $(f_{\text{min}}^{k}, f_{\text{max}}^{k})$ of objectives.
7.7 Discussion on Results

Crisp output of the fuzzy inference or the overall activation score of a solution is the weighted average of all the individual rule consequents for this given set of objective values. The overall activation score (OAS) can be computed using (7.10) as shown below.

\[
AF_{ij}(f_j, \bar{f}_ij, \sigma_{ij}) = ae ^{-(f_j-\bar{f}_ij)^2 / 2\sigma_{ij}}
\] (7.8)

\[
w_i = \prod_{j=1}^{N} AF_{ij}
\] (7.9)

Crisp output of the fuzzy inference or the overall activation score of a solution is the weighted average of all the individual rule consequents for this given set of objective values. The overall activation score (OAS) can be computed using (7.10) as shown below.

\[
Y = \frac{\sum_{i=1}^{N_f} (w_i y_i)}{\sum_{i=1}^{N_f} w_i}
\] (7.10)

The fuzzy dominant front number \(Y^*\) for a solution can then be obtained from this crisp output using (7.11), where floor(.) is used to represent the function which returns the integer which is less than or equal to the argument.

\[
Y^* = floor(Y)
\] (7.11)

In context to the fuzzy objective values and their overall activation score, the fuzzy dominance criterion can now be defined as below. This definition is similar to the definition of crisp dominance criterion [109]; however in the present definition the overall activation score is considered in place of individual crisp objective values.

**Definition 1 (fuzzy dominance criterion)** Solution \(x_1\) will dominate another solution \(x_2\) provided the overall activation score of solution \(x_1\) is not inferior to \(x_2\). In other words, the overall activation score of \(x_1\) must be superior or equal to the overall activation score of \(x_2\).

The overall activation score of a solution is termed as its fuzzy front index and the solution is accordingly placed in this front. To further explain the formation of fronts, illustrations showing placements of fuzzy dominant fronts for two objectives (represented by four AFs) are provided in Figure 7.5. The distribution of OAS has been shown in Figure 7.6 wherein it is apparent that depending on function values, the solutions shall be ranked between 1 and 7 where first front is most desirable for a minimization goal and the seventh front is the target for a maximization problem.

This is further illustrated in a planer fashion in Figure 7.7 where the solution space is divided in seven distinct fronts.
Figure 7.5: Activation functions and their placement in the solution space for fronts
(a) 1, 2 & 3, (b) 4 & 5, (c) 6 & 7 using EFSGA approach

Here the first front is the most desirable and subsequent fronts have decreasing grades of acceptability when the objectives are required to be minimized. It is apparent from the arrangement of these fronts that the solutions are gradually guided in the most desired region while exploring the entire solution space. The floor function is applied on the crisp outputs \( Y \), helps to maintain diversity in the solutions. If the crisp output for fuzzy front index is not
converted to an integer, all the solutions are likely to have a unique real number and as such there shall be rare instances of a tie between two solutions.

Table 7.5: Activation scores of linguistic variables

<table>
<thead>
<tr>
<th>Number of AFs</th>
<th>Activation Functions</th>
<th>Activation Scores for AFs</th>
<th>Possible outcomes for Fuzzy Objective Values for two objectives</th>
<th>Activation Score (AS)</th>
<th>Fuzzy dominant Front Index (AS+i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>AF1</td>
<td>0</td>
<td>AF1₁ &amp; AF1₂</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>AF2</td>
<td>1</td>
<td>AF1₁ &amp; AF₂₂</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AF₂₁ &amp; AF₂₂</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>AF1</td>
<td>0</td>
<td>AF1₁ &amp; AF₁₂</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>AF2</td>
<td>1</td>
<td>AF1₁ &amp; AF₂₂</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>AF3</td>
<td>2</td>
<td>AF₁₁ &amp; AF₃₂</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AF₂₁ &amp; AF₂₂</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AF₃₁ &amp; AF₃₂</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 7.6: Fuzzy fronts and degrees of fulfillments of the rules in the solution space for EFSGA

Consequently, the crowded distance operator shall not be invoked and the solutions which are distantly placed shall become extinct over successive iteration. It is important to note here that the preferences among objectives which were incorporated during fuzzification of objectives shall be apparent in the width of these fuzzy fronts. Preferred objectives are likely to have slender front width compared to the less preferred objectives.
Despite above merits there is a possible drawback in the *Equitable fuzzy sorting* approach that, though, it can effectively search the most desired region of the solution space, it may not be able to provide us the extreme points of the Pareto front. This limitation is due to the inherent characteristics of the selection scheme and to overcome this limitation an alternate sorting approach has been proposed as discussed below.

### 7.5.2 Biased fuzzy sorting

While the previous approach tries to encourage solutions that are more in line with the user’s preference, an alternative approach is proposed which stresses on the solutions which excel in at least one objective. The algorithm proceeds in a similar manner as the *Equitable fuzzy sorting* approach until the fuzzification of objectives and computation of activation score. Nevertheless, total number of fuzzy dominating fronts and their placement in the solution space are the distinguishing characteristics of *Biased fuzzy sorting* approach. Here the total number of fuzzy dominating fronts is same as the number of activation functions used to define objectives. Thus, *MOPs* consisting of two and three objectives defined by four activation functions shall each have four fuzzy fronts altogether. Further, while constructing the fuzzy rule base, the antecedent part is once again the combination of all possible arrangements of activation functions of the objectives. Following this the total number of rules shall be exactly same as the previous approach and can be given by (7.6). However the consequents of individual rules in this approach are different and can be found using (7.12). It
is important to note here that the rule output is only influenced by the objective function with the lowest activation score.

\[ y_i' = 1 + \min_j [AS(m_{ij})] \]  

(7.12)

The degree of fulfillment for solutions in the population can now be computed using (7.8) and (7.9) as before. Overall activation score of a solution is found using (7.10) and once again to preserve diversity in the solutions and prevent algorithm from converging to a local optima, a \( \text{floor}(.) \) operator is applied to the output from (7.11). This final output is termed as the Fuzzy dominant front number \( Y^* \) of a solution.

As a result of above sorting algorithm, solutions in the population are classified in fuzzy dominating fronts which are equal to the number of activation functions. The placement of activation functions in the solution space and distribution of degree of fulfillment for various combinations of activation functions (antecedents) for an example problem of two objectives using four \( AFS \) are displayed in Figures 7.8-Figure 7.10. The four fuzzy dominating fronts, which are quite apparent in the illustrations, represent the Pareto fronts.

The preferences incorporated by the end user at the fuzzification stage shall be more visible here in the form of widths of various fuzzy fronts.

![Figure 7.8: BFSGA Fuzzy fronts and their placement in the solution space](image-url)
7.7 Discussion on Results

Figure 7.9: BFSGA Fuzzy fronts and degrees of fulfillments of the rules in the solution space

Figure 7.10: BFSGA Fuzzy fronts and their placement in the solution space

Figure 7.11: BFSGA Fuzzy fronts incorporating preference, $f_1$ is preferred over $f_2$
For instance the fuzzy fronts formed by the two objectives shown in Figure 7.3 shall have non-uniform width of fronts and the first two fronts shall be slender for function $f_1$ as shown in Figure 7.11.

When this approach used in parallel with the *equitable fuzzy sorting* approach, it can help preserve diversity in the solution population and simultaneously highlight solutions which are highly competent in at least one performance criterion.

A flowchart showing various implementation steps of fuzzy sorting based evolutionary algorithm is shown in Figure 7.12. The steps involved are further explained below.

1. **[Start]** Specify a termination criterion, in the present case the algorithm terminates when solution in the first front are explored.
2. **[Initialize]** Initialize a random population of $Q$ chromosomes/binaries.
3. **[Fitness]** Evaluate the values of objectives for each chromosome in the population.
4. **[Rank]** Compute the OAS for these solutions following either *EFSGA* or *BFSGA* approach and classify population into fronts using fuzzy dominant rank obtained from (7.11).
5. **[Merge]** Make a copy of parent population in random order and merge with parent $Q$ solutions.
6. **[Selection]** Perform binary tournament selection by filling all the $Q$ places according to the fuzzy dominance ranks of the chromosomes while using crowding distance operator in cases when two solutions carry equal ranks, discard the left out solutions.
7. **[Crossover]** Cross over the binaries using a crossover probability to form new offspring.
8. **[Mutation]** Mutate new offspring at each bit with a mutation probability.
9. **[Merge population]** Merge the new offspring in the parent population and form a population of $2Q$ size.
10. **[Rank]** Compute fuzzy dominant rank of offspring and once again classify the combined population into fronts based on their ranks.
11. **[Check]** If the termination criterion achieved, go to step 13 otherwise go to step 12.
12. **[Iteration]** Go to step 6 with the offspring population of $N$ solutions.
13. **[End]** Stop and return the final population when solutions in the first fuzzy front are obtained.
14. Choose the final solution providing maximum OAS.
Selection of most desirable solution from Pareto front

As a result of the optimization we finally get a fuzzy Pareto front consisting of solutions which are all equally good as they all have same activation score. However the end user is always interested to find one particular solution which is to be finally used. Normally the end user makes a decision based on the different objective values and picks up a solution which is good for certain chosen objectives.

This approach is quite subjective and is purely intuitive. The overall activation score index has been proposed to be used which is obtained from (7.10). All the solutions in the fuzzy Pareto front have a unique real number for this activation score (before using floor(.) function) and the solution with minimum $AS$ has best compromised combination of all the objectives when the goal of optimization is minimization.

Figure 7.12: Flowchart representation of the FSGA process
Hand calculations

The example problem shown in Table 7.3 is used here to explain the enhanced discriminating capabilities of fuzzy sorting. The minimization MOP has six objectives and five solutions are picked up from certain population. Except for the first objective which is more important, other objectives are valued equally. The preference to the first objective is incorporated in the fuzzification process by altering the standard deviation values as shown in Figure 7.13. Next following Equitable fuzzy sorting approach and using (7.5), total number of fuzzy fronts can be calculated as 19 (1+3×6=19).

Number of fuzzy rules is found using (7.6) which in this example shall be 4⁶ or 4096. Thus 4096 combinations of activation functions are possible for six objectives with four AFs. The consequents (7.7) for these rules are simply the sum of the activation scores of their constituent AFs and can be easily found by implementing simple code. Finally using (7.8-7.10) the overall activation scores are calculated for above five solutions. These are further converted into integers which are less than or equal to their respective arguments by applying floor(.) operator (7.11).

![Figure 7.13: Preference incorporation during fuzzification of objectives. Here (a) is preferred over (b)](image)

The final fuzzy front indices obtained for the five solutions are displayed in Table 7.3. If it is desired to select one solution out of five alternatives, the user can check the AS of these solutions obtained from (7.10) before applying floor(.) function. The solution with minimum AS (which is the first solution in this case) shall be selected. The discriminating ability of
fuzzy based sorting to select better solutions over NSGA II is quite evident from results shown in Table 7.3.

Fuzzy Sorting Genetic Algorithm (FSGA) is further used to perform the design analysis of a parallel robot developed for ankle joint rehabilitation. The problem formulation, FSGA implementation and results obtained are discussed in the next Section.

