Innovative Techniques for Atraumatic and Precise Electrode Array Insertion in Cochlear Implantation

Nauman Hafeez

Department of Mechanical and Aerospace Engineering
Brunel University London

This document is submitted for the degree of
Doctor of Philosophy

College of Engineering, Design and
Physical Sciences
October 2022

To my parents (late), my wife, Hussain and Sophia

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this document are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Nauman Hafeez
October 2022

Acknowledgements

First and foremost, I would like to extend my greatest gratitude to my supervisors Dr Xinli Du and Dr Nikolaos Boulgouris for their invaluable advice/supervision, patience and continuous support during this hectic process of my PhD study. Their immense knowledge and experience have encouraged me to do well both in academic research and daily life. I would also like to thank our collaborators at Queen Elizabeth Hospital. Their valuable advice around clinical aspects of this work has helped improve this study in different aspects. This work would not have been possible without access to commercial electrode arrays, thanks to Oticon Medical for providing their electrode array for experiments.

Behind every research there is a funding body that is as excited about the work as the people who are carrying it out. This research was no different, I am grateful to our main funding body Royal National Institute for Deaf People (RNID), previously known as Action on Hearing Loss. I am also indebted to Charles Wallace Trust and Wali Mohammed Trust for partial funding support during my PhD.

I have been lucky to have wonderful colleagues during my PhD who were always there for any technical discussion or social mingling. I thank all of them for their precious time. I would like to thank Dr Johan for both research and career advice during my last year of the PhD. We had some resourceful discussions. Last but not least, I am immensely thankful to my wife Jaweria to be incredibly supportive throughout this time while doing her own PhD. Shout out to our children Hussain and Sophia for being so wonderful and a source of happiness.

List of Publications

N. Hafeez, X. Du, N. Boulgouris, P. Begg, R. Irving, C. Coulsan, and G. Tourrels,“Towards unblinding the surgeons: Complex electrical impedance for electrode array insertion guidance in cochlear implantation,” Proceedings of the International Symposium on Auditory and Audiological Research, 2019, vol. 7, pp. 29–36.

N. Hafeez, X. Du, N. Boulgouris, P. Begg, R. Irving, C. Coulson, and G. Tourrel, “Electrical impedance guides electrode array in cochlear implantation using machine learning and robotic feeder,” Hearing Research, vol. 412, p. 108371, Dec. 2021, doi: 10.1016/j.heares.2021.108371.

N. Hafeez, X. Du, N. Boulgouris, P. Begg, R. Irving, C. Coulson, and G. Tourrel, “Real-Time Data-Driven Approach for Prediction and Correction of Electrode Array Trajectory in Cochlear Implantation,” Applied Sciences, vol. 12, no. 13, p. 6343, Jan. 2022, doi: 10.3390/app12136343.

Abstract

Cochlear implant provides hearing perception to people with severe to profound hearing loss. The electrode array inserted during the surgery directly stimulates the hearing nerve according to the frequency bands. The complications during the cochlear implant insertion may cause trauma leading to infection, residual hearing loss and poor speech perception. The motivation for this work is to reduce the trauma induced during electrode array insertion process by carefully designing a sensing method, an actuation system and data driven control strategy to guide electrode array in scala tympani. Due to limited intra-operative feedback during the insertion process, complex bipolar electrical impedance is used as a sensing element. For this, custom impedance meter is implemented along with multiplexers to scan the electrodes to record their bipolar impedance magnitude, phase, and their resistive and reactive components. A 3-DoF actuation system is used for automated insertion and machine learning algorithms are employed for feedback control to steer electrode array atraumatically.

This work is mainly focused on electrode array insertion direction, trajectory, and depth. Insertion trajectory is thought to influence trauma induced, final array placement and insertion failure. Initial experiments were performed to see the changes in complex impedance when certain electrode rub against the scala typani wall. It has been concluded that when the electrode array slide along the wall, the electrodes involved show high impedance magnitude and less negative impedance phase. Further several experiments were performed when array was inserted from three different directions; medial, middle and lateral. Complex impedance data recorded during insertion from different directions has the potential to discriminate different trajectories. Supervised machine learning approach is used to train and test the models for the prediction of different insertion trajectories and insertion depth. This method has shown the efficacy for not only achieving high prediction accuracy on full insertion complex impedance data but also on sub-sequence data. Further, prediction based on shorter data windows are utilized in feedback control loop for real-time control strategy that helps insert the array without any damage to its surroundings. Finally, linear insertion depth estimation of electrode array has been carried out using complex impedance data. A hybrid convolution and recurrent neural network regression model is utilized to predict insertion depth at every millimeter. It has been found that our complex impedance data is consistent and reliable and sensitive to the contact of electrode array with scala tympani walls during insertion.

The contribution of this work is to predict different insertion direction, electrode array path and insertion depth using complex impedance data and machine learning tools. Furthermore, a preliminary autonomous system is demonstrated for real time classification and correction of electrode array path during insertion using closed loop control strategy.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

Cp	Polarization Capacitance
Ra	Access Resistance
Rp	Polarization Resistance
Zc	Contact Impedance
Zf	Fluid Impedance
AF	Adhesive Force
AGM	Image Guided Microsterostatic
AOS	Advance Off Stylet
CI	Cochlear Implant
CT	Computerized Tomography
DoF	Degree of Freedom
EA	Electrode Array
FF	Friction Force
GUI	Graphical User Interface
ISFF	Insertion Speed Force Feedback
NIH	National Institute of Health
PEA	Perimodiolar Electrode Array

PW	Partial Withdrawal
RF	Relaxation Force
RW	Round Window
SIT	Standard Insertion Technique
ST	Scala Tympani

Chapter 1

Introduction

1.1 Motivation

Medical robots have great potential to provide precision based surgical capabilities, rehabilitation and care. Surgical robots are designed keeping in mind workspace or clinical setting of a specific surgical procedure, sterilization of the equipment (easily detachable), accuracy and safety [1]. Surgical robots have exceptional geometric accuracy and they are able to integrate multiple sources of numerical and sensor data, however, they have poor judgment, limited ability to integrate and interpret complex information and they are less adaptive to new situations. Therefore, it is vital for robots to have strong feedback sensing systems to cater for complex situations.

Surgical robots have been employed in knee replacement drilling [2] and minimally invasive laparoscopy [3] in recent times. Research is on going in order to take surgical robotics in to micro-level procedures such as in opthalmic and otology fields. In otology, for instance, there has been studies going on to automate cochlear implant insertion process since 2005. An actuation system with precise feedback control systems is suitable for such procedure that needs to alleviate induced trauma during insertion process. Also, such robots are capable of better position and force control than human threshold and they do not tremor like humans.

A cochlear implant (CI) is an electronic substitute to the normal hearing that bypasses the mechanical mechanism of stimulating the hearing nerves [4]. It instead directly stimulates the hearing nerve through electrical pulses using an electrode array. Among other CI components (mic, sound processor, transmitter and receiver), a thin flimsy electrode array is inserted into the inner ear that has multiple embedded electrodes (according to the manufacturer and model) to carry different frequency sound signals. After the relaxation of CI candidacy criteria in several countries [5], it is fundamentally important to preserve residual hearing during the surgery to avoid variability in speech outcomes post implantation.

It is widely accepted that residual hearing loss is due to the trauma induced during the surgery [6]. The trauma induced may also lead to infection and reduced implant performance. Trauma induced could be due to different reasons. For example, electrode array touching the inner ear walls with sensitive tissues, array tip fold-over and basilar membrane puncture due to array translocation may cause more or less damage according to the performed insertion [7]. Currently, cochlear implantation procedure involves preoperative planning and postoperative evaluation using CT scan images [8]. It is possible to get limited intraoperative information through flouroscopy during insertion [9] but it involves radiation and could be dangerous. There is, therefore, no safe inexpensive intraoperative procedure available to point out trauma or faulty array placement during surgery.

During manual surgical procedure, surgeon can not see inside the inner ear while inserting the electrode array. In order to perform atraumatic insertion, the insertion forces should be limited due to sensitive inner ear anatomy that is beyond the sensitivity of human hand for such small forces in manual surgical procedure. Due to these reasons, different forms of damage is expected. For measuring forces during insertion, several research works have been done with different EA designs, insertion techniques and speed [10][11]. However, the limitation of force sensing method is its inability to measure local insertion forces across different section of EA.

All the cochlear implant systems have built-in impedance measurement mechanism to check the functionality of electrodes after the implant. Impedance measurements and tomography have been employed in various biomedical research areas, for example in needle steering, catheter insertion etc. Although CI systems have built-in impedance capabilities, none of them have clinically used the complex impedance yet. In current CI systems only impedance magnitude is measured and analyzed clinically after the surgery. However, the interface between the CI electrode and its surroundings (perilymph fluid and tissues) is also important both in terms of postoperative speech outcomes as well as during the surgery. Impedance measurements of electrode contacts are influenced by their position in the cochlea [12],[13],[14]. Therefore, complex impedance measurements during insertion can be a good sensing element for a robotic tool for speed, trajectory and position control.

1.2 Aims and Objectives

The aim of this thesis is therefore to develop an automated system that could help to reduce trauma during electrode array insertion. An automated system compromises of an actuation unit, a sensing modality and closed loop control strategy to steer the electrode without damaging the inner ear and for optimal placement. The objectives of the work are as follows:

1. A sensing system that is able to give useful information regarding behaviour of different parts of electrode array during the insertion. The information should be useful enough to discriminate between different electrode array trajectories, position and depth.

2. Sophisticated signal processing algorithms utilisation for analyses of sensor data that is helpful in electrode array atraumatic steering.

3. An actuation system with displacement and velocity control with reasonable degree of freedom for electrode array steering into the scala tympani model.

4. A feedback control system that is able to use sensing measurements and make corrections to the electrode array insertion where and when needed.

5. Integration of all components of a robotic system in the form of an easy to use interface. The overall system should be able to work on real time insertion process.

1.3 Contribution

The research has been on going to find avenues for replacement of manual surgical procedure of cochlear implantation with automated and atraumatic steering of electrode array into scala tympani canal. Some studies have introduced integrated sensors into the electrode array or on its tip for atraumatic steering [15], [16], [17]. These integrated sensors come with the problem of introducing stiffness to the array which negate the purpose. Other studies have used external force sensors to measure insertion forces; either under the cochlear model or on the insertion tool or both. However, they can only measure overall insertion forces and unable to give discrete electrode trajectory or position information. An alternate solution is to use electrode impedance measurements that does not need integrated sensor and can also provide information about discrete parts of electrode array during insertion. Contributions of this thesis are outlined below:

1. An impedance meter is developed using digital acquisition devices that can measure bipolar impedance magnitude, phase and resistive and reactive parts of electrodes. A switching circuit is integrated with it to scan through multiple electrode pairs for impedance recording.

2. A method for electrode array localization is developed in terms of its trajectory and position prediction. Supervised machine learning algorithms are employed to make predictions using complex impedance recordings of electrodes.

3. Real time prediction and correction of electrode array path is demonstrated using an integrated robotic system with actuation, sensing and control unit.

4. A linear insertion depth estimation model is developed that may help surgeons with intra-operative and post-operative electrode array depth during surgery.

1.4 Thesis Outline

This thesis has seven chapters including the introductory chapter. After the first chapter, the thesis is organized as follows:

Chapter 2 presents current state of the art in the domain of robotic cochlear implantation. It starts off with explanation of auditory system and cochlear implant system. It also explains different types of trauma induced during the insertion process. Further, different robotic systems developed so far so CI application; briefly the drilling robots and extensively electrode array insertion actuators. In the end, the sensing elements that could be used with such robotic system are presented. Finally, machine learning and deep learning algorithms are presented with focus on classification and regression problems.

Chapter 3 introduces the methodology and experimental setup that has been used for this work. It describes different parts of the experimental setup in detail along with how impedance measurements of electrodes are carried out. Preliminary results of such experiments are also presented at the end.

Chapter 4 evaluates machine learning algorithms for electrode array trajectory prediction using full insertion complex impedance data. Moreover, machine learning pipeline is explained and results are presented using different combination of measurement variables and number of electrode pairs used in the prediction.

Chapter 5 further evaluates machine learning algorithm for trajectory and path predictions on raw complex impedance data for electrode array localization. Further to that, main focus of this chapter is to present control strategy for real time trajectory and path prediction and correction. Classifier in a loop system is introduced as a closed loop feedback system for steering of electrode array.

Chapter 6 describes an advance model developed for electrode array linear insertion depth estimation. First, it evaluates machine learning models for final insertion depths.

Furthermore, a hybrid regression model of 1D convolution and recurrent neural network is presented that can estimate linear depth at every instance of 1mm.

Chapter 7 concludes the thesis with its potential for atraumatic electrode array localization, limitations of the research and finally the future directions to extend this research.

Chapter 2

Literature Review

2.1 Cochlear Implantation

It has been four decades, cochlear implant (CI), an electronic device, is providing restoration of auditory perception in patients with sensorineural hearing loss [18]. The CI mechanism directly stimulate the nerve system by electric means bypassing the damaged sensory hair cells which are responsible for transconduction of acoustic signals into electrical signals in normal human ear.

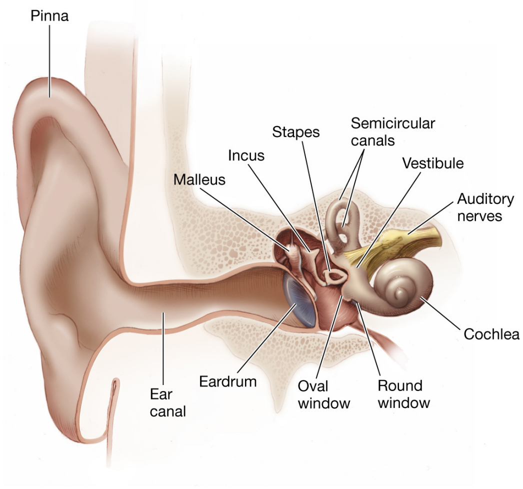


Fig. 2.1 Human ear anatomy [NIH open source image]

The outer part of normal human ear anatomy (Figure 2.1) receives sound waves through pinna, travels through the ear canal and vibrates the tympanic membrane (ear drum) due to their pressure. There are three bony structures (malleus, incus and stapes) in the middle ear which forms osscîles and it delivers mechanical energy to the oval window received from tympanic membrane. Oval window is one of the two openings of cochlea, the other is round window. Cochlea is a spiral shaped structure in the inner ear which is filled with fluid. Mechanical energy from oval window transfers to the fluid in the cochlea and produces waves in the fluid. Inner hair cells bend due to these waves and change this movement into electrical pulses for auditory nerve. Electrical impulses (information) are then sent to the brain through auditory nerve, where they are interpreted as sound [19].

There are three fundamental types of hearing loss. (1) Conductive, which is due to abnormalities in external ear (ear canal/drum) or ossciles in the middle ear. This type of hearing loss can be treated with hearing aids. (2) Sensorineural, which occurs when the inner ear (cochlea) is damaged. This malfunction is the result of corrupt hair cells or auditory nerve which in turn unable to send electrical pulses to the brain. Sensorineural hearing loss is always almost permanent and can be treated with cochlear implant. (3) Mixed hearing loss is the combination of conductive and sensorineural loss. Hybrid electro-acoustic cochlear implants are used to treat such hearing loss.

Cochlear implant system has mainly two major parts; one is worn outside the ear consists of a microphone, speech processor and, a transmitter and one part is inserted inside the ear anatomy, by drilling a hole, consists of a receiver and, a thin flimsy electrode array (EA) (Figure 2.2). Speech signals are picked by the microphone, converted into digital codes by the processor and transmitted to the transplanted part of the device. The inner part is responsible to convert the digital codes in to electrical pulses which eventually stimulate different portion of the nerve system via electrode array, placed into scala tympani (ST) of cochlea.

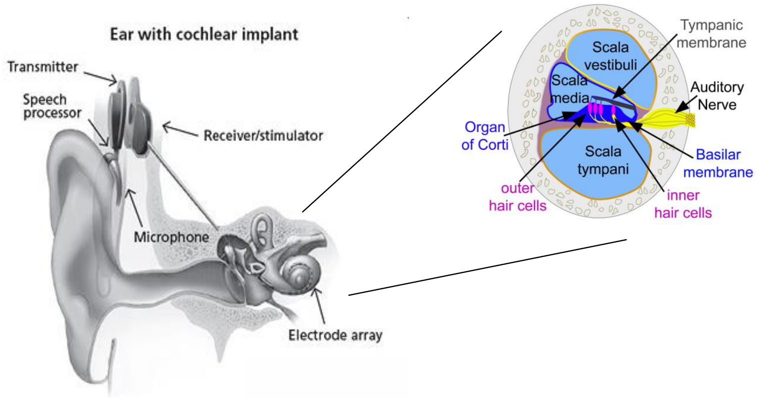


Fig. 2.2 Cochlear implant system and anatomy of cochlea[NIH open source image][20]

Electrode array is inserted in to ST through round window membrane by drilling the mastoid bone. Insertion process of EA can result in complications. A patient can lose residual hearing if electrode array insertion results in damaging the delicate structure of the inner ear [21]. This includes injury to the lateral or modiolar wall and distortion of basilar membrane. There are three objectives of EA insertion for better hearing outcomes: (1) deep insertion in to cochlear to cover lower frequency range, (2) close proximity to the modiolus wall to ensure greater operating efficiency and (3) to preserve residual hearing by preserving inner ear structure [22]. In order to accomplish first two goals, there is always a high chance of insertion trauma.

2.1.1 Electrode Array Designs

Electrode arrays are designed with specific properties to keep intra-cochlear trauma to the minimum during implantation [23]. The EA design has become more important as the benefit of electro-acoustic stimulation has been established [24]. Three types of electrode arrays are commercially available: (1) Straight EAs are positioned close to the lateral wall of cochlea; (2) perimodiolar EA hugs modiolar wall of the cochlea; (3) mid-scala EA which resides in the middle of the ST, however, often considered as perimodiolar type. Electrode arrays come in different lengths to accommodate diverse cochlea structures and sizes. The overall length (base to tip) of most commonly available EAs ranges from 16mm-31.5mm, however, active stimulation length containing all stimulation channels is different than overall length. The number of stimulation channels ranges from 12-24 in different EA designs. Complete insertion of active stimulation length of array is usually not possible due to inner ear anatomical variations and the angle at which round window opening lies [23]. Figure 2.3 is showing electrode arrays segregated according to their overall lengths and angular insertion depths. Deeper insertion due to lengthy EA may be advantageous in terms of processing wider frequency band but it increases chances of damage to the cochlea structure during insertion.

EA design also determines its efficiency which is considered to be at peak when it is placed close to the neural activation site. Modiolus hugging EA operate at lower current thresholds than the straight lateral wall EA and believed to be performing better [25].

Atraumatic EA insertion is an important factor in the design process. It is almost impossible to avoid insertion trauma with any EA design currently available. Pre-curved perimodiolar design is mostly likely to cause intra-cochlear injury because of the stylet wire which is responsible for bringing the EA in the curved shape [22]. There is possibility of penetration in spiral ligament at the end basal turn due to stiff tip. Another factor of trauma is stiffness due to bundling of wires and basal stiffener in EA. It is also impossible to abort pre-curved EA insertion in case of buckling, however, surgeon can abort insertion in case of straight design.

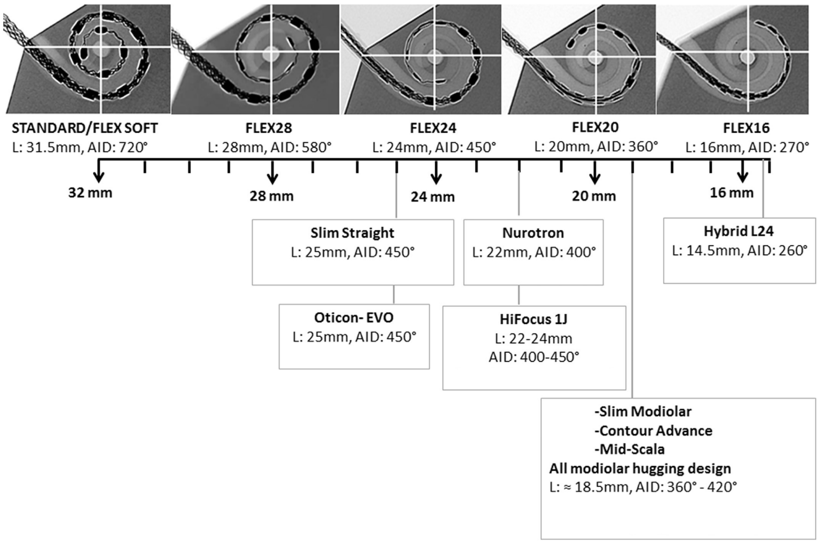


Fig. 2.3 Images of MED-EL electrode arrays and information of other commercially available electrode arrays. Figure from Dhanasingh[23]

2.1.2 Surgical Techniques

There are two aspects to the surgical techniques; firstly whether electrode array is inserted through round window (with or without drilling its edges) or neighboring cochleostomy, and secondly the insertion technique utilized. Trauma prevention, ease in insertion and better visualization are epitome of selection of surgical techniques. However, human ear anatomy also plays a factor in selection, for instance, in case of position of the insertion (round window or cochleostomy) [26].

There are three insertion techniques most frequently employed for EA insertion; Standard Insertion Technique (SIT), partial withdrawal and Advance Off-Stylet (AOS). Standard insertion technique is used to insert straight lateral wall EA with bare surgeon hands. Partial withdrawal and advance off-stylet techniques are both utilized to insert perimodiolar EA. The difference is the way stylet is removed during insertion process. In partial withdrawal, stylet is removed at the end of the insertion process to resume pre-curved state to hug modiolar wall. In advance off-stylet technique, stylet is withdrawn in a continuous manner after EA reaches close to lateral wall. The advantage of this technique is to avoid array lateral wall contact and ultimately reducing the trauma induced [27]. Although this technique is advantageous than other techniques, however, even the experienced surgeosn are unable to avoid translocation due to stiff tip of perimodiolar electrode array [28].