7.6 Design Optimization of the Wearable Ankle Robot

7.6.1 Experimental results from simple FSGA

FSGA was used to carry out the design optimization problem of the wearable ankle robot. The algorithm was initialized with 1000 design solutions each represented by 96 bit binary
strings. All the six objectives were defined as fuzzy objective functions as shown in Figure 7.14. Note that the preferences among objectives were incorporated at this stage. Two objectives namely, condition number and moment error were important and it was desired that the condition number should be well below 5 and the solutions providing almost no moment error be encouraged. Spreads of the first activation functions for all the objectives were chosen to extend up to the accepted values of respective objectives. Thus the first fuzzy dominant front which had all the objective values corresponding to the first $AF$s was the desired solution space. It is also important to note here that the universe of discourse for all the fuzzy objectives was not kept same for the successive iterations but was decided based on the limiting values of objectives found during the iteration. The standard deviation and the mean for the $AF$s were thus proportionally changed throughout the optimization. Thus a solution may have unlike $AS$s in different iterations. However, the preference assigned to the objectives was maintained throughout the optimization. Number of fuzzy rules once again was found using (7.6) to be $4^6$ or 4096. Consequents (7.7) for these rules which are sum of the activation scores of their constituent $AF$s were also computed.

The overall activation scores for prospective design solutions were calculated using (7.8-7.10). While selecting solutions from the combined population of parents and offspring, binary tournament selection operator [141] was used. Selection was made on the basis of fuzzy dominant ranks of the solutions.

![Figure 7.15: Simulation results following simple FSGA approach shows that the final solution obtained lies in the third front for two different experiments (a) quick conversion, (b) slow conversion](image-url)
Varieties of crossover techniques exist [141] but in the present case, a six point crossover operator was used with crossover probability as 0.95 as a usual practice. Mutation was performed on solution binaries after crossover operation with 0.02 probabilities. No significant change in the final solutions could be observed by changing crossover and mutation parameters. The algorithm terminated when a solution in the first fuzzy dominant front was obtained.

Initially the optimization was conducted by selecting solutions solely based on their fuzzy inference (7.10) without applying the scheme of fuzzy dominant fronts. A similar approach was used in [108, 172, 178, 179] wherein, overall activation score of the solutions was computed without applying the floor(.) function. This further means that all the solutions were carrying a unique real number for their fuzzy dominance ranks. Consequently during binary tournament selection, the crowding distance operator was not used as there was never a tie between solutions. Results from two experiments, initialized with 100 and 1000 solutions respectively are displayed (Figure 7.15) and analyzed.

Unique solutions from the entire population of all the iterations are plotted after sorting them in the order of their fuzzy dominance ranks. The algorithm terminated after 25 and 10 epochs for the two experiments giving two different final design solutions sited in the third fuzzy dominant front defined by (7.11). By convergence it was meant that all the solutions in the final iteration, carried approximately same objective values i.e. the solutions present approximately same final design solution. It was thus concluded that the final solutions obtained are not globally optimal.

7.6.2 Experimental results from EFSGA

The design optimization was performed using EFSGA where selection, crossover and mutation operators are applied as discussed in the previous Section.

The universe of discourse for the fuzzy objectives was computed from the limiting values of objectives during iterations. Algorithm terminated after 25 iterations when some of the solutions converged to the first fuzzy front. Due to difficulty in plotting all the six objectives together, three important criteria (condition number, stiffness and norm of actuator forces) are plotted in Figure 7.16. Solutions are plotted in different colours representing their respective fronts.
7.7 Discussion on Results

Figure 7.16: Results from EFSGA approach for three vital criteria with enlarged final front solutions shown in the inset. Color sequence represents fronts in increasing order starting from blue, red, green, magenta, cyan, black and grey.
In the final set of design solution, 522 solutions out of 1000 are placed in the first front whereas the rest are on the second front. To select the best design solution among these 522
solutions was a big cognitive load on the designer and certainly he cannot analyze all of them to select one.

Interestingly if we look into the overall activation scores of these final front solutions which have been obtained after using (7.10) we will find that these scores are unique real numbers which basically are the fuzzy inference for objectives. Thus it is possible to select a solution which has the minimum score and shall have the best values of all the objectives, we intend to minimize.

In the present case a solution having overall activation score for objectives as 1.94 can be selected (Table 7.6) among the first front solutions since it has a better combination of all six objective values compared to other solutions on the same front.

### 7.6.3 Experimental results from BFSGA

Following above approach (EFSGA), a best combination of objectives can be obtained however in some instances the end user is interested in finding solutions which are highly competent in a particular performance criterion.

The end user has to later make a trade-off between competing solutions and select the one more suiting to a particular application. In such circumstances it is proposed to use a *biased fuzzy sorting* approach instead. As explained earlier, this algorithm encourages solutions which excel in at least one objective. It differs in the formation of fuzzy fronts from the previous approach and is able to explore the extremities of the Pareto front. The final front solutions obtained using this approach, are not the best combination of all the objectives but individually they excel in at least one criterion.

This fact is further evident from Table 7.6, wherein solutions containing best of the individual objectives are displayed and their fuzzy fronts are computed using EFSGA approach. Here $F_1$ to $F_6$ are the six objectives which are being optimized and have been defined earlier. Here, final design solutions, obtained using EFSGA approach and NSGA II are compared with the best solutions acquired from BFSGA approach.

Best individual objective values are shown with bold face and it is clear that most of the best values come from BFSGA approach. However, following BFSGA approach, the best objective values come from different design solutions. Therefore the end user has to select a solution pertaining to the most preferred objective. Design solutions for three important criteria obtained after 25 iterations, following BFSGA approach are displayed in Figure 7.17. Apparently the best design solutions obtained from this approach are found to be placed in the second front and the first front solutions could not be explored during the stipulated 25
iterations. These results are quite similar to the results received from \textit{NSGA II} approach; however \textit{BFSGA} approach excels \textit{NSGA II} in finding extreme points on the Pareto set (Table 7.6). The enhance performance of BFSGA is on part of its selection mechanism, wherein the extreme objective values are more emphasized.

\textbf{7.7 Discussion on Results}

Initially the design optimization was carried out following NSGA II approach and the results were analyzed. It was revealed that due to the continuous nature of the solution space and large number of objectives, the non-dominant criterion lost its significance since all the initial design solutions became non-dominant after the third iteration. Clearly the use of non-dominance criterion was not the best approach in the present case. Furthermore, the lack of discrimination among solutions resulted in calling the crowding distance operator frequently, which made NSGA II a computationally expensive algorithm.

As an alternate to non-dominance criterion a fuzzy based approach has been advocated in the literature [108, 172] and a similar approach was used as a next step and the results obtained were examined (Figure 7.15). It was observed that after certain iterations all the design solutions converge at a single point. It was further noticed that this optimal point was not the global optimal point since different experiments yield different optimal solutions. One of the probable reasons for this local optimization was that the algorithm was guided in a direction which minimizes the fuzzy inference of solutions (7.10) thus making it a single objective optimization. Secondly, since all the solutions carry a unique fuzzy dominance rank, the crowding distance operator which maintains diversity among solutions was not being called and thus distant solutions gradually faded away, leading the algorithm to converge to some local optimal solution.

Subsequently, EFSGA was used and experimental results were investigated. Three important criteria were plotted in Figure 7.16 where the fronts were represented using different colours as explained in the previous Section. These fronts cannot be visualized distinctly due the difficulty in comprehending six objective values together. However, the final front shown in blue colour is apparent in the inset windows (Figure 7.16) wherein the enlarged view of a selected region is shown. The results obtained exhibited an expected relation among various objectives. It has been shown [27] that the value of maximum actuator forces can be minimized by maximizing the minimum singular value thus when the condition number is minimized, the minimum singular value also gets maximized which in
turn reduces actuator forces to some extent. This statement is evident from the results displayed in Figure 7.16. Further, stiffness in the task space was computed from actuator stiffness’s using Jacobian matrix of the robot. Thus stiffness improves when the Jacobian matrix is well conditioned and vice versa (Figure 7.16). Similarly, Actuator forces show an expected positive correlation with stiffness.

It is evident from these illustrations that the solutions are converging towards improved values of the objectives which certainly is due to the mechanism of proposed selection scheme. Referring back to Figure 7.7, the pre-placement of fuzzy fronts guides solutions to the first front which accepts solutions with most preferred objective values. The improved discriminating power of the selection scheme also contributes positively by selecting vital few from useful many.

Analyzing results acquired from BFSGA, it was observed that though the results were similar to those obtained using NSGA II, this approach was able to effectively explore the extremities of the Pareto front. These solutions cannot be found following EFSGA approach. Though a complete Pareto front cannot be visualized due to the nature of optimization problem and its constrained solution space, this approach was able to provide the best individual values of objectives (Table 7.6). This approach is useful when the end user is interested to find solutions which excel in one of the objectives regardless of other objective values.

From above results this can be deduced that EFSGA approach should be used when design solution providing best of all the objectives is desired as in the present case, whereas BFSGA approach is recommended when certain criterion is more valued at the expense of others. Final design solutions obtained from all the above mentioned approaches are found to abide by all the constraints (7.1, 7.2) and thus are feasible.
Table 7.6: Final robot design solutions from EFSGA and BFSGA compared with NSGA II final result (F₁: Condition number, F₂: Percentage workspace utilization, F₃: Stiffness (N/m), F₄: Norm of actuator forces (N), F₅: Normalized error in moment (Nm), F₆: Maximum actuator force (N)). Robot design parameters q₁, q₂...q₈ have been explained in (Figure 6.1)

<table>
<thead>
<tr>
<th></th>
<th>q₁</th>
<th>q₂</th>
<th>q₃</th>
<th>q₄</th>
<th>q₅</th>
<th>q₆</th>
<th>q₇</th>
<th>q₈</th>
<th>F₁</th>
<th>F₂</th>
<th>F₃</th>
<th>F₄</th>
<th>F₅</th>
<th>F₆</th>
<th>Fuzzy front index</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFSGA</td>
<td>37.50</td>
<td>0.12</td>
<td>37.67</td>
<td>0.16</td>
<td>58.04</td>
<td>0.12</td>
<td>62.41</td>
<td>0.12</td>
<td>1.99</td>
<td>100</td>
<td>385</td>
<td>340</td>
<td>0.37</td>
<td>261</td>
<td>1</td>
</tr>
<tr>
<td>BFSGA</td>
<td>44.18</td>
<td>0.10</td>
<td>44.89</td>
<td>0.14</td>
<td>63.79</td>
<td>0.10</td>
<td>79.15</td>
<td>0.11</td>
<td>1.63</td>
<td>100</td>
<td>270</td>
<td>407</td>
<td>0.37</td>
<td>423</td>
<td>4</td>
</tr>
<tr>
<td>BFSGA</td>
<td>52.62</td>
<td>0.11</td>
<td>46.26</td>
<td>0.16</td>
<td>77.18</td>
<td>0.10</td>
<td>73.34</td>
<td>0.11</td>
<td>2.63</td>
<td>100</td>
<td>435</td>
<td>345</td>
<td>0.48</td>
<td>437</td>
<td>4</td>
</tr>
<tr>
<td>BFSGA</td>
<td>56.69</td>
<td>0.12</td>
<td>43.90</td>
<td>0.15</td>
<td>58.08</td>
<td>0.10</td>
<td>69.65</td>
<td>0.10</td>
<td>2.42</td>
<td>100</td>
<td>238</td>
<td>356</td>
<td>0.31</td>
<td>496</td>
<td>5</td>
</tr>
<tr>
<td>BFSGA</td>
<td>37.91</td>
<td>0.12</td>
<td>43.40</td>
<td>0.15</td>
<td>73.88</td>
<td>0.11</td>
<td>66.66</td>
<td>0.10</td>
<td>2.30</td>
<td>100</td>
<td>185</td>
<td>384</td>
<td>1</td>
<td>234</td>
<td>2</td>
</tr>
<tr>
<td>NSGA II</td>
<td>46.43</td>
<td>0.10</td>
<td>48.53</td>
<td>0.15</td>
<td>60.94</td>
<td>0.10</td>
<td>67.43</td>
<td>0.11</td>
<td>2.59</td>
<td>100</td>
<td>230</td>
<td>377</td>
<td>0.53</td>
<td>299</td>
<td>2</td>
</tr>
</tbody>
</table>
7.8 Chapter Summary

The wearable robot design optimization was treated as a multi-objective optimization problem in this Chapter. To begin with a brief description of the available multi-objective optimization approaches was provided. The design optimization was carried out using two most popular approaches, namely, preference based optimization method and the evolutionary algorithm based method called NSGA II. Preference based approach was discarded due to the subjectivity involved in deciding the weights or preferences. Furthermore, in the present design optimization problem where the objectives are of conflicting nature, a decision on the preferences of objective was difficult to make. The NSGA II was implemented later while investigating the use of evolutionary approach in the present problem. It was found that due to increased number of objectives and the continuous solution space, the NSGA II could not discriminate between better and worse solutions and after three iterations all the prospective solutions were declared as non-dominating. In view of this, the algorithm later became computationally expensive and it was observed that despite the large variation in the values of objectives all the competing solutions shared the first non-dominant front, making it difficult to select one or more potential design solutions.