2.1.3 Insertion Trauma

Damage to the inner ear during insertion process is a concerning factor as this dictates long-term survival of the neurons to be responsive [29]. Another factor is to preserve residual hearing which may be lost due to insertion trauma. It is primarily factual for patients who are recipient of electro-acoustic cochlear implant where preservation of remaining auditory function is of utmost importance. Preservation of residual hearing also improves the post surgery hearing outcomes in other patients. Trauma induced during electrode insertion can be classified as a grade. Grade 0 represents no injury to the internal structure of cochlea during the insertion process (Figure 2.4 (A)). This means either electrode array does not touch the inner walls or touches but with very little force that there is no damage. Roland [21] performed cochlear implant experiments and observed damage to the different parts of the intra-cochlea structure. Damage to the lateral wall or basilar membrane is consider as grade 1 trauma. In Figure 2.4 (B), there is damage to the lateral wall around the brackets although basilar membrane is not punctured. This type of injury occurs due to intrusion of the array tip or array body pressure imparted on the area where basilar membrane and spiral ligament conjoin. Lateral wall damage does not only happen in case of straight array insertion, it is also observed in perimodiolar type arrays if the stylet is not removed in time (Figure 2.4 (G)). When basilar membrane is involved in such injury, it may just be lifted or array penetrates in to scala media. Figure 2.4 (C) demonstrates rapturing of basilar membrane which takes it away from its spiral ligament attachment. In such an injury, basilar membrane is torn but electrode array does not enter scala vestibuli. Grade 2 trauma is due to the penetration of spiral ligament by the electrode array tip. In this kind of penetration, cochlear duct remains in its position, however, it is up-lifted moved by the array tip with damage to the scala media (Figure 2.4, (D)). Grade 3 trauma is due to the displacement of electrode array from ST to scala vestibul which results in mixing of endolymph and perilymph fluid. The highest degree of trauma is grade 4 in which electrode array fractures the osseous lamina (Figure 2.4 (E,F)). This type of trauma is severe because there is possibility of degeneration of ganglion cell in the damaged area [30] . An image guided robotic drilling system was designed by Majdani [31] in which cameras are placed at the center point of the robot, at the base of the robot and at the temporal bone holder (Figure 2.6 (A)). The robot used for this purpose is a small industrial robot with 6-DoF and loading capacity of 3 Kg. The Robot has remote sensor interface to connect with remote-control unit which operates on build in control program. This system was tried on 10 cadaver specimens and target position deviation was 0.78±0.28 mm.There was no violation of facial nerve, however, there was chorda tympani nerve violation in one of the experiments [31].

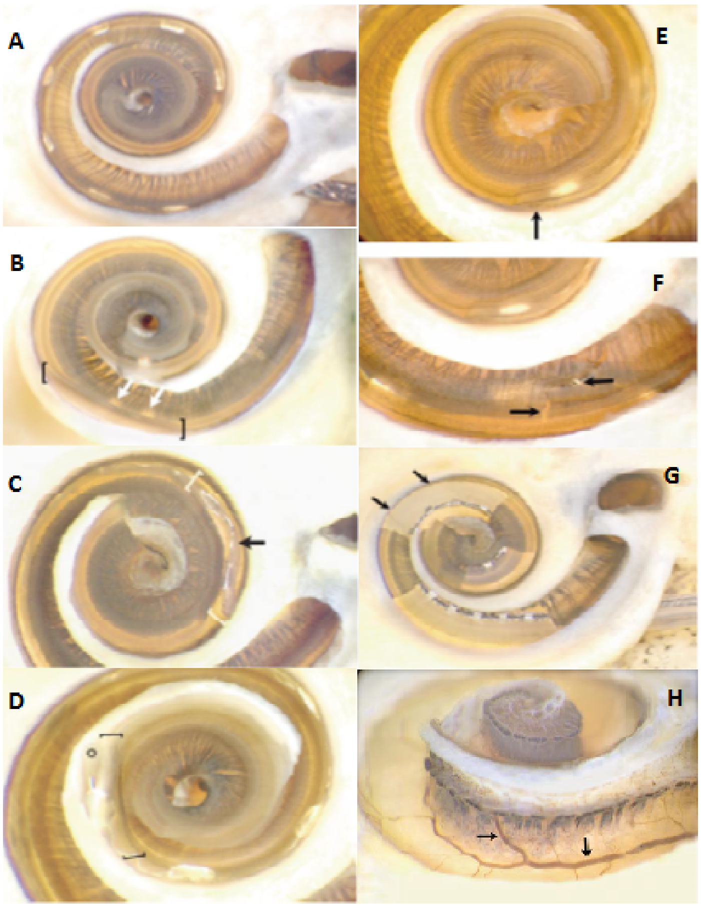


Fig. 2.4 Insertional Trauma:(A) Normal insertion without damage to lateral wall. (B, C, D) Lateral wall trauma. (E,F) Osseous lamina fracture. (G) Lateral wall and basilar membrane trauma due to perimodiolar electrode array insertion. (H) Posterior spiral vein susceptible to electrode-induced trauma. Figures from Roland [21]

2.1.4 Insertion Failure

Several types of complications can arise during cochlear implant insertion which may or may not result in trauma. Figure 2.5 (left) represents tip scrapping during insertion process and it is already discussed in section 2.1.3. Figure 2.5 (center) dictates the tip fold-over failure which is rare occurrence during insertion process of straight array. However, it mostly happens in case of perimodiolar array especially when stylet is removed earlier than it should be. In this type of failure, the tip of the array folds over itself during the insertion process which results in poor placement of the electrode array. When array folds over, some of the electrodes would poorly stimulate the auditory nerve. This failure also doubles cross-sectional area of array which essentially increases the risk of trauma induced as it slides in to ST [14]. Figure 2.5 (right) demonstrates the buckling of the EA during insertion. This failure occurs when electrode tip is experiencing resistance during insertion and further effort of insertion would result in either internal or external buckling. It is assumed that stiffer rear of EA avoids buckling failure but its downside is more pressure exerted on tip during insertion.

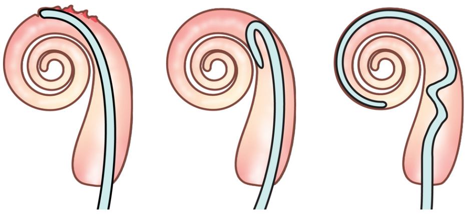


Fig. 2.5 Three potential sources of intracochlear insertion trauma. Left (Tip Scraping), middle (Tip Fold-over), and right (Buckling). Figure from Clark [15]

2.2 Robotic Cochlear Implantation

In order to avoid trauma during insertion process, several techniques have been employed such as developing new types of soft and flexible EAs, novel insertion tools and techniques, measuring and minimizing the insertion forces, measuring electrical properties of EA to predict and approximate the position of the EA during insertion, magnetic and image guided insertion, and employing robots for automated mastoidectomy drilling and EA insertion.

Drilling robots will be discussed first in this section followed by the robots used for EA insertion.

2.2.1 Robotic Drilling Systems

For cochlear implantation, surgeons have to use drill for an opening to the cochlea either through round window or cochleostomy. The objective of drilling procedure is to access cochea for EA insertion avoiding proximity to the facial recess and chorda tympani nerve. This is the most time consuming and invasive procedure in the CI surgery and it may also result in inducing trauma. A better opening to the cochlea may also help in less trauma induced during the EA insertion process [31, 32].

Figure 2.6 is displaying selected robotic drilling systems designed by different researchers around the world. However, the robotic system designed by Bell [33] is the only one deployed for an actual surgery in 2017 (Figure 2.6 (C)). The rest of them are in testing phases and employed in cadaver drilling tests.

The robot built by Bell [33] is for stereotactic CI with high accuracy levels and suitable for complex clinic environment as compared to other robots. As it can be seen in Figure

(2.6(C)), this robotic system consists of five DoF serial arm with reasonable weight and dimensions. Its design allows for easy mounting on the operating room table using regular flexible arms. A typical surgical drill is connected to its effector as well as a built-in force sensor for an intuitive haptic interaction for the surgeon. Optical tracking system is used for tool position verification. The robot is connected to a touch screen interface that is used to control the robot and surgeon's guidance through the whole process. After testing on 15 temporal bones, its accuracy (with confidence interval) turned out to be 0.51±0.41 mm with an angular misalignment of 0.88±0.40°.

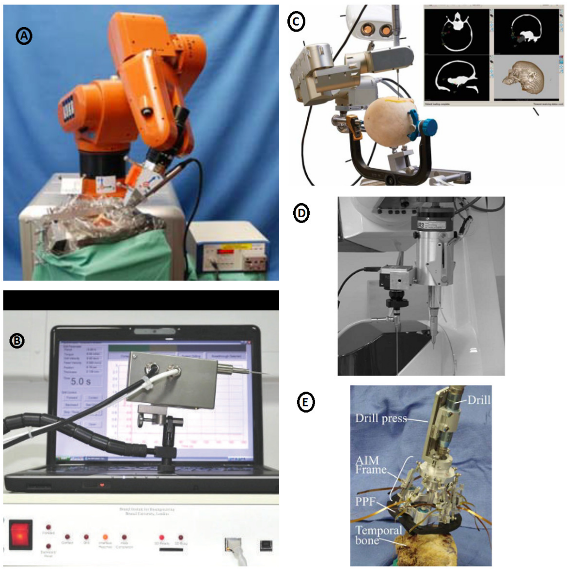


Fig. 2.6 Diverse Robotic drilling systems for cochlear implantation. Figure from (A) Majdani [31] , (B) Bret [34] , (C) Bell [33], (D) Klenzner [35], (E) Kratchman [32]

A force controlled guided robot presented by Klenzner [35] is aimed at demonstrating collision free drilling path from facial nerve to the planned mark of cochleostomy (Figure 2.6 (D)) . Screw markers were placed for registration on temporal bone and a 3D trajectory is planned by using computer tomography (CT) data. In order to visualize the trajectory, a 1.9 mm endoscope is placed on the robotic system. Experiments were performed four times with registration errors (deviation from planned mark) from 0.17-0.37 mm and a mean of 0.25 mm which is less than tolerance level of 0.5mm.

An arm supported (Figure 2.6(B)) and a hand guided robotic drills are designed by Brett [34]. The main feature of both types of drilling system is intelligent interaction with tissue structures by using smart sensor embedded in to drill bit and adjust motion strategy accordingly. The advantage of a hand guided robot is the flexibility of adjusting trajectory. In hand guided robot, sensing system also distinguishes between surgeon's intended and involuntary motion during the procedure.The system compromises of drill unit, a hard wired control unit for interpreting sensory signals and a PC to visualize signals. However, arm supported system (drilling unit) is mounted on flex-lock arm which gives flexibility of free movement and stabilization. The force level during hand guided robotic drilling of porcine cochlea has a range of of 1.4-2.86N with a mean value of 1.99N.

Figure 2.6 (E) shows a re-positionable automated image-guided microsterotactic (AIM) frame designed by Kratchman [32]. It is designed using standard Gough-Steward parallel robot. Similar to Klenzner approach, trajectories were calculated using CT scans data of patients. The actuation part of the part is six linear piezo-electric motors which can generate 2N maximum output force. It also has position sensor with minimum resolution of 0.5 µm. A custom control program written in MATLAB provides approximation of a target pose using trajectory data. Experiments performed on phantom cochlea model present an accuracy of 0.20±0.07 mm and an accuracy of 0.38 mm in a cadaver experiment.

2.2.2 Robotic Electrode Array Insertion

The main concern during CI surgery is the chances of trauma induced during electrode array insertion procedure. This is attributed to the fact that surgeon performs insertion by hand (potential tremor) without any visualization of inner ear and limited sensing feedback. A robotic insertion system with precise speed, trajectory and position control can be a solution.

An automated insertion tool for EA insertion was the emphasis of early research in this domain. A very basic automated tool was designed by Roland [36] shown in Figure 2.7 to evaluate intracochlear insertion dynamics, forces involved, trauma induced and electrode position using a contour array. The automated insertion tool has Instron® force sensor to measure basal turn intracochlear forces. The accompanying Instron® software application measures force and displacement data at regular intervals during the insertion of 2 mm/s rate. Automated insertion experiments were performed on cochlear models as well cadavers using contour EA (using standard insertion technique) and contour advance EA (using advance off-stylet technique). Fluoroscopic images were taken during the insertion process to study array dynamics in the cochlea turns. The mean insertion forces were far less when using advance off-stylet technique compared to standard insertion technique.

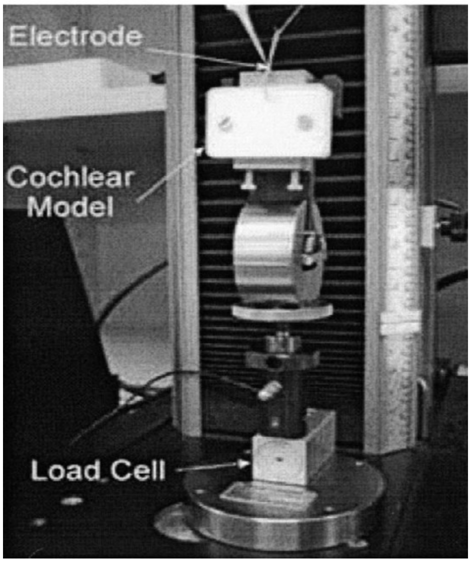


Fig. 2.7 Mechanial system for electrode array insertion in cochlear phantom model. Load cell is placed to measure insertion forces. Figure from Roland [36]

Going forward, a 2 DoF experimental robotic insertion system with force monitoring was developed by Zhang [37] to study ability of a robotic system using active-bending EA (Figure 2.8). The setup is comprised of a linear actuator, amplifier, force sensor, electrode actuator and 2-D cochlea phantom model. Mathematical model, kinematics and path planning for steerable EAs are developed to reduce intra-cochlear forces and resulting insertion trauma. Insertion forces were evaluated in case of steerable and non-steerable electrode arrays where the former type was supported by a mathematical model and path planning algorithm. Insertion forces were 70% less in case of steerable electrodes with planned path and control as compared to non-steerable electrodes.

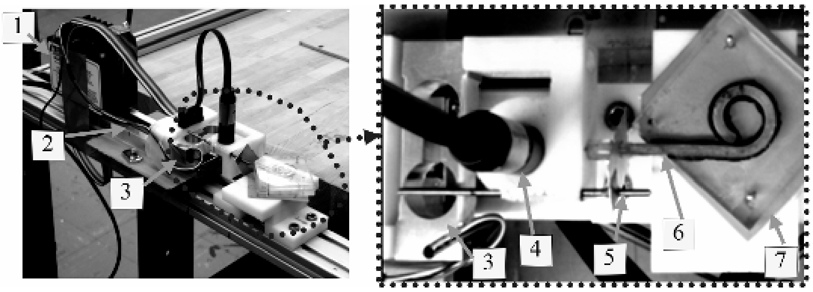


Fig. 2.8 Experimental setup of automated electrode array insertion. (1) amplifier (2) linear actuator (3) force sensor (4) electrode actuator (5) implant support (6) electrode array (7) 2D cochlea model. Figure from Zhang [37]

Zhang further his research of robotic insertion in [38] with the design of 4-DoF tool and insertion speed force feedback (ISFF) system (Figure 2.9). As can be seen in Figure 2.9, there is freedom of 4 way movement (q1,q2,q3,q4). Insertion forces, steering of tip and approach angle were evaluated using different DoF systems. It was found out that 4-DoF system has the least insertion forces as compared to 1-DoF and 2-DoF systems due to added freedom of adjusting insertion angle in 4-DoF tool. Insertion forces decreased when insertion speed force feedback mechanism was used with steerable EA whereas an increase was observed with non-steerable EA. It is also observed that steerable electrodes help in preventing array buckling.

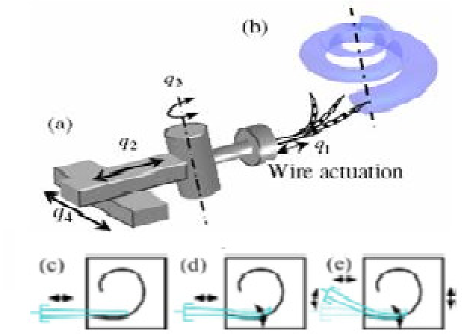


Fig. 2.9 (a)Robotic system model. (b) cochlea model (c) 1DoF insertion. (d) 2DoF insertion. (e) 4 DoF. Figure from Zhang [38]

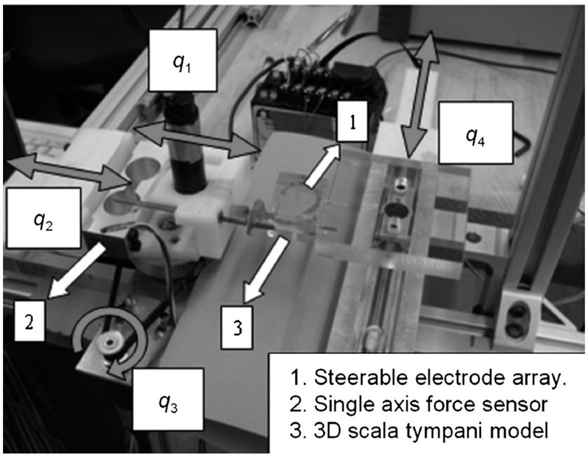

q_1
1
q_4
q_2
2
3
q_3
1. Steerable electrode array.
2. Single axis force sensor
3. 3D scala tympani model

Fig. 2.10 Experimental test-bench with 3D ST phantom model. Figure from Zhang [38]

Based on the results obtained in [37][38], a second generation robot has been designed and constructed (Figure 2.10). A small and compact parallel robot is designed using 6 actuators that manipulate its moving platform through 6 independently controlled linear actuators (cylinders). The design is similar to Stewart-Gough platform. The base of the robot is held stationary and steerable electrode is placed on the moving platform. An additional motor is used to control the pull of the strand when using steerable EA. A force sensor is mounted on the moving platform which is capable of detecting spatial forces and moments in any direction. The robot is able to move at a maximal speed of 5.7 mm/s in space. A Phontom Omni haptic device is used to control the robot. This joystick allows the surgeons to move the robot, following the hand motions. The computer shows and records the surgery information, such as insertion depth, insertion speed and sensed force.

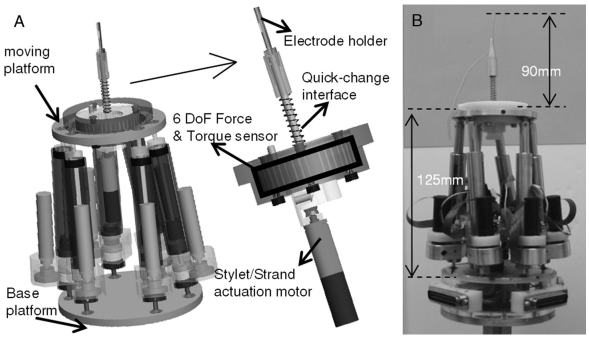


Fig. 2.11 Advance robot design with 6 linear actuators. Figure from Zhang [39]

An automated insertion tool is also presented by Rau [10] as shown in Figure 2.12. Perimodiolar EA were inserted in the cochlea model using AOS technique. The robotic system has two gearless motors with top velocity of 5 mm/s, position certainty of 1 μm and displacement range of 45 mm. One of the drivers is connected to the gripper to grab the electrode carrier. Other driver is connected to the stylet which is used to keep it in a desired position. A protective layer is used to cover the EA and gripping mechanism.The upper jaw of the tool has Instron 5542 force sensor (10N load cell) connected with cochlea plastic model to measure insertion forces. Insertion forces were measured with different insertion feed rate and insertion depths were observed. These results can help better electrode array designs and insertion techniques for implantation.

Hussong [40] proposed a custom automated insertion tool for CI (Figure 2.13a) which is optimized for better insertion depth. It consists of a plier-like grasping mechanism for the electrode and a wired hook to retract the stylet. The pliers and the wired hook are connected to the gearless linear nano-positioners with integrated position sensors. The actuators are based on piezo technology, offering a resolution of 5nm and positioning accuracy of 1μm. The actuators travel range is 45 mm and the maximum velocity 5 mm/s and provide translational 1-DoF movement. The actuators are sequentially placed upon each other with a plastic plate between them to which the pliers are connected. A u-shaped tube, covering the electrode as well as the grasping mechanism, acts as a protection and guidance with the drill canal. The actuators are controlled using a custom made software written in C++ on a standard PC. The software provides a GUI and allows setting actuator trajectories independently. An external Modular Control System is provided by the actuator's manufacturer, includes a position control device and realizes the set trajectories.

Insertion of Nucleus Contour EA were carried out in cochlea phantom model and human cochlea specimens using the testbed in Figure 2.13b. It was found that a mean angular insertion depth of 410° was achieved in lubricated cochlea model and 330° in cadaver.

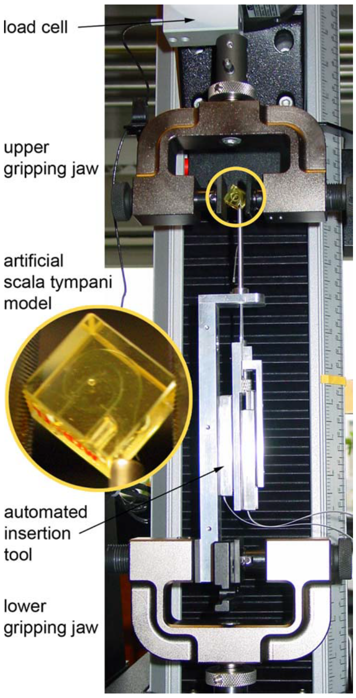
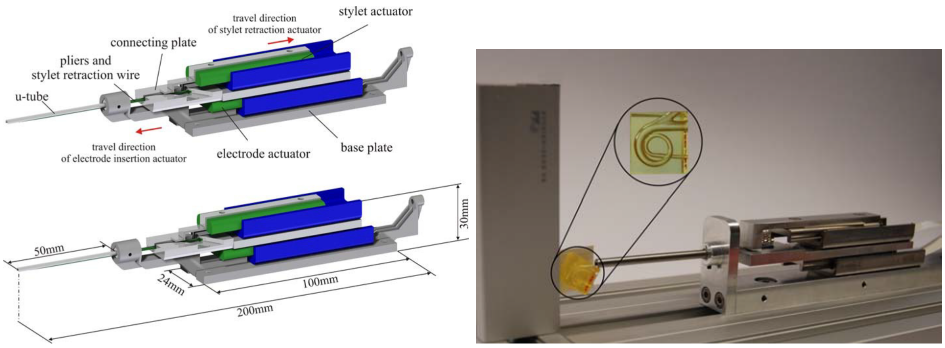


Fig. 2.12 Automated insertion tool setup with load cell to measure forces. Motion of the electrode array in to ST model and style withdrawal are carried out by lower gripping jaw. Figure from Rau [10]

Schurzig [41] took the design of Hussong [40] to the next level by incorporating a force sensing mechanism with automated insertion tool (Figure 2.14). Force sensing mechanism is housed alongside the insertion tool and is based on flexible thin-film elements. The force profiles of this experimentation will be discussed in 2.3.1.