Thus to address the shortcomings observed in the existing EA approaches, a new fuzzy dominance based EA was proposed in this Chapter. Two variants of the fuzzy sorting based genetic algorithms (FSGA) namely; EFSGA and BFSGA were developed to explore the two important regions of the Pareto front. The design optimization was performed using both these variants of the fuzzy sorting genetic algorithms. It was found that the EFSGA exhibited enhanced discrimination power in selecting good solutions from the population and was able to converge solutions to the desired optimum levels set by the end user. Conversely, the BFSGA performed well in searching the extreme zones of the Pareto front. Though the pattern of results obtained from BFSGA was similar to that obtained from NSGA II, BFSGA was able to provide better results. Finally, the optimal design obtained using EFSGA approach was selected for the prototype construction.
Chapter 8 Pneumatic Muscle Actuator Modelling and Fuzzy Logic Control

Wearable ankle rehabilitation robot has been constructed using PMA, replacing conventional electromagnetic actuators. Owing to their benefits over conventional actuators, PMA have been found ideal actuators for a wearable rehabilitation robot; however their non-linear and time dependent behaviour is a matter of concern. Reading through the previous research, it was found that in earlier works, static and dynamic modelling of PMA had been performed either for a constant external force or without any external loading. Moreover, the accuracy from earlier models was also required to be improved in order to achieve better kinematic compliance. Therefore, in the present research, dynamic models of PMA have been developed using three approaches, namely, ANN, Takagi-sugeno fuzzy model and Mamdani fuzzy model. Performance of these models in terms of accuracy and interpretability was evaluated. Mamdani based fuzzy model, which showed better interpretability, was finally used to model PMA characteristics. The proposed model has been found to be robust against external force acting on PMA. To facilitate motion therapy of ankle joint with the aid of wearable robot, an iterative fuzzy controller was developed which makes use of the fuzzy dynamic model of PMA. The fuzzy controller was implemented on the actual robot prototype, to perform experiments, with the aim of evaluating robot’s performance in ankle rehabilitation treatments.

8.1 Actuator Modelling

PMA are new types of actuators which have been recently used in the fields of rehabilitation and industrial robotics and gradually gaining popularity over conventional electromagnetic actuators and pneumatic cylinders. Pneumatic muscle actuators have a thin walled rubber (preferably latex or non-vulcanized rubber) bladder confined into a braided sheath. One of the ends of the rubber tube is closed whereas an air hose is inserted into the other end to inflate or deflate. The braided sheath is made of flexible but inextensible threads arranged in a criss-cross fashion. When the inside latex tube is inflated, the outer sheath is stretched outward like a scissor and as a result its length is shortened. The PMA has high axial stiffness but they are compliant along radial direction.
8.1 Actuator Modelling

When the inner rubber tube is inflated, it expands and the diameter of the rubber-sheath combine, called ‘muscle’ commonly, easily increases. Since the external sheath is inextensible, the muscle length decreases axially exerting large axial force at the actuating end of the muscle. If a mechanical load is attached to this end of the muscle and an external work is done at a higher rate on the load, high power can be produced. When compared with conventional actuators such as DC motor driven linear actuators and pneumatic cylinders, PMA exhibit several advantages [118]. Primarily, PMA are very economic and easy to construct; they are light in weight and are inherently flexible. Therefore they can provide safe and soft interaction due to high compliance and are more suitable for robot-human interactions applications [123].

A detailed discussion on the previous work relating to the static and dynamic modelling of PMA has been provided in Chapter 2. It was revealed from the collective previous work that there are at least two vital issues that need to be properly addressed. Firstly, majority of the research on PMA modelling has been done for constant loading or in other words, PMA has not been subjected to varying loads. It is shown later in the following Section that the pressure-length characteristics of PMA greatly depend on the applied loads which cannot be taken as constant. Secondly, the prediction accuracy from the previous models also needs to be improved in view of more precise and sophisticated applications of PMA such as the rehabilitation robotics.

To exploit these opportunities of improvement, in the proposed work, two possible AI approaches namely, ANN and FIS are used to perform dynamic modelling of PMA. A multilayer feed forward neural network with a single hidden layer is developed and used due to its suitability in the present application. Similarly fuzzy logic based approaches namely, the TS fuzzy model and the Mamdani fuzzy model have been developed to construct accurate PMA models. The two Fuzzy models have been optimized using the MGA discussed in Section 4.4.3.

Results from the ANN and fuzzy models were analyzed and compared for a final selection of a suitable approach to model the complex behaviour of PMA. The parameters of ANN and both the fuzzy models were optimized with the help of a training database. To obtain a real time training database consisting instantaneous sensor readings for pressure, length and actuator force of the PMA, experiments were performed on a test rig shown in Figure 8.1. Pressure values from the experiments were recorded and mean squared values of error in pressure values were obtained separately by comparing experimental values of pressure with pressure predictions from individual models.
8.2 Experimental Setup

The individual model parameters were altered in order to achieve minimum error in pressure prediction. Details of the experimental setup and deployment of various sensors are provided in the following Sections along with the major experimental findings.

8.2 Experimental Setup

Experiments were conducted on a single PMA to acquire sensor data for its instantaneous length, pressure and force. The test rig as shown in Figure 8.1 was developed using a robust structure to position and support the muscle assembly and sensors. The setup (Figure 8.1) includes a PMA assembly and three sensors namely, Linear potentiometer (for displacement measurements of PMA), load cells for force measurement and pressure sensors. The PMA assembly consisted of a hanger attached with a pan for placing loads on one side of the PMA. The other end of PMA was suspended from top of the frame by means of a load cell (shown in Figure 8.1). The linear potentiometer was positioned parallel to the PMA to read instantaneous muscle length. Pressure sensor, load cell and the potentiometer are shown in Figure 8.1 and Figure 8.2.

To ensure proper alignment of the linear potentiometer with the muscle, a guide rail was provided as shown in Figure 8.1. The PMA in the experimental set-up was placed with similar fittings as it was designed to be placed in the final robot design. Connections of the
two ends of the PMA and three types of sensors were arranged to emulate their intended placement in the wearable ankle robot.

The pressure sensor was positioned close to the air input port attached on the PMA to avoid error due to air losses in the pipe. The load cell, used to measure the actuator pull force dynamically, was connected to the PMA at its point of suspension. Loads were applied using the hanger attached to the muscle on its non-actuated end. The muscle was loaded with an increment of 25 Newton starting from 25N to 600N. All the three sensors (position, force and pressure) were duly calibrated by fitting suitable curves to their recorded values.

Pneumatic muscle was inflated by connecting it to the pressure supply from a compressor. The supply pressure was fixed at 4 bar and two pressure regulating valves each for inflating and deflating were used to control pressure inside the muscle. These valves were capable of providing a switching frequency with a period of 10 ms and were used to fill, leave inflated and empty the muscle actuator. Pulse width modulation was used to control switching of pneumatic valves and thereby achieve pressure control by controlling the air flow.

For fast communication with sensors and valves, a DSPACE processing system DS1103 was used [181]. All the codes to read and write sensors and valves were written in Matlab and realized in its Real Time Workshop interface. Matlab codes were converted into executable files that were compatible to work with DSPACE control desk interface. It is important to note here that the output signal from the pressure, force and position sensors was analogue and thus these were connected to ADC ports of DS1103 I/O panel. Pressure regulating valves, on the other hand worked with digital I/O’s, hence were connected to the digital I/O
panel of DSAPCE processing system. The overall system configuration including the PMA, various sensors and the DSPACE processing system is shown in Figure 8.3.

![System Configuration Diagram](image)

**Figure 8.3:** The overall system configuration used for the PMA testing

![Characteristics Graph](image)

**Figure 8.4:** PMA characteristics during inflating and deflating.

Initial experimental findings, while employing a single PMA (as shown in Figure 8.4) show that the characteristic curve between length and pressure of the PMA is non-linear. Different characteristics are obtained for inflating and deflating PMA. As a result two different lengths are possible for a given pressure in the PMA. Hence information on whether the actuator is inflated or deflated is essential while controlling its length. Further analysing Figure 8.5, it is found that the pressure-length characteristics is also dependent on the variable
external load on the actuator. A clear shift in the actuator behaviour can be observed in Figure 8.5 for higher loads on PMA. The characteristic curves tend to become straight when PMA is subjected to higher loads. These findings further support the observations made in the literature by previous researchers [114-128].

By virtue of its construction and actuation, PMA exhibit highly non-linear and time dependent behaviour. Primarily, the muscle deformation is non-linear because it is made of a viscoelastic material. Secondly, during operation, the muscle temperature increases with time due to inherent friction which severely affects its dynamic characteristics [123]. The working medium for PMA in the present work was air which is a compliant agent and further augments PMA’s uncertain behaviour.

Literature review concerning PMA modelling suggests that ANN and Fuzzy logic are the two AI approaches which are generally used to establish a mapping between inputs and their associated outputs for PMA [25, 126, 127]. Taking into account ANN first; there are fundamentally two architectures of ANN modelling namely, *feed forward networks* and *recurrent networks*. Out of the two networks the *recurrent network* has been found to exhibit limited performance compared to the feed forward architect of ANN [182]. The feedback loop of a recurrent network passes the data back and forth thus sometimes results in instability. Therefore in the present research a feed forward network of ANN has been developed to model PMA. The two variants of fuzzy models namely, Mamdani fuzzy model and TSK fuzzy model have been discussed earlier [154]. In the present work ANN and the
two variants of fuzzy model were developed and their performance was evaluated in terms of accuracy in predicting pressure inside PMA.

### 8.2.1 Tuning of AI models

To tune the three AI models (ANN and two Fuzzy models), normally a training database is required from the real system. Therefore, in the present case, the sensor data obtained from the above mentioned test rig was used as the training data. Instantaneous actuation of PMA, gradient of this actuation over time, and instantaneous force on PMA were considered as the input variables. Similarly, instantaneous pressure inside the PMA was taken as the output variable.

![Diagram](image_url)  
**Figure 8.6:** Overview of AI models used to predict pressure inside PMA.