(a)

(b)

Fig. 2.13 (a) Robotic electrode array insertion tool with description of the components. (b) System setup for insertion. Figure from Hussong [40]

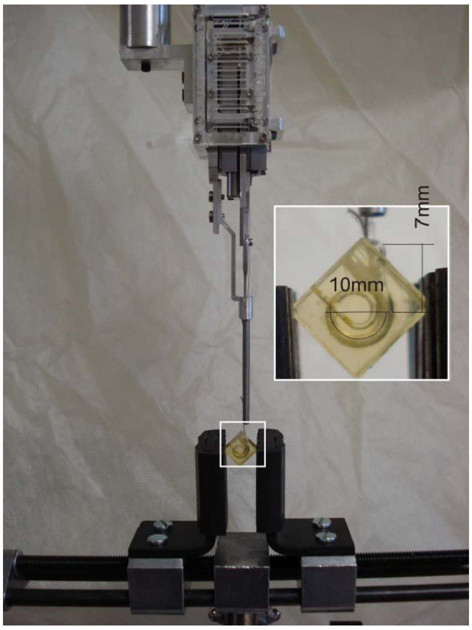


Fig. 2.14 Automated insertion tool setup with integrated force sensing unit. Figure from [27]

Most of the robots discussed are in vertical placement, however, Mirrior [11] has designed an automated insertion tool which is placed horizontally (Figure 2.15). Again, the purpose of this automated tool is to measure insertion forces. The test bench includes a electro-mechanical inserter with 1-axis force sensor and an insertion tube to cover the EA. This sensor is placed between inserter and blunt pin whereas 6-axis force sensor is placed below cochlea model. The speed of the actuator used for insertion is controlled by an input voltage. Forces exerted by the inserter were measured by 1-axis force sensor whereas the friction forces between cochlea and EA were recorded by 6-axis sensor. Advanced Bionics 1J lateral wall electrode arrays are used in the experiments. The effect of tube material (plastic/metal), the insertion speed and lubrications on friction and insertion forces were observed. The results of these forces will be discussed in 2.3.1.

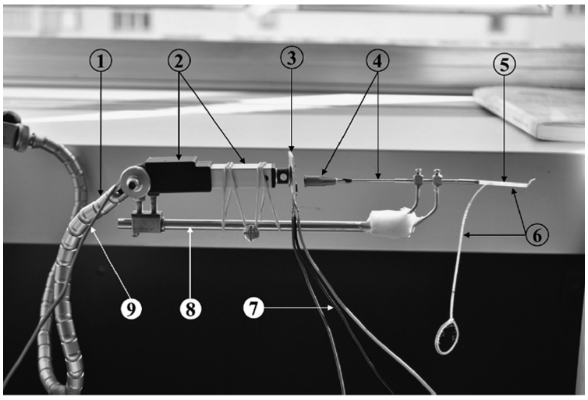


Fig. 2.15 Automated insertion tool: (1) Flexible arm (2) Linear actuator. (3) 1-axis force sensor. (4) Blunt pin. (5) Insertion tube. (6) Electrode array. (7) sensor cable. (8) DC power unit. (9) Digital interface card cable. Figure from Miroir [11]

Pile [42] developed 3-DoF parallel robot with fourth actuator for stylet with wire actuated prismatic legs having new approach of coordinated insertion of perimodiolar array in order to minimize shape discrepancy (between shape EA and inner ear). In order to separate DC motor actuators from the end effectors and better control over weight distribution, wire actuation is applied (Figure 2.16. A 6-axis force/torque sensor is used to hold the end effector compromising of actuator and a gripper. First step in designing the robot is to obtain an accurate specification of allowable orientation workspace for CI surgery. This robot is designed with task specifications as follows: targeting accuracy <0.4 mm, tool maximum diameter 2.0 mm, range of insertion speed 0.5-5 mm/s, angle of approach of 6°-17° and insertion force of nearly 0.05 N. An optical tracker is used to collect position and orientation information of EA gripper in an experimental setup. Digital microscope was used to capture images during the insertion process in transparent phantom model provided by Cochlear Ltd. Insertion forces results from this robotic insertion experiments will be discussed in 2.3.1.

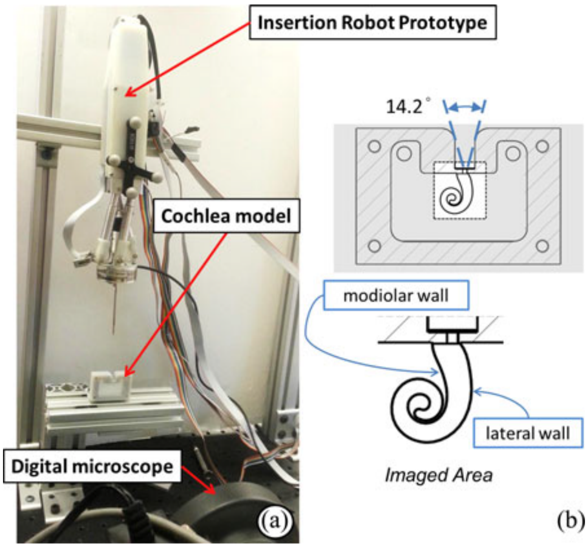


Fig. 2.16 (a) Experimental setup of the insertion robot with digital microscope recording the insertion process. (b) ST plastic model covered in a case. Figure from Pile [42]

An image guided automated insertion system was introduced by Bruns [43] in 2017 (Figure 2.17). This imaging system is capable of accurately positioning the insertion tool at the entry point of the cochlea and aligned with favorable entry vector. For the localization of automated tool, an optical tracker is utilized and a graphical user interface (GUI) system with specific patient's CT (planning) scan data guides the surgeon for any possible error in real-time. It has been observed that 0.31 mm is the average alignment accuracy for even layman users.

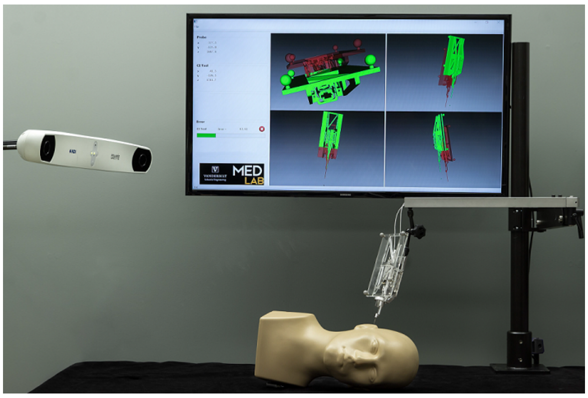


Fig. 2.17 Image guided automatic insertion tool for cochlear implantation. Figure from Bruns [43]

2.2.3 Integrated Steering Mechanism

All of the automated systems presented thus far are using commercially available electrode arrays to find solution for atraumatic insertion. In this section, the systems with integrated steering mechanism will be discussed.

A magnetic guided insertion system was presented by Clark [15] in which magnetically tipped EA is guided as it is inserted in to the cochlear with the help of a manipulator magnet placed near the patient's head which apply magnetic torque to the array tip causing it to bend away from the ST walls (Figure 2.19).

To achieve EA guidance, a small permanent magnet is placed at the tip of a scaled array (83 mm) and a large manipulator magnet is placed near patient's head to apply a magnetic field to the array tip which is controlled by a motor. The motor is allowed to translate along its rotation axis, varying the distance between the manipulator and the patient, effectively changing the strength of the applied magnetic field on EA tip and hence manipulating bending movement of the EA.

Two control algorithms have been deployed to synchronize the motions of the manipulator magnet and insertion of EA. 1) Maximum field method (only rotation needed) 2) Maximum Torque method (variable distance needed). Experiments were performed with non guided and magnetically guided arrays. Non guided EA made its first contact with ST wall at 17.5 mm while max-torque and max-field at 39 mm and 37.5 mm respectively.

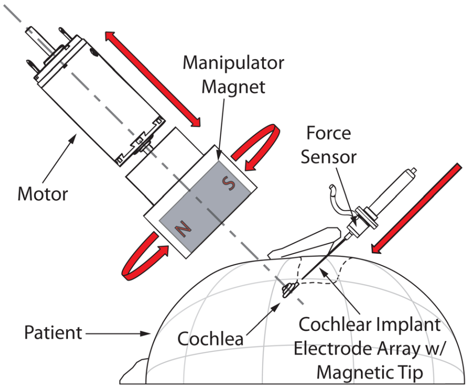


Fig. 2.18 Concept of magnetic guidance of electrode array in to ST with 3DoF. Figure from Clark [15]

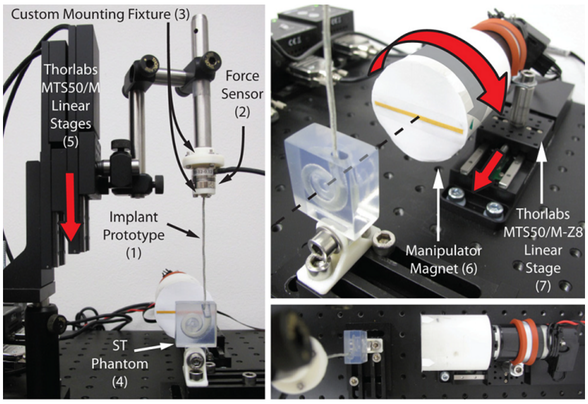

Custom Mounting Fixture (3)
Thorlabs MTS50/M Linear Stages (5)
Force Sensor (2)
Implant Prototype (1)
ST Phantom (4)
Manipulator Magnet (6)
Thorlabs MTS50/M-Z8 Linear Stage (7)

Fig. 2.19 Test bench of electrode array insertion through magnetic guidance.(1) prototype. (2) Force sensor. (3). Custom mounting fixture (4) ST model. (5) Vertical linear actuator. (6) Manipulator magnet. (7) Horizontal linear actuator. Figure from Clark [15]

The solution provided by Wise [16] is based on MicroElectroMechanical (MEM) technology. High density silicon based electrode arrays are developed with integrated mechanism for stimulation, recording, and position control (Figure 2.20). The system consists of 32 electrode sites which are handled by 4 channels for stimulation, 8 strain gauges for real-time shape sensing and a tip sensor to avoid possible contact with intra-cochlear structure by measuring the tip forces. For the position localization, the array can be pre-stressed to hug the modiolus.

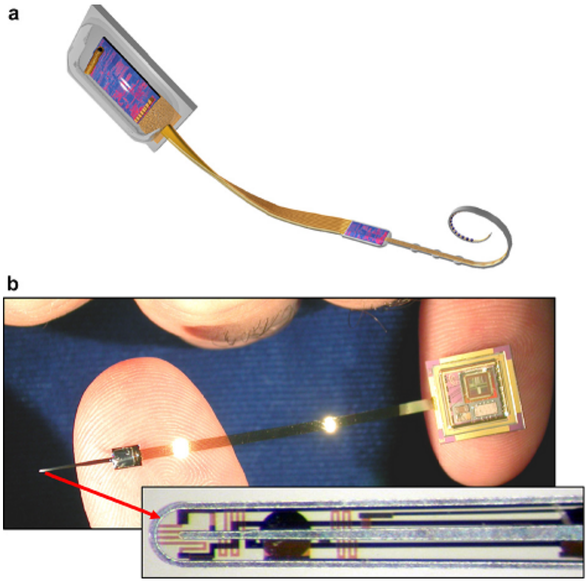


Fig. 2.20 (a) Cochlear microsystem with microprocessor, data converter and wireless interface connected to an electrode array. (b) Electrode array with wall contact and position sensing strain gauges. Figure from Wise [16]

2.3 Sensing Methods

The aim of applying sensing techniques during the insertion process is to relate the sensing measurements with induced trauma, position of the array with respect to intra-cochlear structure, and insertion depth and angle.

2.3.1 Insertion Forces

Force sensing is an important research topic for an atraumatic EA insertion. The main aim of an atraumatic insertion is to reduce the insertion forces due to sensitive inner ear anatomy. A human hand is unable to detect such small forces (in mN range) and ultimately different forms of damage are expected. For these reasons, automated insertion tools discussed in 2.2.2 are investigated with insertion speed and angle control, and force sensing capabilities. Researchers have used different types of electrode arrays, insertion techniques, ST friction, and insertion speed to evaluate the impact of insertion forces.

In 2004, Adunka [44] compared a new prototype electrode array (Flex AES) with C40+ array type in terms of the trauma induced, their intra-cochlear proximity and the insertion forces. However, there was no correlation drawn between insertion forces and the extent of trauma induced. Microscope and radiography techniques were used in the cadaver insertion experiments to observe intra-cochlear trauma. Forces were measured during the 23 mm insertion (only one cochlear turn) experiments in ST phantom model. Insertion forces were reduced by more than 40% when the new prototype Flex AES was used compared to C40+ array. This is due to relatively softer tip, high flexibility and small size of the new array. There was no substantial damage to the inner cochlea structures while using new prototype. In a similar study, Nguyen et al. [45] compared two EA designs with different proximal diameter (0.94 mm vs 0.5 mm) and distal tip diameter (0.5 mm vs 0.4 mm). An automated insertion tool developed by Miroir was used for perform 23 insertions in human cochlea specimens and insertion forces were measured. According to the results, lower insertion forces were recorded beyond 270°depth in case of thin array and it was also inserted deeper than the other EA.

According to Zhang [37], robotically inserted novel steerable electrode array has comparatively less insertion forces (70%) than straight array. Steerable EA was also favorable in preventing buckling during insertion. However, tip folding or over insertion can occur in such EA insertion. In order to reduce the insertion forces, kinematics and path planning for steerable EA are developed. Two different EAs are designed to yield different bending shapes that best approximates the shape of the cochlea. Mathematical modeling for direct kinematics and path planning of EA have been done which is first simulated on MATLAB and then experimentally validated. A 1-DoF robotic insertion system with force monitoring was deployed. Experiments are performed with and without control of EA bending and it is evident that there is reduction in insertion forces if bending control is in place. Insertion forces were measured with 2-DoF and 4-DoF robotic systems in [39] and found reduced forces with these systems compared to 1-DoF system. In another study, conducted by Clark [15], reported 50% reduction in insertion forces with magnetically tipped array as compared to non-guided straight EA. Magnetically tipped EA is guided as it is inserted in to the cochlear with the help of a manipulator magnet placed near the patient's head which apply magnetic torque to the array tip causing it to bend away from the ST walls. The detailed working of this system is explained in 2.2.3. Overall, max torque method is experiencing less insertion forces as compared to max field method. Non-guided method is inferior to both the guided methods in terms of insertion forces by 50%.

The research carried out by Todd [46] employed three insertion methods (standard insertion technique, advance off-stylet, and partial withdrawal) on three different electrode array designs (Contour Practice, Contour 24 and Contour Advance) to access the force delivery and electrode trajectories (position) during implantation. Net insertion forces (frictional force, input force from user, relaxation force, and adhesive force) are measured using Instron® 5543 measurement device with 10N load cell and analyzed along with electrode trajectories. The impact of frictional forces (after measuring for varying degree of roughness and lubrication using Tribometer) on final force delivery is separately assessed. It was observed that insertion forces were lower when soft tipped Contour Advance EA was inserted using AOS technique as compared to other EA designs and insertion techniques. Result analysis suggests that carrier strength (due to stylet), contact pressure (soft/hard tip), frictional force (rough/ lubricated surface, carrier contact with ST outer or inner walls), electrode trajectory (turn/straight, close to outer/inner wall) and surgical techniques are all contributing factors in force delivery. In an earlier study by Roland [36], similar results were observed. Forces involved in inserting Contour advance electrode using AOS technique was lower than insertion using SIT (mean 0.012 vs 0.075 N).

Insertion forces were recorded by employing advance off-stylet technique to insert Contour Advance electrodes (Cochlear Ltd.) in to an artificial arcylic ST model using an automated tool [10]. Insertion forces were measured with Instron® 5542 sensor with 10 N load cell which is accompanied with measurement software .20 insertions were performed with different velocities (0.3,0.4,0.5 mm/s) to note its effect on the insertion forces. Tip fold-over insertion failure were observed on 5 instances out of 20. Furthermore, insertion forces rose by 40% at the end of AOS movement. During the instances of tip fold-over, insertion forces instantly increased 8 times with respect to mean value. It was observed that mean insertion forces were higher when array was inserted with lower insertion feed rate. In contrast, results from experiments performed by Kontorinis et al. [47] showed that insertion forces increases with increased insertion speed (insertion speed varies from 10-200 mm/min resulted in mean insertion forces of 180-240 mN). On the same lines, another recent study done by Rau et al. found that insertion forces decreased with ultra low insertion speed of 0.03 mm/s as compared to speed of 2 mm/s and 4 mm/s [48]. According to Zhang [49], insertion forces decreases with increasing insertion velocity. However, Mirior et al. performed similar experiments with straight electrode and did not find any difference in forces with respect to insertion speed.

Schurzig et al. [27] compared standard insertion technique and advance off-stylet techniques in terms of the force profiles using an automated insertion tool with in house force sensing capability. Four insertions were performed with SIT and four with AOS technique in a 3-D ST model. Force profiles were nearly similar upto 7 mm insertion depth (before the first basal turn) during both types of insertion which was averaged at 6 mN. During the insertion depth between 7-17 mm, average forces experienced in AOS insertion were 8±6 mN with maximum of 34 mN. On the other hand, average forces remained at 46±27 mN with a peak value of 93 mN. There was significant difference in force profiles beyond 9.74 mm insertion depth because AOS technique avoids array to touch lateral wall during the first basal turn.

Comparative study was conducted by Majdani [50]between forces due to robotic insertion and human operator insertion. 26 insertions were carried by three different surgeons and 8 insertions by an automated tool. Forces due to robotic insertion were higher than the insertions by surgeons, however, robotic insertion was more reliable with less force maxima.

The test bench for experiments performed by Mirior et el. in [11] using automated insertion tool was discussed in 2.2.2. A straight lateral wall EA was used to perform insertion in a 3D ST model and impact of insertion tube material, ST model/temporal bones and insertion speed rate on the friction forces were examined. Array friction forces (measured with 1-axis force sensor) in plastic insertion tube were lower as compared to metal tube and stand at 90±28 mN versus 140±34 mN. Lubrication inside insertion tube also helped decrease the friction forces. 6-axis force sensor was to measure friction forces between EA and cochlea model/temporal bones and compared with force profiles of 1-axis sensor. There was mean error of 10±2 mN when both force profiles are compared in ST phantom model whereas a mean of 10±1 mN was observed in all temporal bone experiments.

There is dire need to establish strong relationship between insertion forces and the histological proof of extent of the trauma induced. In the experiments conducted by Seta et al. [51], automated insertion tool developed by Mirior [11] was used to implant straight lateral wall EA in 12 cadavaric temporal bone specimens. Relationship between insertion forces and trauma was established after cochlea anatomy CT scan data, array proximity and force profiles are correlated with post operative microscopic images and histological damage assessment. There were 6 traumatic insertions out of 12; translocation in 5 and basilar membrane fracture in one. An early force spike of 0.03±0.018 N in the region of 150-180°insertion depth was observed in traumatic insertion experiments. It was due to array touching the lateral wall on the first cochlea turn. Analysis of Variance (ANOVA) test was carried on force profiles data of both traumatic and atraumatic insertions and found that different values were observed at the same insertion depths ($p<0.001$).

Pressure is directly related to the force and stable intra-cochlear fluid pressure is necessary to preserve residual hearing. Mittman et al. conducted study to find frequency and amplitude of fluid pressure during the EA insertion process [52]. Two types of EAs, midscalar and lateral wall, were used for cochlea plastic insertion. A 0.8 mm diameter micro-optical Pressure sensor was placed at the apex of cochlea model. The lateral wall array has smaller mean volume per mm than the mid-scala type (0.23 vs 0.35 mm^3). It was found out that insertion of midscalar array generates more mean pressure (ranges from 0.3-0.5 mmHg) than lateral wall array (ranges from 0.1-0.3 mmHg). Similar to the concept of integrated steering mechanism of EAs [15][16], Wade et al. presented a custom array design with integrated force sensor at the array tip [53].Optical fiber Bragg grating was utilized for integrated force sensor in custom EA for guinea pigs. Insertions experiments were performed to measure insertion forces on the array tip in real time. Insertion forces were successfully correlated with trauma histological data. The system was able to measure maximum insertion force of 254 mN which was associated with grade 3 trauma. Grade 1-2 trauma had a force range of 76-200 mN Wheres grade 3 has a range of 201-254 mN in 12 experiments.

Besides detection of induced trauma, insertion failure (tip fold-over and buckling) detection is equally important for working atraumatic cochlear implantation. Complications due to tip fold-over insertion failure is discussed in 2.1.4. Pile el al. employed machine learning support vector machine (SVM) algorithm to design a system to predictive tip fold-over failure [54]. The failure was predicted with the help of real time intra-operative insertion forces with force profile data history of traumatic and normal insertions. Insertion force data was collected and used with respect to the insertion depth. Insertions of periomodiolar EA were carried out using an automated tool discussed in 2.2.2 in phantom cochlea model. An accuracy of 88% was accomplished to detect tip fold-over insertion from a normal insertion.