![Diagram](image_url)  
**Figure 8.7:** Optimization of Fuzzy models which are used to predict pressure inside PMA.
AI systems were developed initially by intuitively initializing their model parameters. Input variables from the first set of training data was given to AI systems and output produced from AI systems was compared with the corresponding output variable of the same set of training data. Further iterations were performed by changing AI system parameters in order to reduce the error of their output prediction. The set of system parameters, providing minimum prediction error was accepted as the final AI system parameters. Subsequently, another set of data which is also obtained from the real system (test data) is given to this tuned AI system and the prediction error produced by this system is analyzed. An AI system is said to be interpretable, if it is able to provide output data with similar accuracy for all the test datasets.

To begin with a feed forward ANN model was tuned using training dataset from the real system and its prediction performance was analysed. The ANN system description is given in the following Section.

8.3 Artificial Neural Networks

Artificial Neural Networks are massively parallel distributed processing systems which are made up of intensely interconnected artificial neurons mimicking biological neuron system. In the literature they are commonly referred to as the parallel distributed processors, connectionist network and neuro-computers[183]. They show strong mapping capabilities and learn by examples or in other words they can be trained using known examples and later can be used to infer unknown instances of a problem.

Multilayered feed forward networks, as the name suggests are made up of three or more layers consisting of input, output and one or more hidden layers. Mathematical relationship between the layers is expressed in terms of several weights which are called as input-hidden layer weights and hidden-output layer weights. Algebraic computations are performed at the hidden layer before passing the inputs to the output layer. Identification of correct weight matrices is the key to the accuracy of an ANN. Various learning methods such as Reinforced learning, Hebbian learning, Stochastic learning and GD learning etc are used by researchers to identify weight matrices and other key parameters of ANN [154]. Back-propagation learning, a kind of GD learning is implemented in the present research using Matlab toolbox to identify ANN parameters [183].
The ANN network developed has three layers such as input, hidden and the output layer. A single set of training data for the model consisted of three inputs and an output as discussed in the previous Section. In the network developed in the present work the hidden layer had eight neurons each consisting of three weights (for individual inputs), which are processed along with the input variables to provide the requisite output variable. The number of hidden layer neurons was selected after a manual tuning. When more than eight neurons were used the model was found to be over fitting data values thereby loosing on interpretability. An additive bias term was also used to enhance the interpretability or the ‘universal approximation’ of the model. To evaluate the hidden layer neurons a hyperbolic tangent sigmoid transfer function was selected to closely emulate the non-linear behaviour of conduction current mechanism of a biological neuron [183]. Linear transfer function was used for the output layer as a usual practice. The database, obtained from the experiments on above mentioned test rig, is divided in two parts and these data segments are used to train and validate the ANN model. The network was trained using back propagation algorithm wherein weights and bias were updated using Levenberg-Marquardt optimization routine [104]. The ANN network is shown in Figure 8.8.

Weight matrix of the ANN system, which is basically set of ANN system parameters, obtained after the model was trained using training database, is provided in the Table 8.1.

**Figure 8.8: Artificial neural network showing three layers (input, hidden and the output layers) and eight hidden layer neurons.**
below for a quick reference. This is important to note here that the weight matrix does not provide any information and act as a mapping matrix only; it has been provided to facilitate reproduction of the results. Bias was used in both, the hidden layer (as well as for the output node). Numerical values of the bias used have also been provided in the Table 8.1 along with hidden layer neuron weights. Negative bias was obtained in the output layer and its value was -0.1859.

**Table 8.1: The complete weight matrix and the bias used in the hidden layer of ANN.**

<table>
<thead>
<tr>
<th></th>
<th>w1</th>
<th>w2</th>
<th>w3</th>
<th>w4</th>
<th>w5</th>
<th>w6</th>
<th>w7</th>
<th>w8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4199</td>
<td>2.3282</td>
<td>-0.3282</td>
<td>0.9763</td>
<td>0.5042</td>
<td>0.0854</td>
<td>0.6995</td>
<td>-2.5815</td>
</tr>
<tr>
<td></td>
<td>-5.4189</td>
<td>-1.9371</td>
<td>-0.9830</td>
<td>-0.5026</td>
<td>-2.8826</td>
<td>-0.2784</td>
<td>2.9146</td>
<td>0.7307</td>
</tr>
<tr>
<td></td>
<td>-0.3373</td>
<td>2.4508</td>
<td>-4.1580</td>
<td>-2.1361</td>
<td>0.1735</td>
<td>0.0743</td>
<td>3.9288</td>
<td>-4.8146</td>
</tr>
<tr>
<td>Hidden layer bias</td>
<td>2.5430</td>
<td>-0.1619</td>
<td>1.0581</td>
<td>0.2971</td>
<td>-2.6718</td>
<td>0.3029</td>
<td>2.8609</td>
<td>-4.3043</td>
</tr>
</tbody>
</table>

**Figure 8.9: Prediction errors from ANN model using testing and training data.**

Results from the network on the train and test data after being trained for 50 epochs are displayed in Figure 8.9. Mean Squared Error (MSE) for pressure prediction is found to be 0.0038 bar while using training data which is approximately 0.2% of maximum pressure value. This error increased to 0.0042 bar when the model is tested with the testing database.
The maximum deviation in pressure values from the model is obtained as 0.2743 bar and 0.3436 bar respectively for the training and testing data amounting to 13.7% and 17.18% of the maximum pressure value respectively.

Despite the fact that the variation in MSE between the training data and the testing data is small, the difference in the maximum deviation from the two databases is significant. The training errors also show an increasing trend which is also a matter of concern. Vacillation in the testing error shows that the model cannot be taken as interpretable. In another words to achieve greater accuracy from this model larger training data with finer resolution and wider spectrum is essential. The intrinsic fluctuations in the prediction errors and the increasing trend in the training error suggest that the ANN model cannot be used in the present work and alternative approaches should be explored.

Fuzzy logic is an alternate AI approach which can be used in the present problem of modelling PMA behaviour. By virtue of the continuous activation functions which are used to defining variables, fuzzy logic shows better interpretability [154], however its accuracy is not guaranteed unless its parameters are optimized. The accuracy from a FIS mainly depends on the proper selection of its parameters such as the rule base selection, minimum fuzziness points and spread of activation functions. In the present research, a scheme based on MGA was proposed in Chapter 4, to optimize parameters of fuzzy models for enhancing its prediction accuracy. Details of MGA implementation has been discussed in Chapter 4 while performing FK optimization.

In the present case, both variants of FIS namely, Takagi-Sugeno (TS) and Mamdani FIS were developed, optimized and implemented in order to select the better of the two approaches. It has been mentioned in the literature that fuzzy systems based on Mamdani inference, exhibit better abilities to interpret system behaviour between discrete data. On the other hand, TS fuzzy systems, though are accurate, lack interpretability [156]. Another fundamental distinction between the two kinds of fuzzy approaches is that, in Mamdani approach both input and output are fuzzy variables and a defuzzification method is required to apply fuzzy outputs to a real system. Suge no approach, in contrast, uses fuzzy input variables and crisp output variables and therefore defuzzification of outputs is not required [156]. Results obtained from both these models were compared with the outcomes of ANN approach discussed in the previous Section. Due to significant difference in the inference mechanism of these approaches, their optimization using MGA was implemented in different fashions. Development methodology and optimization of both kinds of fuzzy models are discussed in the following Section in the context of the dynamic modelling of PMA.
8.4 Takagi-Sugeno Fuzzy System

The main building blocks and inference mechanism of the TS fuzzy system were discussed in Chapter 4. This Chapter focuses on how the system is developed for the modelling of PMA.

8.4.1 Fuzzification

To begin with, the input variables of the system were converted into fuzzy variables and represented by their respective fuzzy sets. For the dynamic modelling of PMA, input variables were, load on the muscle, its instantaneous length and change in length whereas the output or the control variable was the instantaneous pressure inside the PMA. During fuzzification three Gaussian activation function (AF), linguistically named as Low, Medium and High were used to represent the fuzzy input variables as shown in Figure 8.10. Thus a total of nine AFs were constructed for the three input variables. The initial positioning ($q_1$) and spread ($q_2$) of the AFs were decided simply by dividing the universe of discourse or the range equally to accommodate all the AFs (Figure 8.11). Spread of these AFs, which is six...
times their standard deviation, was decided to be equal to the universe of discourse of the subject variable. Consequent or the output variable was not fuzzified but is obtained during the process of inference as explained later in this Section.

8.4.2 Rule-base

A typical TS fuzzy rule-base has following structure:

\[
\text{if } F \text{ is } A_{i1} \text{ and } L \text{ is } A_{i2} \text{ and } \Delta L \text{ is } A_{i3} \text{ then } P \text{ is } r_i
\]

Here \(i, (i = 1, \ldots, n)\) denotes the number of rules, \(A_{i1} \ldots A_{i3}\) are the activation functions in the antecedent part whereas \(r_i\) are the consequent parts (real numbers) of TS fuzzy models. Further, \(F, L\) and \(\Delta L\) are respectively the input load, instantaneous length and change in length for the PMA. Total number of rules for both types of fuzzy models (i.e. Takagi-sugeno and Mamdani fuzzy models) is same and is given as below by reproducing (4.10).

\[
N_r = \prod_{k=1}^{N} m_k
\]  

(8.1)

Where \(N_r\) represents the total number of rules, \(N\) is number of input variables and \(m_k\) is the number of linguistic terms of \(k^{th}\) input variable. Since there are three input variables represented by three AFs each in the present case, the total number of rules is \(3^3\) i.e. 27.

8.4.3 Inference Engine

Inference mechanism of the TS fuzzy model has been explained in Section 4.3.3 of Chapter 4 using relevant equations. These equations are once again provided here in the context of the present problem which has a single output. The two step inference procedure
(8.2&8.3) computes weighted average of the outputs from all the rules for given input values. First of all, the degrees of fulfilment of input values in all the rules are calculated using a product operation using (8.2). Later, the final output (8.3) is computed by taking the weighted average of all the rule fulfils.

\[ w_i = \prod_{k=1}^{m} Q_{ik} A_k(I_k) \]  \hspace{1cm} (8.2)

Where \( A_k(I_k) = [A_{k1} \ldots A_{KM}]^T \) and \( A_{km}(I_k) = ae^{\frac{(I_k - I_{km})^2}{2\sigma_{km}}} \) for \( m = 1, \ldots, M \)

Here \( Q_{ik} \) is a selection vector which chooses the activation function to be used in input \( k \) and rule \( i \). The final output in terms of pressure values \( P \) is obtained as shown below.

\[ P = \frac{\sum_{i=1}^{N_r} (w_i, r_i)}{\sum_{i=1}^{N_r} w_i} \]  \hspace{1cm} (8.3)

In above equations, \( w_i \) is the Gaussian activation value of \( i^{th} \) inference rule where \( I_k \) stands for the \( k^{th} \) input variable. Number of MF’s to represent a variable or the number of linguistic terms is defined by ‘\( m \)’. As is evident, all the activation functions \( A_{km} \) can be defined by two variables such as mean \( I_{km} \) and standard deviation \( \sigma_{km} \). Vector of consequent variables for TS fuzzy model is denoted by \( r_i \). It is useful to note that (8.3) can be further rewritten as (8.4), with \( x \) being a regressor vector and \( \Gamma \) is a vector formed by augmenting all the parameter vectors \( r_i \) from \( i = 1 \) to \( i = N_r \).

\[ P = x^T \Gamma \]  \hspace{1cm} (8.4)

When multiple observations (\( N \)) are available from the experiments, these observations can be combined together to give a system of equations of the form shown in (8.5), where \( \Theta = [P_1 \ldots P_p \ldots P_N]^T \) is the vector of desired output and \( X = [x_1 \ldots x_p \ldots x_N]^T \) is the regressor matrix. Further, \( p \) denotes the pattern/observation number and can range from 1 to \( N \).

\[ \Theta = X \Gamma \]  \hspace{1cm} (8.5)

As long as the number of observations is greater than the total number of consequent parameters to be identified, equation (8.6) can be used to find the parameters for the
consequent part of the fuzzy system. This can be done by solving the least squares problem using the pseudo-inverse operation as shown below.

\[ \Gamma = \text{pinv}(X)\theta \quad (8.6) \]

Initially, the fuzzy system parameters such as number, location and spread of the fuzzy AF’s are decided intuitively. As a result such a system is not very accurate and to achieve higher accuracy these parameters are required to be tuned. There exist optimal values of these fuzzy parameters for which the model can emulate the system with finer accuracy. Hence, to minimize the inference error from the fuzzy model, it is required that these parameters be optimized within their limiting values. Thus, it is a constrained optimization problem involving multiple variables which are non-linear in nature. Modified GA has been used to optimize fuzzy parameters in the present research and its methodology is discussed while solving the FK problem in Chapter 4. The improved performance of MGA over conventional GA and other numerical approaches has also been emphasized. Therefore to identify fuzzy parameters of TS model, MGA method developed and explained in Section 4.4.3 has been used again.