2.3.2 Impedance Measurements

Most cochlear implant systems have built-in impedance measurement mechanism to check functioning of electrodes after the implantation. Impedance measurements and tomography have been employed in various biomedical areas. So research is also on going to use it for localization of electrode array during insertion in ST.

2.3.2.1 Stimulation Strategies

The importance of techniques used to stimulate the hearing nerves through electrodes is significant. It dictates the spread of input current stimuli in the inner ear and effects loudness and pitch levels. Impedance measurement values are also dependent on these techniques. There are five types of stimulation strategies governing the cochlear implants 1) Monopolar 2) Bipolar 3) Tripolar 4) Quadrupolar 5) Common Ground

In monopolar systems (Figure 2.21a), an array (active) electrode in the cochlea generates the field potential with respect to an extra-cochlear return (reference) electrode placed in tempolaris muscle. The current flows between these two electrodes and field spreads along a wider area since there is substantial distance between two electrodes [55].

Common ground strategy (Figure 2.21e) is similar to monopolar except the return electrode is the parallel collection of all the remaining electrodes in the array other than one driving the current. The field distribution in this strategy depends on the impedances of all the return electrodes.

In bipolar devices, stimulative active electrode is in close proximity to the return electrode (Figure 2.21b). They are either next to each other or with a gap of an electrode in between. These intra-cochlear electrodes work in opposite polarity and hence no net current leaving or entering the cochlea. The distance between active and return electrode dictates the amount of current spread and eventually the number of neurons stimulated [56].

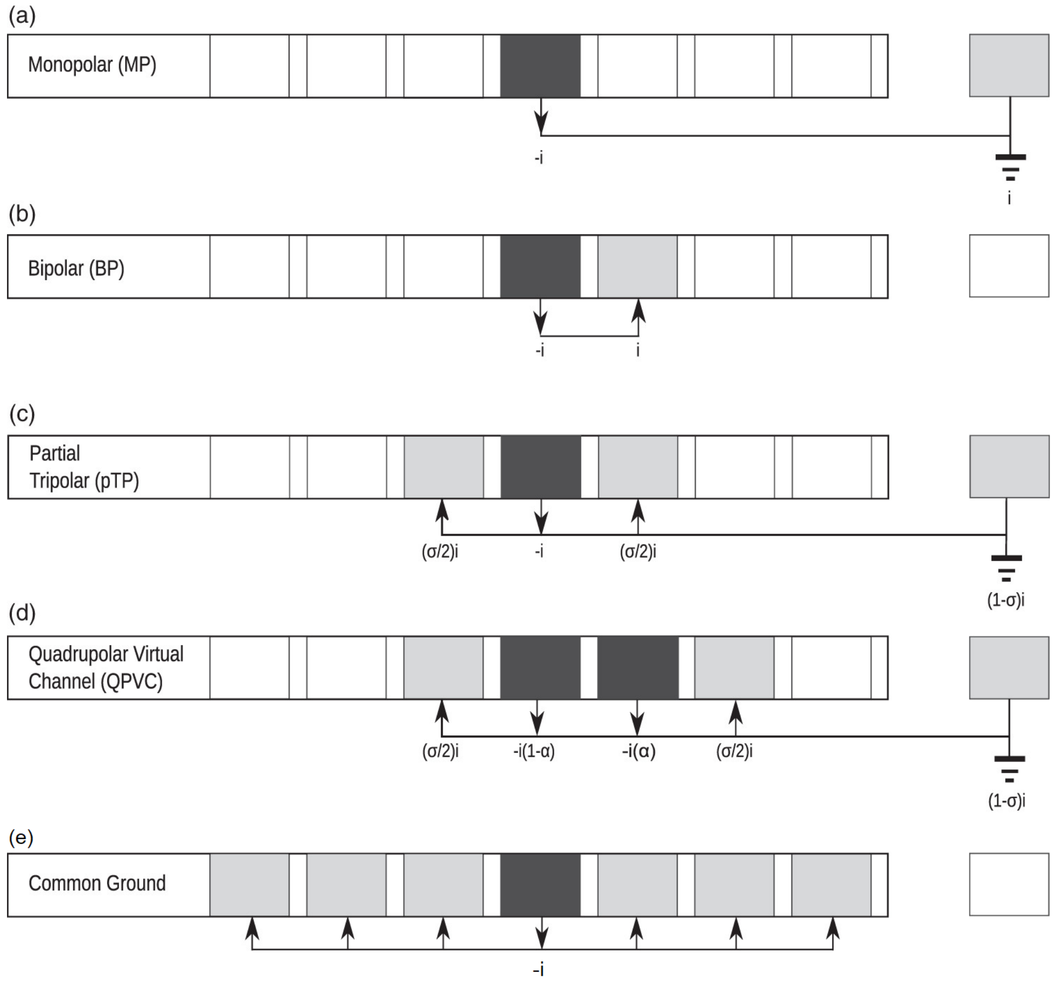


Fig. 2.21 Five stimulation strategies for cochlear implant. a) Monopolar, b) bipolar, c) tripolar, d) Quadrupolar, e) Common Ground Figure from [57]

Tripolar stimulation has three electrodes in action with center electrode acting as an active stimulating electrode whereas two outer electrodes operate in the opposite polarity as center electrode. The algebraic sum of the current in cochlea is zero as shown in Figure 2.21c.

Each stimulation technique has its advantages and disadvantages. For example, loudness level is governed by the number of neurons stimulated and loudness level is higher in monopolar strategy due to wider spatial spread of current but produces poor pitch. On the other hand, bipolar gives more spatial selectivity than monopolar and provides better pitch performance [55].

Zhu et al. compared three stimulation methods in cochlear implants to record their performance in terms of spatial selectivity. They found out after measuring both physiological and psychophysical spatial selectivity that focused stimulation (multipolar) presented sharper spatial tuning than monopolar method. In a similar study conducted by Jolly et al. compared field strength and focus in mono, bi and quadrupolar (a form of tripolar (Figure 2.21d)) stimulation strategies. They also found that quadrupolar has more focused field potential in the layers of ST than other methods. The focused field improves the pitch, reduce channel interference and limit facial/tactial induced stimulaton by current spread.

The stimulation strategy used in the cochlear implant system also influence the impedance equivalent model which will be discussed in next section.

2.3.2.2 Equivalent Circuit

Electrodes are widely used to stimulate living tissues as in the cochlear implant system. Electrodes in CI system are immersed in Perilymp fluid in scala tympani. This fluid is similar to Sodium-Chloride (NaCl) electrolyte solution. Therefore, it is important to study the electrode-electrolyte interface and its corresponding equivalent circuit models.

A. Volta first discovered in 1800 that electro-electrolyte junction has electrical potential which led to the investigation of direct current [58]. Ohm's law was also discovered in 1826 during the study of this interface and states that current is directly proportional to the voltage potential and inversely proportional to the resistance in the circuit[59]. In 1834, Farady coined the term "electrode" during his electro-chemistry experimentation [60]. Earlier electrode-electrolyte interface was considered as a capacitor and it was Varley [61] who first measured this capacitance. He was astonished to measure capacitance value of 311 µF of 1-inch-square platinum electrode. This high value paved the way for electrolytic capacitance discovery. On the circuit modeling front, Helmholtz first pitched the idea of circuit model of electro-electrolyte interface [62]. In 1879, Helmholtz found out that a charged double layer exists at the interface which is equivalent to a charged capacitor (Figure 2.22, A). It was concluded that a resistance must also be part of the circuit because direct current delivered by voltaic cell passes through the interface. Second attempt to model equivalent circuit was carried out by Warburg in [63]. He found out that a resistor and capacitance in series where polarization capacitance C_w and its reactance X_w varies inversely to the square-root of frequency. The reactance X_w and resistance R are found to be equal and phase angle was constant at 45 °. Resistance also varies with frequency as capacitive reactance changes with frequency.

Warburg model was limited to infinitely small current density, on the same line Fricke [64] reported in 1932 that for a low current density and limited frequency range (0.1-3.5 kHz), polarization capacitance and phase angle remains constant.

Randles [65], in 1947 described a model comprised of a polarization capacitance (C_p) in parallel with a series capacitance (C) and resistor (R) as shown in Figure 2.22, C. This model as well as models by Helmholtz, Warburg and Frikie did not account for direct current property of interface.

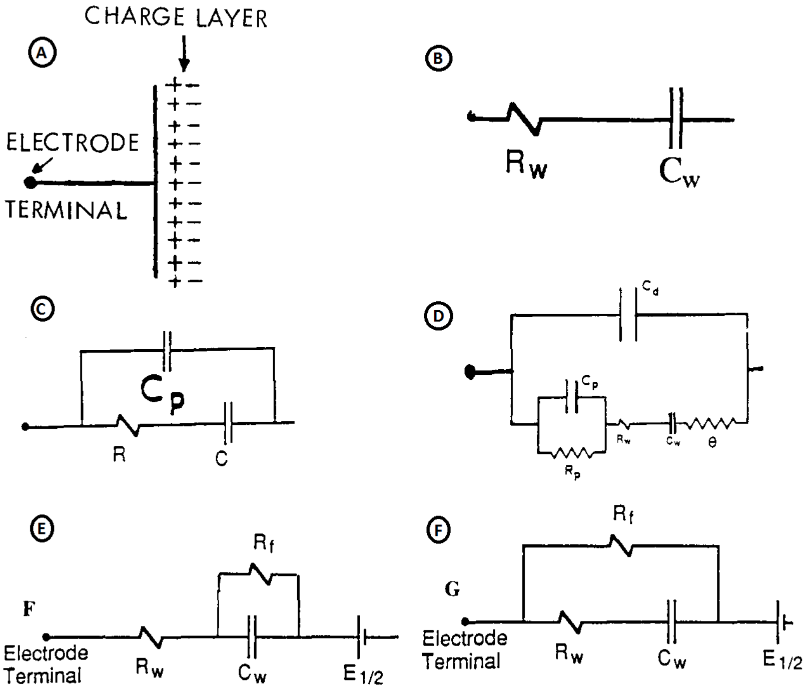


Fig. 2.22 Transformation of the electrode-electrolyte equivalent model. Model by (A) Helmholtz [62] (B) Warburg [63] (C) Randles [65] (D) Sluyters-Rehbech [66] (E)-(F) Geddes-Baker [67]

In 1968, Geddes and Baker suggested two models on the basis of various frequency-Impedance (f-I) curves of electrodes in saline solution. One of the models has Warburg capacitance (C_w) which was placed in parallel with a Faradic resistance (R_f); this combination is in series with half-cell battery ($E_1/2$) and Warburg resistance (R_w). In the other model, Warburg capacitance-resistance pair ($C_w R_w$) was shunted by Faradic resistance (R_f). A half-cell potential is placed in series with this combination. The shunted Faradic resistance (R_f), in both models, account for direct current property of the electrode-electrolyte interface.

In 1970, Sluyters-Rehbach [66] presented a rather complex model of electrode-electrolyte interface as presented in Figure 2.22, D. This model contains the Warburg impedance ($R_w C_w$) and a resistance (θ) in series with a parallel resistance-capacitance pair ($C_p R_p$); this combination has then a parallel capacitance (C_d).