8.4.4 Optimization of TS Fuzzy System

Experimental data obtained from the actual system is used in optimization and the MSE of pressure values from the experiment and the model is considered as the objective function. The optimization problem is formulated and the objective function derived is written as (8.8).

\[ H = \frac{1}{N} \sum_{p=1}^{N} \left[ \frac{1}{2} (|P_r| - |P_p|)^2 \right] \quad (8.7) \]

\[ \text{Maximize} \quad \text{Fitness} = \frac{1}{1 + H} \quad (8.8) \]

Here H is the mean squared error between the reference signal \( P_r \) (from experiments) and the signal obtained from the fuzzy model \( P_p \) for the same set of input data \( (F, L, \Delta L) \). Since MGA can only be used to maximize a fitness function, the minimization objective (8.7) is converted into a maximization problem and presented as (8.8). Ahead of the optimization, the experimental database is divided in two parts and these data segments are used to train and validate the fuzzy model. Various steps used in MGA algorithm are similar to those discussed in Section 3.5.4; however the set of parameters of MGA used in the present problem differ slightly, and these are listed below.
Population Size: 100
Length of a binary solution: 216 bits (12 bits for each of the 18 fuzzy parameters discretizing the solution space up to the order of $\frac{1}{2^{12}} \approx 10^{-4}$).
Crossover probability: 0.95
Mutation probability: 0.02
Termination criterion: 100 iterations

Size of the population is kept at a moderate number of 100, because fewer solutions in the population may result in localized search and may take long time to converge. Further, it is found during optimization that the value of the objective function does not change significantly in the neighbourhood of a solution. This may be due to the smooth variation of pressure against length and force values (Figure 8.12). Larger population of solutions in the light of above finding may not benefit the optimization accuracy and therefore a moderate number of solutions selected to start the optimization. The solution space is discretized and a solution accuracy of $10^{-4}$ derived from a 12 bit binary number is considered sufficient for the present problem.

![Figure 8.12: Surface plot showing relationship between pressure, position and force on PMA.](image)

Crossover and mutation probabilities are kept similar to the ones used previously in Section 4.4. Termination criterion is a vital parameter to decide and due to lack of prior knowledge on accuracy from the fuzzy model, number of iterations is taken as the stopping criterion for the MGA. Consequent upon several trails, it is found that after approximately 50 epochs the algorithm stopped to converge therefore the algorithm is terminated after 50
epochs in the final experiment. The TS fuzzy model obtained after this MGA based optimization has been shown in Figure 8.13.

![Graph showing pressure prediction](image)

Figure 8.13: Antecedent fuzzy variables used in the TS fuzzy modelling after MGA based optimization.

The optimized TS fuzzy model was evaluated using training and testing datasets. Results of pressure prediction from this model are displayed in Figure 8.14 and Figure 8.15. Apparently, the model shows very close agreement to the training data values however it fails to map the testing data and predicts erroneous pressure values intermittently.

One of the possible reasons for large testing errors could be the overfitting of training data by this model. However, as has been stated earlier, TS fuzzy model lacks interpretability and thus large errors for the testing data is anticipated. One of the major outcomes of this research is the establishment of the fact that in uncertain and ambiguous environments, the TS fuzzy models lose prediction accuracy and though they can provide higher accuracies for the training data, they cannot interpret testing data (having slightly different bounds) with similar accuracy. Prediction errors for training and testing data bases are computed and displayed in Figure 8.15. This illustration provides further insight into the magnitudes of errors. While MSE for the training data is 0.0039, it is found to be 0.9864 for the testing data.
8.4 Takagi-Sugeno Fuzzy System

At the same time maximum deviations for the training and the testing data are found to be 0.2451 bar and 10.39 bar. Clearly, this model cannot be accepted for the dynamic modelling of PMA.

![Figure 8.14: Trajectory following response of TS fuzzy system using (a) training and (b) testing data.](image)

Alternative actions could be to acquire a larger database to be trained with TS fuzzy model or use Mamdani based fuzzy model instead. Acquiring and using a large data base to train the fuzzy model shall increase the computational complexities and yet it is difficult to ensure that while in actual use the model will not be subjected to inputs falling outside the bound within which the model has been trained. Therefore another type of fuzzy model propounded by Mamdani [184-186] was developed and optimized for use in the present problem as explained in the following Section.

![Figure 8.15: Trajectory following errors from TS fuzzy system using (a) training and (b) testing data.](image)
8.5 Mamdani Fuzzy System

Fuzzy systems based on Mamdani inference also known as *linguistic fuzzy models* are mainly employed as fuzzy controllers. Dynamic systems, exhibiting uncertain and ambiguous characteristics, can be effectively controlled using Mamdani fuzzy controller. Over last three decades this controller has been successfully implemented in multitude of applications in engineering and science [187]. Contrary to the TS fuzzy system, Mamdani fuzzy system shows better interpretability. However, as stated before, Mamdani fuzzy inference mechanism lacks accuracy of prediction wherein TS fuzzy inference excels. Various building blocks of Mamdani Fuzzy system are explained here emphasizing their constructional details in context to the present problem.

8.5.1 Fuzzification

Fuzzification process to translate crisp variable into fuzzy variables is similar to the one mentioned in Section 8.4.1. As before three Gaussian AF’s, linguistically named as ‘Low’, ‘Medium’, and ‘High’ were used to represent each of the input fuzzy variables (Figure 8.10). Here, one of the major distinctions from TS fuzzy model was that the output variable of the system was also defined as fuzzy variables. Therefore pressure, which is the output or the control variable in the present problem, was defined as fuzzy variables using five AFs (Figure 8.16). Higher number of AFs was chosen to provide better discrimination while predicting pressure values.

![Figure 8.16: Fuzzification of the pressure as a consequent variable.](image)

Two parameters were required to define the position and spread of a Gaussian AF. Therefore while developing Mamdani FIS a total of 28 parameters were needed to define a total of 14 AFs, 9 for the three antecedents and 5 for the pressure which is a consequent
variable. However, initial positions were chosen as before by equally dividing the universe of discourse of individual variables to accommodate all the AFs. To begin with, the spread of these AFs was again chosen as explained in the previous Section 8.4.1.

8.5.2 Rule-base

Rule-base structure of Mamdani fuzzy systems is again similar to the TS fuzzy systems with the only difference in its consequent part as shown below.

\[
\text{if } F \text{ is } A_{i1} \text{ and } L \text{ is } A_{i2} \text{ and } \Delta L \text{ is } A_{i3} \text{ then } P \text{ is } B_i
\]

It is important to note here that the consequent part is also a fuzzy variable and not a real number as in the case of TS fuzzy systems. The total number of rules is still governed by the antecedent part of the rule-base and hence once again can be given by (8.1).

8.5.3 Inference Engine

Function of the inference engine is, to provide model output for a given set of input variables. The Mamdani fuzzy model output was computed by working together with the input values and the consequents from related rules. Three kinds of inference mechanisms were proposed in the literature namely, Zadeh max-min inference, Mamdani min-max inference and Larson product inference [156]. Out of these the Mamdani min-max inference is more popular owing to its simplicity in software and hardware implementations [154]. The Mamdani inference was used in the present work and various steps involved in its implementation are explained below.

Initially the crisp set of input variables was converted into respective fuzzy variables. The fuzzy operator ‘\text{min}’ was applied on these fuzzy variables to compute the fuzzy value of the consequent variable for all the rules. Subsequently, available outputs from the rules were aggregated and then defuzzified to provide a crisp output value. Mathematically this process is explained using following equations.

The inference result \(\mu_{\hat{B}_i}(P)\) was computed applying a ‘\text{min}’ operator on the input values and the fuzzy rules.

\[
\mu_{\hat{B}_i}(P) = \mu_{A_{i1}}(F) \land \mu_{A_{i2}}(L) \land \mu_{A_{i3}}(\Delta L) \land \mu_{B_i}(P) \tag{8.9}
\]

Here \(\land\) stands for ‘\text{min}’ and \(\mu_{A_{i1}}\) etc. are the activation values for the individual input variables in \(i^{th}\) rule. The final value of the fuzzy consequent variable was found by applying a ‘\text{max}’ operator to the individual rule outputs as shown below.
8.4 Mamdani Fuzzy System

\[ \mu_{B}(P) = \mu_{B_1}(P) \lor \ldots \lor \mu_{B_N}(P) \]  

(8.10)

Here \( \lor \) stands for ‘max’.

The final output is a fuzzy variable and hence cannot be used in real time application unless converted into a crisp real number. The process of translating fuzzy variables into real numbers is opposite to the fuzzification process and hence termed as defuzzification. There exist several defuzzification methods proposed by researchers and these methods can be broadly classified as methods based on the geometry of the fuzzy set [188] and methods using statistical interpretations [189]. Amongst these the centroid or the center of gravity method is often used due to its simplicity and easy hardware implementation. Centroid method was used in the present work to convert the fuzzy output \( (P) \) in a usable crisp form \( (P_0) \) as explained below.

\[ P_0 = \frac{\int P \mu_{B}(P) dP}{\int \mu_{B}(P) dP} \]  

(8.11)

To avoid the complex computation involved in evaluating above integral a simplified form was used here.

\[ P_0 = \sum_{i=1}^{N_r} P_i \frac{\mu(P_i)}{\sum_{i=1}^{N_r} \mu(P_i)} \]  

(8.12)

Here \( P_0 \) is the crisp output for pressure from the model, \( P_i \) are the fuzzy outputs from individual rules and \( \mu(P_i) \) represent their activation values.

8.5.4 Optimization of Mamdani Fuzzy System

The initial design of the Mamdani fuzzy system cannot guarantee good accuracy and thus its parameters were required to be optimized. Modified GA was used afresh owing to its better performance over other methods. Total number of antecedent and consequent parameters in the Mamdani fuzzy system developed for the problem of PMA modeling was 28 as discussed before. Apart from the antecedent and consequent parameters, the rule-base in the Mamdani fuzzy system was also required to be optimized to achieve better accuracy. The consequent pressure variable was represented using five AFs; hence each rule shall have one of these five AFs as a consequent. The rule outputs or the consequents were considered as parameters to be identified and were optimized using MGA. Extra 4-bits were added to the binary solution strings to provide rule outputs \( (B_i) \). The length of a binary solution string in
8.4 Mamdani Fuzzy System

this case was 340, out of which $28 \times 12 = 336$ bits were kept for the optimization of the AF parameters and remaining four bits were used to provide information about the rule outputs.

\[ i = 1 + \sum_{j=337}^{340} b_j \]  

(8.13)

These four bits have values between 0000 and 1111 and to get rule output information their sum was augmented by one as shown in (8.13). Rule output was given by $B_i$ where $i=1$..5 and the index of an AF were found using (8.13). An example binary solution is shown below for further elucidation.

\[ b_j = \begin{array}{cccc} 1101 \ldots 1 & 1010 \ldots 0 & 1011 \ldots 1 & \ldots \ldots & 1011 \ldots 1 & 1001 \end{array} \]

Rule output

Specifications of the MGA used are given below.
Population Size: 100

Length of a binary solution: 340 bits (12 bits for each of the 28 fuzzy parameters and 4 bits for the rule outputs).

Crossover probability: 0.95; Mutation probability: 0.02; Termination criterion: 100 iterations.

Figure 8.17: Antecedent fuzzy variables used in the Mamdani based fuzzy modelling after MGA based optimization.
These specifications were decided as discussed in Section 8.4.4. Consequent upon the optimization, fuzzy model parameters were obtained which were used to construct activation functions of antecedent and consequent variables of the Mamdani fuzzy model. The optimized designs of the antecedent and the consequent variables of the Mamdani fuzzy model have been provided in Figure 8.17 and Figure 8.18.

![Figure 8.18: Consequent fuzzy variable (Pressure) used in the Mamdani based fuzzy modelling after MGA based optimization.](image)

Finally the results obtained from the optimized Mamdani fuzzy model to predict pressure values, while using training and testing data, are displayed in Figure 8.19 and Figure 8.20.

![Figure 8.19: Trajectory following response of the Mamdani fuzzy system using (a) training and (b) testing data.](image)
8.4 Mamdani Fuzzy System

Figure 8.20: Trajectory following errors of the Mamdani fuzzy system using (a) training and (b) testing data.

Above illustrations show a close agreement of the model predicted pressure values with the training as well as the testing data pressure value. Error exists at the points where the trajectories are changing and this error is mainly due to the changed behaviour of PMA resulting in the slow response against the excitation. The external force on PMA in the test rig was steadily increased during the course of entire trajectory and it has not affected the model performance adversely.

Figure 8.21: Hysteresis in PMA closely modelled by Fuzzy modelling.

The MSE of the model predictions were found to be 0.0043bar and 0.0013bar for the training and testing data respectively. Further, MSE for the training data is slightly more than the one obtained using TS fuzzy model, nevertheless MSE in the testing data is far small which justifies the selection of Mamdani fuzzy model.
Maximum deviation from this fuzzy model is found to be 0.2464bar and 0.3273bar respectively for the training data and the testing data. Hysteresis present in the PMA due the presence of a latex tube is a major source of error which is quite apparent from Figure 8.4 and Figure 8.5. There were two sources of hysteresis in the PMA, such as hysteresis due to internal friction of the rubber tube during actuation and on part of the thermal exchange with the environment while the actuator is stretched and relaxed. At higher operating frequencies the interaction time of the PMA with the environment decreases and hence the hysteresis effect becomes more pronounced. This inference is further supported by the experimental finding as discussed later in Section 8.7. Nevertheless apart from the effect of higher operating frequency, the proposed fuzzy model was able to model the hysteresis effectively. The illustration provided in Figure 8.21, shows the accuracy of the proposed model in the face of existing hysteresis, wherein a cycle of inflation and deflation of the PMA has been closely followed by the model.

The difference, which was observed from the optimized fuzzy activation function distribution of both the fuzzy models, was that the standard deviation of the AFs in TS fuzzy model was quite large compared to the Mamdani fuzzy model. Thus, an important inference which can be drawn is that, the large standard deviation helps in more accurate predictions; whereas small standard deviation facilitates universal approximation or interretability of a fuzzy model at the expense of its accuracy.

Finally the proposed fuzzy model was used to develop a position controller for the wearable ankle robot and various implementation steps are discussed in the following Section.

## 8.6 Fuzzy Controller

The wearable robot is intended to perform range of motion and muscle strengthening exercises. This requirement can be achieved by controlling lengths of the four actuators and bringing the end-effector in a required pose. The end-effector pose is acquired by the coupled motion of its actuators and therefore inverse kinematics relation can be used to find the required link vector comprising of actuator lengths to achieve a desired end-effector pose [18]. However in the present case, where PMA were used, a controller working on this scheme cannot be accurate. To further understand the inaccuracy anticipated the characteristics of PMA displayed in Figure 8.4 and Figure 8.5 will have to be analyzed. It is quite apparent from these figures that the characteristics were non-linear and time varying
and depending on whether the actuator is inflated or deflated, there exists two different lengths for a given pressure. Moreover, these characteristics were also severely affected by the external force on the PMA. Now, if the length of these actuators is controlled, it is obvious that orientations of the end-effector will be achieved with different link velocities. The actuator shall behave sluggishly when the external force is large and eventually take more time to achieve a target. Since the simultaneous actuation of four PMA results in the desired end-effector orientations, different velocities of individual actuators shall result in the orientation errors along with large trajectory following errors. Further, to realize muscle strengthening exercises on the robot, a force control strategy will have to be implemented which can only be achieved by controlling pressure in the actuator. To author’s best knowledge, PMA have not been employed before in parallel robots working under external forces. One of the major contributions of the present research is in the field of dynamic modelling of PMA wherein the actuation is controlled taking into account the external force, the gradient of actuator length and the pressure in the actuator.

In the light of above discussion it can be concluded that the inverse kinematics approach, wherein the actuator lengths were controlled, cannot be used in the present case as this will lead to trajectory following errors. Hence pressure was considered as a control variable and actuator lengths were controlled by appropriately inflating or deflating them with requisite pressure.

Figure 8.22: Fuzzification of the antecedent and consequent variables of FLC2.
The Mamdani fuzzy model developed for PMA in the previous Section was used in this controller and is called as FLC1 henceforth. The actuators used in the wearable robot were all identical and were similar to the PMA which has been used to develop the proposed fuzzy model. Furthermore, the arrangement of actuators in the robot, as discussed before, was identical to the way they were arranged in the experimental set-up of a single PMA while performing dynamic modelling. Once the FLC1 which is also an inverse model [190] of PMA, has been developed, the next step was to use this FLC1 in implementing an overall fuzzy control scheme as shown in Figure 8.23. A control scheme based on an open loop feed forward controller cannot be considered as stable. Moreover rehabilitation application involving human user has higher safety requisites thus a closed loop feedback controller was preferred over an open loop controller.

Another fuzzy controller (shown as FLC2 in Figure 8.23) which is also based on Mamdani inference mechanism was developed to compensate a possible error in the FLC1 or sudden variation in the environment and was placed in the closed loop feedback as shown in the same illustration. The FLC2 has two inputs which are error in actuator length and the gradient of this error. Gradient was chosen over integration in view of the PMA characteristics which were sensitive for the gradient of actuation (Figure 8.4).

<table>
<thead>
<tr>
<th>P</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZE</td>
<td>ZE</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

Output from this controller (FLC2) was pressure similar to the FLC1. The main building blocks of this fuzzy controller i.e. activation functions of the antecedent and consequent variables and the rule-base are shown in Figure 8.22 and in Table 8.2 respectively. This rule-base was developed intuitively as a standard practice after analyzing the database obtained for a single PMA [157]. The fuzzy variables were represented by five Gaussian AFs, linguistically defined as Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS) and Positive Big (PB). The universe of discourse of these variables was decided manually after analyzing the available data from the experiments. Second derivative of error was not been considered because it did not provide useful results.
The overall control scheme of the proposed robot along with the placement of the fuzzy controllers is shown in Figure 8.23. System configuration of the robot along with the placement of hardware facilities and software routines with the graphical user interface has been shown in Figure 8.24. The desired orientations of the end-effector were converted into required sets of link lengths using the inverse kinematics module. The vector of required link lengths along with force sensor readings was given to the FLC1 whereas the desired link lengths and the vector of link lengths obtained from the linear potentiometer was given to FLC2 as inputs.
Output from both FLC was pressure which was aggregated and converted in terms of duty cycles (using PWM) to be provided to the individual pressure valves. This information was passed onto the valves through the DSPACE processing system. Appropriate pressure was applied to the individual PMA and the required orientation of the end-effector was achieved. Observations from the sensors for position, force and pressure were read through DSPACE system. Force information was used by the FLC1 whereas position information was shared by FLC2 and the FK module. Position or link length information was converted into end-effector pose using FK module and this information was used to find the error in the end-effector orientation.

8.7 Experimental Evaluation of the Fuzzy Controller

In order to evaluate performance of the fuzzy controller, applied to the wearable robot, two kinds of experiments were carried out. Initially the fuzzy control scheme using pressure as a control variable was evaluated in terms of its feasibility in the rehabilitation process. This includes evaluation of the response of PMA to different input signals, trajectory following errors and robustness against operating frequency. In the second stage, experiments were performed involving a healthy human subject with the wearable robot and basic range of motion exercises were carried out to evaluate the potential of wearable robot in ankle rehabilitation treatments.

8.7.1 Performance Evaluation of the Fuzzy Controller

First of all, the robot was subjected to a step signal in terms of link lengths to check the response of group of PMA for the input signals. Response to a step input provides two sets of information such as maximum velocity of the robot and its overall stability. The maximum velocity of the robot motion from the experiment was found to be approximately 0.0417m/s. Since the rehabilitation treatments are normally carried out at a slower rate, the present robot velocity can be taken as acceptable. Further, it is evident from Figure 8.25 that there’s no overshoot or ringing of system response about the steady state signal, hence, the robot may be considered stable for a step input signal. Robot’s response and resulting error for the input step signal were displayed in Figure 8.25 and Figure 8.26. It is important to note here that the robot was not subjected to any external force/torque during the first stage of experiments.
Subsequently, in order to closely mimic the rehabilitation trajectory of the ankle joint, the robot was subjected to two types of input signals for link lengths namely, a ramp signal and a sinusoidal signal. To study the effect of change in the input signal frequency on the PMA response, these experiments were carried out at frequencies of 0.1Hz and 0.2Hz.
The robot was observed to be closely following the ramp trajectory at 0.1Hz with a small lag of less than one second. This lag occurs mainly due to the time spent in filling the PMA with air. This was called the response time of PMA and the average response time from the experiments was found to be 0.8sec to achieve 90% of the steady state signal value (Figure 8.27). While using the PMA, response time becomes a vital parameter to ascertain. Due to the presence of a visco-elastic element and compliant operating agent (air in the present case) the response of PMA cannot be comparable to the motors.

Figure 8.27: Trajectory following response of individual actuators for a ramp input signal (0.1Hz).

Figure 8.28: Trajectory following errors of individual actuators for a ramp input signal (0.1Hz).

Providentially, rehabilitation applications involve smooth and slow motions and as such some response delays were admissible. Error in the input signal and the actual linear
potentiometer observations for a step input signal were also shown with the help of an illustration (Figure 8.28). While the largest error observed was less than 8mm, errors do not show a specific trend. However, it was seen that, the error pattern was same for all the actuators. When the same input signal was applied to the robot at 0.2Hz, the trajectory following error was increased to two times the error observed in the previous experiment.

Figure 8.29: Trajectory following response of individual actuators for a ramp input signal (0.2Hz).

Figure 8.30: Trajectory following errors of individual actuators for a ramp input signal (0.2Hz).

As can be seen from Figure 8.29, a lag of approximately 0.25sec exists in the actual signal and the reference signal. This increment in the trajectory following error at higher operating frequency may be due to the pronounced hysteresis effect of the PMA as discussed.
earlier in Section 1.1. This was a useful inference drawn from this study and can be used for advanced PMA modelling tasks in future. Increased trajectory following at higher operating frequency error is seen in Figure 8.30.

The wearable robot was further subjected to a sinusoidal trajectory in the joint space which closely represents the actual rehabilitation trajectory of the ankle joint [144]. Experiments were performed at 0.1Hz and 0.2Hz frequencies and results for robot’s response were displayed in Figure 8.31 and Figure 8.33. Close trajectory following response of the robot was observed for both the cases with slightly larger errors at higher frequency (Figure 8.34).

A close inspection of these results reveals that the trajectory errors were more pronounced when the lengths were increasing which further mean that deflation part of the PMA actuation cycle produces larger errors compared to the inflation instance. Since the hysteresis effect of PMA comes into play during deflation of elastic muscle, this observation further confirms the previous assertion wherein the hysteresis was considered to be the main reason for the trajectory following errors. The error in the reference sinusoidal signal (0.1 Hz) and actual actuator positions is displayed in Figure 8.32. After an initial disturbance, a cyclic trend with a maximum error of 6.5 mm is seen in the individual actuator position errors. Marginally large errors while inflating PMA is clearly visible in this illustration.
8.7 Experimental Evaluation of the Fuzzy Controller

Figure 8.32: Trajectory following errors of individual actuators for a sinusoidal input signal (0.1Hz).
Actuator1 and Actuator2 were actuated using similar reference trajectories; however they show different positional errors during inflating wherein Actuator2 showed slightly higher error than Actuator1. This may be due to the modest difference in the individual muscle characteristics, thus it is recommended that in the future, different model be developed for individual actuators before using them in the overall fuzzy controller.

When the same sinusoidal signal was applied at slightly higher frequency (0.2Hz), an increased lag in the response of PMA was observed. However the maximum tracking error was found to be 8mm which was approximately 6% of the actuation stroke and was considered to be small. Tracking errors exhibited by Actuators 2 and 3 were slightly larger than their counterparts and a probable cause for this anomaly is the slight difference in their construction. This error difference can be corrected by developing different models for individual actuators as discussed above.

Figure 8.33: Trajectory following response of individual actuators for a sinusoidal input signal (0.2Hz).
8.7 Experimental Evaluation of the Fuzzy Controller

Two important conclusions can be drawn from the above experiments and subsequent analysis of the responses and the tracking errors. Firstly, the controller was able to closely track the trajectories during inflating the muscles but a delayed response was seen during deflation. Secondly, the tracking error increased with an increase in the excitation frequency. It was learnt that since the error in trajectory tracking was due to the presence of elastic hysteresis, it was only visible while deflating the PMA. Furthermore, this phenomenon becomes more pronounced when the PMA were speedily filled up and exhausted.

The proposed fuzzy controller has been developed mainly to investigate the use of pressure as a control variable in the present research. It is clear from the above experimental findings that the pressure can be successfully controlled to achieve required trajectories of robot motions. It is important to note here that the involvement of the external environment with the robot was not the scope of the present work and hence this has not been modelled. However experiments were performed with a human user as described in the following Section to gather some initial data and information and make future recommendations.

8.7.2 Experimental Evaluation of the Fuzzy Controller with a Healthy Subject

The wearable robot was finally used with a healthy subject in order to analyze the controller performance while interacting with the external environment. These experiments were carried out to draw some important recommendations for the future work wherein mainly the human-robot interaction shall be studied. In order to emulate the rehabilitation process, three trajectories normally recommended by the therapists were employed. These trajectories were dorsiflexion-plantarflexion, inversion-aversion and adduction-abduction motions as discussed in Chapter 3.
8.7 Experimental Evaluation of the Fuzzy Controller

Figure 8.35: Trajectory following response of individual actuators for dorsiflexion-plantarflexion trajectory while the robot is being used by a healthy subject in passive mode.

Figure 8.36: Trajectory following response in Euler angles for three types of trajectories, namely, (a) dorsiflexion-plantarflexion, (b) Adduction-abduction and (c) inversion-aversion; and trajectory following errors (d) while the robot is being used by a healthy subject in passive mode.

To begin with, the right foot of the subject was secured in the end-effector of the wearable ankle robot conveniently and the user was asked to sit comfortably. The passive mode of the rehabilitation treatment was investigated wherein the user was asked not to exert any force and remain relaxed while the robot was commanded to implement the three trajectories mentioned above.
The desired orientations of the end-effector for the commanded trajectories were $\pm \pi/3$, $\pm \pi/6$, $\pm 5\pi/36$ radians for the dorsiflexion-plantarflexion, inversion-aversion and adduction-abduction motions respectively.

Desired actuator lengths were computed using inverse kinematics and were plotted along with the resulting actual link length information obtained from the sensors in Figure 8.35. A small segment representing the complete data was plotted here for preliminary analysis. Initially all the actuators were at their mean position which is shown as 0.1125m. Trajectories shown using dotted lines are the desired values for link lengths whereas the actual link lengths recorded from the sensors is shown using solid lines. Large errors for the link lengths were observed and the maximum deviation of 0.02406m was recorded. Trajectory following errors can be seen showing an increasing trend which is not desirable. Later, using FK module, the actual end-effector orientations were computed and plotted for the three trajectories separately along with the desired orientations as shown in Figure 8.36.

The unpredictable trajectory following response and the irregular error pattern was analyzed and the external environment was identified as the main source of error. While developing the fuzzy controller, the external environment such as the inertia, stiffness and damping of patient’s leg and foot was not accounted. This explains the inconsistent pattern of the errors while the robot was used with a human subject. Involuntary application of force on the wearable robot by the subject is another possible source of trajectory error.

### 8.8 Chapter Summary

This Chapter presented two major aspects of wearable robot actuation and control. While analyzing the actuation aspect, dynamic modelling of the PMA was performed using artificial intelligence approaches. Artificial neural network and two variants of fuzzy logic inference were used to model the non-linear behaviour of PMA. Both fuzzy models, namely, Mamdani fuzzy model and TS fuzzy model were optimized using MGA approach. It was found through simulation that since the interpretability was an important issue in the present modelling work, a Mamdani based fuzzy model was a better choice over ANN and TS fuzzy model. Subsequently, an iterative fuzzy logic controller was developed which incorporated the earlier developed and optimized Mamdani fuzzy dynamic model of PMA. The fuzzy controller, developed for the position control of the wearable robot was implemented on the actual robot prototype and results for two kinds of experiments were recorded. Initially motion trajectories were implemented on robot without any intervention from the
environment and the results for various trajectories at two different operating frequencies were obtained. The performance of the proposed iterative fuzzy controller was evaluated against these trajectory inputs. The accuracy of the dynamic model and closed-loop tracking performance were found to be acceptable. Performance of the proposed controller on the wearable robot was later investigated with a healthy subject using the robot. It was observed that the controller was able to compensate for the external force exerted by the subject with an error, of the order of 0.2 rad.
Chapter 9 Conclusions

The principal goal of this research was to investigate various research issues concerning the development of wearable rehabilitation robots in the pretext of an ankle rehabilitation robot. The research was carried out to meet the growing need for an automated rehabilitation solution for the benefit of therapists, patients and the rehabilitation process at large. This research proposed a parallel mechanism based wearable robot design to provide requisite motions and force trajectories for rehabilitation treatments of the ankle joint. Research objectives were drawn from the challenges posed by the wearable robot for its design, modelling, actuation and control. This Chapter summarizes the important conclusions drawn from the entire research work, carried out during development of the proposed robot. Major contributions extended to the existing knowledge have also been highlighted in the following Sections of this Chapter.

9.1 Major Outcomes and Contributions

One of the major outcomes of this research was the conceptualization and development of a wearable design for the ankle rehabilitation robot, which enjoyed several benefits over the existing ankle robots as discussed in Chapter 3. While carrying out system modelling of the subject robot, the difficulty in obtaining a closed form solution for the FK was revealed. Consequently, a new fuzzy logic based model was proposed which was found to be time efficient and more accurate when compared to the Newton-Raphson approach (which is used by most of the past researchers) and other types of fuzzy models [23]. A complete design analysis of the wearable ankle robot was carried out as a next step to identify vital performance indices to evaluate robot design from the kinematics, actuation and structural aspects. Mathematical formulations of these performance indices were derived and important robot design parameters were selected to perform optimization of the robot design. Following the past practice, the optimization was initially performed considering the GCN as the sole objective. Modified genetic algorithm, developed during this research, was used as an optimization tool. It was found that the single objective approach was not able to optimize all the performance indices and thus a multi-objective optimization of the robot design was recommended.
Consequently, the robot design optimization was performed using existing multi-objective optimization methods, namely, the preference based optimization and the evolutionary algorithm (EA) based optimization. Owing to the conflicting nature of objectives, their large number and continuous solution space, these two existing optimization methods were found to be ineffectual and therefore further investigations in the optimization methodologies were carried out. A new fuzzy dominance based evolutionary optimization method was developed, in the present research, as a major breakthrough, to address the shortcomings of the existing EA approaches. The new method was named as fuzzy sorting genetic algorithm (FSGA) and the robot design optimization was carried out using this new approach. After successful design optimization, the wearable robot was constructed. Pneumatic muscle actuators were found suitable to be used in wearable robots due to their high power to weight ratio and skeletal muscle like behaviour. However, PMA were found to exhibit non-linear and time dependent behaviour which was required to be modelled ahead of designing an appropriate control scheme for the wearable robot. Consequently, a Mamdani based fuzzy model was developed and optimized to accurately predict the PMA behaviour in the presence of an external force. The fuzzy model of PMA was finally incorporated in an overall fuzzy controller designed for the posture control of the wearable robot. Further details on the above research outcomes and contributions are being provided in the following subsections.

9.1.1 Development of a Wearable Ankle Rehabilitation Robot Design

A new wearable ankle rehabilitation robot, based on a parallel mechanism, was proposed for the first time as an outcome of the present research. The biologically inspired design, which is compact, wearable and light weight, is capable of providing the required ankle motion and force trajectories while maintaining kinematic compatibility with the ankle joint. While using the proposed wearable robot, the shinbone of patients and position of the ankle joint remains stationary, thus allowing better rehabilitation treatment by accurately following prescribed trajectories. Moreover, owing to such configuration, the robot can offer repeatable rehabilitation motions and can be used as a measurement tool for evaluation of the human ankle joint characteristics. Actuators used in this robot are backdrivable, compliant and have force-deformation characteristics close to the skeletal muscle, hence are good candidate for the wearable robot actuation. On the other hand the existing ankle rehabilitation robots did not have a bio-inspired design which causes inconvenience to the user and poses control related problems to the designer. Besides incorrect design, previous ankle robots were heavy, immobile and employed huge actuators which were non-compliant, non-backdrivable and
had an intimidating appearance hence were less likely to be acceptable to patients. Higher cost, safety and aesthetics were yet other concerns which limit common use of these existing ankle robots. Challenges regarding design, actuation and control of wearable rehabilitation robots were identified and addressed during the course of this thesis.

9.1.2 Forward Kinematics Solution for the Wearable Robot

Fuzzy logic based computational model was proposed to carry out forward kinematics (FK) analysis of the wearable ankle robot. Takagi-Sugeno based fuzzy logic algorithm which is less computationally expensive and can provide accurate results, was investigated and as many as eighteen fuzzy models were developed. A Pareto analysis of these fuzzy models was carried out based on the accuracy and computational time offered. Finally, a fuzzy model providing best accuracy was selected which also showed improved time efficiency over the numerical methods.

The Takagi-Sugeno based fuzzy models used in solving the FK problem were all optimized using three approaches namely, gradient descent (GD) method, genetic algorithms (GA) and modified genetic algorithms (MGA). The MGA was developed during this research in order to enhance the local search capabilities of GA. There were two modifications suggested in the GA, one of them was the use of an elitist approach [109] to ensure that the best solution from a particular generation does not become extinct in the evolution process. A two step selection process was proposed as a second amendment to the conventional GA, which enhances its local search capabilities. Firstly, a near optimal solution was obtained using the conventional selection approach, known as the roulette wheel selection method. Thereafter, a gradient-based selection method, which has been developed and proposed for the first time, was employed to fine tune the near globaloptimal found by the conventional GA. This MGA has been used more than once in this thesis to optimize parameters of fuzzy models for enhanced accuracy.

Given the fact that most skeletal joints are actuated by parallel action of muscles in the human body, wearable rehabilitation robot designs are likely to be based on parallel mechanisms. It is also known from the literature that forward kinematics solution of parallel mechanisms is difficult to obtain and this is a research question since more than two decades [21, 22, 191]. Existing methods to solve FK were time consuming and lack accuracy, hence could not be used during real time control applications of wearable robots. Therefore, there was a need to develop computational model for the solution of FK which is fast and accurate.
9.1 Major Outcomes and Contributions

9.1.3 Design Analysis and Multi-objective Design Optimization of the Wearable Robot

One of the major contributions of this thesis was in the design optimization of the wearable ankle robot. A fuzzy dominance based evolutionary optimization approach was proposed to carry out multi-objective optimization of the wearable robot designs. The proposed approach was able to optimize a large number of objectives simultaneously while the solution space was continuous. Existing evolutionary approaches such as NSGA II (non-dominated sorting genetic algorithms) were found to be inefficient in optimizing wearable robot designs.

To begin with, six important performance indices (PIs) were identified encompassing three main aspect of the robot design namely, kinematic design, actuation design and structural design. During the design analysis, it was also established that the robot’s performance was largely dependent on its geometrical design parameters, such as, positions of the actuator connection points on the two parallel platforms and the distance between the platforms. It was further noted that all the PIs were interdependent and could be related to the GCN of the robot. It was expected that by optimizing the condition number, the Jacobian singularities and actuator forces can be reduced whereas the workspace and stiffness of the robot can be increased. Consequently, the design optimization was carried out considering the GCN as a single objective to be optimized, as per the past practice [97, 166]. The MGA, discussed above, was used to perform the optimization. Analyzing the optimization results, it was found that using a single objective approach, it was not possible to optimize all the performance indices and hence a multi-objective optimization of the robot design was recommended. Thereafter, existing multi-objective optimization methods, namely, preference based optimization and the evolutionary algorithm (EA) based optimization, were used to perform the robot design optimization. Weighted average approach was used for the preference based optimization and NSGA II was used for the EA based optimization. Outcomes from these two optimizations were analyzed and it was found that due to the conflicting nature of objectives the preference based method failed to provide an optimal set of PIs. Similarly due to a large number of objectives and continuous solution space, the NSGA II was found to be inadequate and therefore further investigations in the optimization methodologies were carried out. As a result of further research, a fuzzy dominance based evolutionary optimization method named as fuzzy sorting genetic algorithm (FSGA) was developed. This new evolutionary optimization approach was able to address various shortcomings of the existing NSGA II discussed in Chapter six. Using hand calculations it was shown that the proposed fuzzy dominance concept effectively replaces the non-
dominance concept. It is important to mention here that the non-dominance criterion has been in use for more than two decades and has been applied to a wide range of optimization problems [192].

Design optimization of the proposed wearable ankle robot was difficult in the light of its wearability requirement, parallel actuation, use of cables and flexible PMA, and its application in ankle rehabilitation treatments. Wearability requirement from the proposed robot put forth conditions such as, light weight, compact design, comfortable in use, safety and portability. Parallel action of actuators restricted the available workspace of the robot and posed singularity related issues. Cable based actuation required that the robot trajectories be achieved through positive actuator forces and the stiffness of the robot be analyzed taking into account its rigidity. Finally, application of wearable robot in the ankle joint rehabilitation, set higher actuator force requirement and design constraints arising from its use by subjects of varying physical abilities. In the light of above constraints it was essential to carry out a multi-objective optimization of the robot design to establish an optimum trade-off between desired objectives.

Design solutions, for the wearable ankle robot, obtained from preference based optimization, NSGA II and proposed FSGA method were compared and a solution provided by FSGA was accepted for the construction owing to its better results.

9.1.4 Actuator Modelling and Fuzzy Logic Based Position Control

A dynamic model of PMA, in the presence of an external variable force, was developed using Mamdani based fuzzy inference system. Initially, three models, namely, artificial neural network, Takagi-sugeno fuzzy model and Mamdani fuzzy model were developed and their performance in terms of accuracy and interpretability was evaluated. The fuzzy models were optimized using MGA, which has been discussed in Chapter 4. Given the non-linear and time dependent behaviour of these actuators, a Mamdani based fuzzy model, which showed better interpretability, was finally used to model PMA characteristics. The previous modelling work, related to the PMA, had been performed either for a constant external force or without any external loading. It was shown, by analysing experimental results in this thesis, that the external force severely affects the PMA characteristics and it cannot be ignored. Moreover, prediction accuracy for the dynamic behaviour of PMA from previous models was required to be improved to achieve better kinematic compatibility.

In order to carry out motion therapy of ankle joint on the robot, an iterative fuzzy controller was developed using the fuzzy dynamic model of PMA. The fuzzy controller was
implemented on the actual robot prototype, to perform experiments, with the aim of evaluating robot’s performance in ankle rehabilitation treatments. Firstly, the robot end-effector was subjected to a step signal to check the response of the group of PMA in the ankle robot in terms of maximum velocity and overall stability of the robot. The maximum velocity of the robot motion was found to be approximately 0.0417m/s, which is acceptable given the slow motion requirement during rehabilitation treatments. During these experiments the overshoot and ringing were not observed and thus the robot motions were considered stable for the input signal. Subsequently, experiments were performed on the robot without intervention of a human subject and the results were analysed for its positional accuracy. Until this stage, the robot was not subjected to any external force/torque. To closely emulate the rehabilitation trajectory of the ankle joint, the robot was subjected to two types of input signals namely, a ramp signal and a sinusoidal signal. The robot end-effector was found to be closely following both the trajectories with a small lag of less than one second. This lag occurred mainly due to the time spent in filling the PMA which is also called the response time of PMA. The average response time from the experiments was found to be 0.8sec to achieve 90% of the steady state signal value which was acceptable for a slow rehabilitation application.

Final set of experiments was performed while the wearable ankle robot was being used by a healthy subject. This was done to facilitate the analysis of controller performance when the robot interacts with the external environment. Motion trajectories which are normally used by the therapists namely, dorsiflexion-plantarflexion, inversion-aversion and adduction-abduction were implemented. The passive mode of the rehabilitation treatment was investigated wherein the user was asked not to exert any force and remain relaxed while the robot was commanded to implement the requisite motion trajectories. Though the robot was able to follow motion trajectories in the presence of an external disturbance, large and incoherent tracking errors were recorded.

The main source of error identified in the present case was due to involuntary flexions of ankle muscles by the subject and their consequential force on the prototype. The simultaneous control of four PMA to achieve the position control of an end-effector platform in the presence of unknown external forces has been done for the first time to the author’s best knowledge.
9.2 Future Work

This research has developed a wearable robot for the rehabilitation treatment of the ankle joint. Although significant amount of work has been accomplished and the prototype developed can be used for the ankle rehabilitation treatments, few milestones remain to be achieved due to the multi-disciplinary nature and wide dimensions of this project. The proposed wearable ankle robot design requires certain improvement in terms of its conceptual design. One of the main issues compromising its performance is the relatively large friction encountered during its actuation. Design improvements, to decrease overall friction in the prototype, are required to be undertaken. Decreasing friction of the prototype, the overall energy requirement of the robot can be reduced to a large extent. Improvement in the mechanical design in term of material selection is also suggested, which will minimize the aggregated deflection of the robot end-effector in the presence of external loads.

Future research pertaining to the implementation of physical human-robot interaction (pHRI) for the wearable ankle robot is strongly recommended. A software interface is required to be developed which connects the two actors i.e. human and the robot in order to achieve desired scaffolding motions/force trajectories. In other words, during pHRI, the robot is expected to supply auxiliary forces to empower and overcome human physical limits, set naturally or compromised due to an injury. Therefore, physical as well cognitive human robot interaction control strategies are required to be developed to augment future intelligent controllers.

To achieve goals, specific to the ambulatory requirements of wearable ankle robot, further miniaturization of sensors, recorders and processors is recommended. To effectively use the available rehabilitation data which is obtained from various sensors, it is recommended to develop an information management system in thorough consultation with physiotherapists, to convert the acquired data into useful information and help therapists in taking a right decision on further treatments. Development of an information management system is also important to enhance the acceptability of the wearable robot for healthcare professionals, who may not be familiar with the intricacy of robotic controllers.
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Fuzzy models were constructed to solve the FK problem of the wearable ankle robot. Details of the FK problem and construction of these fuzzy models has been provided in Chapter 3. Due to space limitation, these models were not depicted in that Chapter and hence are being provided in this appendix. Constant and linear TS fuzzy models with two, three and four activation functions were modelled. These fuzzy models were later optimized using three methods namely, gradient descent method, genetic algorithm and modified genetic algorithm. Thus in all there were eighteen fuzzy models developed for FK problem. The antecedent fuzzy activation functions for the eighteen fuzzy models have been illustrated below. Abbreviations used to name the models has first digit to show the number of activation functions followed by a letter for the type of TS fuzzy models (whether constant or linear type). The last two/three letters after the type of model have been used for the optimization method.

A.1 Fuzzy Models using Two Activation Functions

Figure A.1: Antecedent fuzzy activation functions for 2CGD
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Figure A.2: Antecedent fuzzy activation functions for 2LGD

Figure A.3: Antecedent fuzzy activation functions for 2CGA
Figure A.4: Antecedent fuzzy activation functions for 2LGA

Figure A.5: Antecedent fuzzy activation functions for 2CMGA
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

A.2 Fuzzy Models using Three Activation Functions

Figure A.6: Antecedent fuzzy activation functions for 2LMGA

Figure A.7: Antecedent fuzzy activation functions for 3CGD
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Figure A.8: Antecedent fuzzy activation functions for 3LGD

Figure A.9: Antecedent fuzzy activation functions for 3CGA
Figure A.10: Antecedent fuzzy activation functions for 3LGA

Figure A.11: Antecedent fuzzy activation functions for 3CMGA
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

A.3 Fuzzy Models using Four Activation Functions

Figure A.12: Antecedent fuzzy activation functions for 3LMGA

Figure A.13: Antecedent fuzzy activation functions for 4CGD
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Figure A.14: Antecedent fuzzy activation functions for 4LGD

Figure A.15: Antecedent fuzzy activation functions for 4CGA
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Figure A.16: Antecedent fuzzy activation functions for 4LGA

Figure A.17: Antecedent fuzzy activation functions for 4CMGA
Appendix A Description of Fuzzy Models Used for Forward Kinematics Modelling

Figure A.18: Antecedent fuzzy activation functions for 4LMGA
References


References


References


References


[181] "[www.dspaceinc.com]."