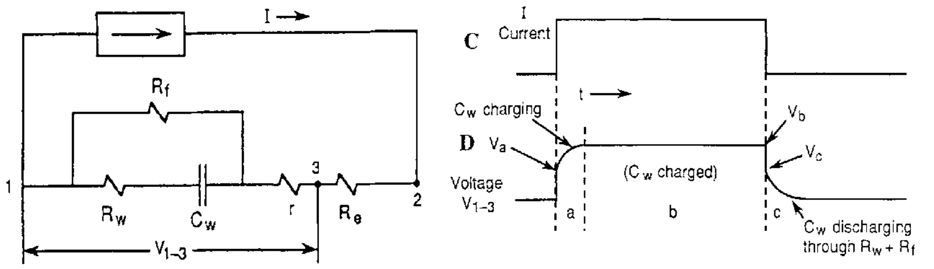


Fig. 2.23 A method to measure potential of electrode-electrolyte interface. Left: Equivalent circuit. Right: Current-Votlage waveform [68]

In 1974, Carr et al. [68] proposed a method to measure potential of a single electrode-electrolyte interface. A current stimulus is applied across test electrode 1 and a reference electrode 2 whereas potential is measured between electrode 1 and electrode 3 as shown in Figure 2.23, left. The resistance (r) resides between electrode 1 and potential-sensing electrode 3 and (R_e) represents resistance between electrode 1 and reference electrode 2 where both these resistances are due to electrolyte between electrodes. According to Figure 2.23, right, voltage appearing across test electrode 1 and 3, when a long constant current stimuli is applied across electrode 1 and 2, can be divided in three phases. One of the phases, voltage (V_a) is across a series combination of r and parallel $R_w R_f$; next the curve presents charging of capacitor C_w. The voltage V_b is across series combination of resistances r and R_f as the capacitor is fully charged and V_c represents the voltage when capacitor C_w is discharging through series R_f and R_w.

Geddes and Roeder, in 2001, did experiments to measure direct-current Faradic resistance (R_f) in the electro-electrolyte interface [69]. Figure 2.24 shows Warburg circuit model between two electrodes. The reactance X_w and Faradic resistance R_f provides the direct current in the circuit. The resistance R_s is the resistance between two electrodes due to electrolytic solution between them. It is understood that polarization capacitance C_w and R_w decreases with increasing frequency. Therefore, at very high frequencies, the total impedance is equivalent to the resistance R_s which can be easily found with frequency-impedance curve. In order to find Faradic resistance R_f, a square wave current stimuli was applied across the electrodes, the pulse width of which is longer than the time constant τ of the circuit. The current will flow through R_{f1},R_s,R_{f2} when C_w is fully charged during the current pulse. Therefore, the voltage between two electrodes would be presented as $E_f = i(2R_f + R_s)$ as R_f is same for both electrodes. Faradic resistance R_f can now easily be calculated by $R_f = 0.5(E_f/i - R_s)$.

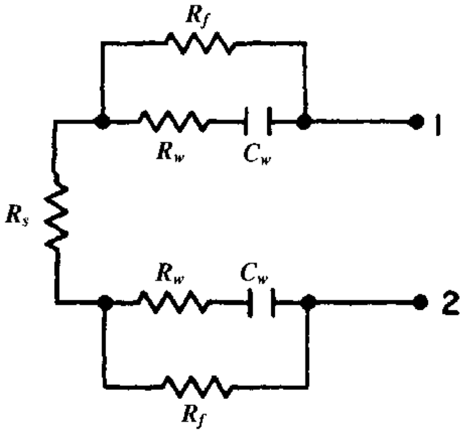


Fig. 2.24 Bipolar impedance model in saline solution (between two electrodes). Figure from [69]

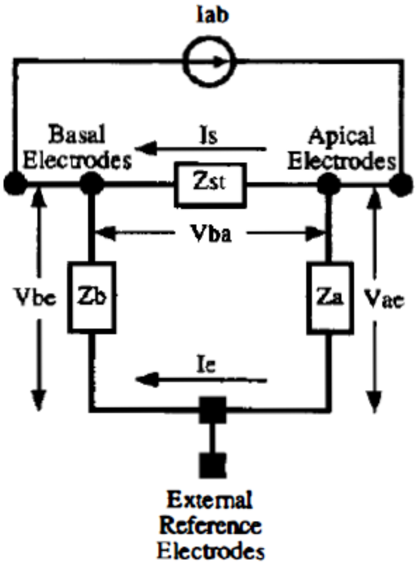


Fig. 2.25 Lumped impedance model of implanted cochlea. Figure from [70]

Suesserman and Spelman came out with a lumped parameter model in 1993 that approximately simulates electrical properties of an in-vivo implanted cochlea [70]. In this lumped model (Figure 2.25), 2 pairs of interscalar electrode and a single pair of extrascalar (reference) electrodes are used to measure one intrascalar (Zst) and two transscalar (Za, Zb) impedances within an implanted cochlea. The electrode pairs are believed to be at same potential due to their close proximity. One of the electrodes in a pair drives the current while the other pair measures the potential. This way, three unknown cochlea tissue impedances can be measured.

The impedance model proposed by Vanpoucke et al. in [71] and shown in Figure 2.26 contains components of both electrode-electrolyte interface impedance as well as tissue impedance. The arrow in the figure shows monopolar impedance model when electrode 2 is active where the excited electrode is inside cochlea and reference electrode is external to cochlea. The current path is through the interface impedance Z_C (capacitance and resistance in parallel), tissue impedance Z_TISSUE (a resistance) and through to the external ground electrode. This work successfully analyzed the structure of intra-cochlear potential measurements with respect to cochlea anatomy and electrode array design.

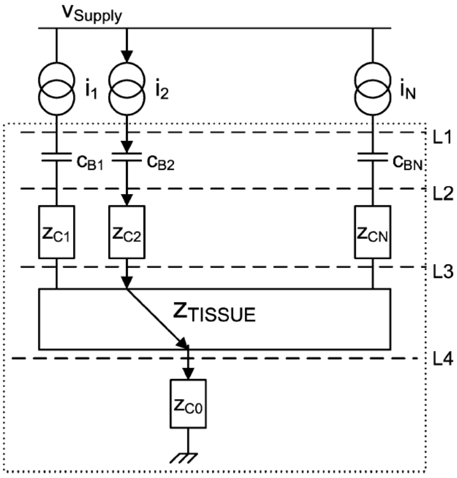


Fig. 2.26 General impedance model of electrode array in scala tympani. Arrow shows a path of monopolar current flow from intra-cochlear stimulating electrode to outer ground electrode [71].

2.3.2.3 Impedance for Insertion Guidance

The first real-time impedance measurement system during array insertion was developed by Tan et al. [12]. Both monopolar and bipolar impedance measurements were taken during insertion with AOS technique to observe impedance change after stylet removal. Impedance was measured during insertion in human cadaver as well as in live surgeries. A biphasic current pulse of 25 µA width and an amplitude of 74 µA is used as input stimuli and corresponding voltage is measured by continuously sequentially scanning the electrodes. Impedance is then calculated by dividing voltage by current. This software system interacted with a commercially available cochlear implant system and hence no extra hardware was required. Using monopolar stimulation technique, impedance change was more pronounced in basal end electrodes whereas it was more prominent in apical electrodes when using bipolar technique. A correlation between impedance values and intra-cochlear array position was also seen where high impedance values observed near the cochlear walls. Monopolar measurement technique delivered better results than bipolar in terms of EA localization. Certainly, it was the first step to use impedance measurements as a way to control EA position during insertion for atraumatic process.

Electrical bipolar impedance was measured during insertion of perimodiolar electrode array using two different insertion techniques (SIT and AOS) in order to evaluate different impedance measurements and conclude relationship between impedance and proximity of EA from modiolar wall. Total impedance comprises of three key impedances; electrode impadance (Ze), cochlear impedance (Zc), fluid impedance (Zf). Relationship between the impedance measured value and proximity of EA to the wall was attributed to the amount of fluid between the EA and the modiolar wall at a specific contact. Analysis of variance (ANOVA) test was applied to the filtered data set (implant used, bone specimen and insertion technique used) and found out there was clear separation between SIT and AOS impedance measurements where impedances during AOS insertion are higher than SIT. It was also concluded that impedance values are dependent on bone and EA properties. Moreover, conductive fluid between electrodes would be more significant in bipolar than monopolar measurements.

The impedance of each electrode in the array is usually measured after the surgery to check any short/open circuit scenario. The current method only tests the overall impedance and not its components. Tykocinski et al. proposed an impedance equivalent model to evaluate impedance components; access resistance and polarization impedance, which is necessary to know the symptoms of changing impedance. The degradation of electrode surface would change polarization impedance whereas change in access resistance would be governed by intra-cochlear environment [72]. Giardina et al. used this model/method to investigate relationship between EA monopolar impedances (Access resistance (Ra), polarization resistance (Rp) and polarization capacitance(Cp)) and its proximity in ST during implantation by employing linear regression model (independent variables were insertion depth and impedances to predict dependent variable-distance to the modiolus)[13]. Voltage response of each electrode, to a biphasic current pulses (amplitude of 34 µA, lasting 179.6 µsec per phase and separated by 100 ms), were recorded at 56 kHz sampling rate by using 9 bit resolution A/D converter. Total of 272 (saline+16 EA insertion depths; 17*16=272) impedance recordings were made per EA insertion. By measuring the impedances of each electrode during insertion, it was observed that deeper locations have a more resistive path to ground. Although distance of EA to the modiolus changes during insertion, however, at the base of the cochlear, the distance from the electrodes to the modiolus was large, array was consistently close at position 8 and array remained roughly at 1.5mm away from modiolar wall. When 8 electrodes were inserted, Ra was higher in the electrodes (1-4) close to the inner wall and when 12 electrodes were inserted, Ra saw exponential increase in electrode (4-8) because these electrodes were close to the modiolus. On the other hand, Cp decreased with the insertion depth and proximity to the modiolus. Total impedance of the system was mostly governed by Ra. After applying linear regression model, a Pearson correlation of 0.885 is achieved. Impedance increased (Ra increased, Cp decreased and Rp slightly increased) with insertion depth in ST and proximity to the modiolar (inner) wall. However, polarization components contributed a small (2.5%) but significant portion to this correlation.

In 2017, Karunasiri was awarded a patent for developing systems and methods to measure real time electrodes impedance during surgery [73]. The patient receives a continuous audio signal of different frequencies through the system during the surgery. Electrodes in the array are then directed to stimulate the nerves by the application of a current signal representing the audio signal. The voltage associated with the current is then measured to calculate the impedance of respective electrode. The real time impedance measuring system as well as impedance data storage are embedded into the processor. This system also allows to adjust the stimulation parameters according to the impedance change during/after the surgery. Stimulation parameters include pulse width, duty cycle, amplitude, waveform type etc.Stimulation management facility communicate with impedance system to perform the corrective measure according to the impedance values.

It is critical to save facial nerve from any trauma during the drilling process in cochlear implantation. Anso et al. conducted a research to find a relationship between impedance and tissue density during drilling [74]. In vivo experiments were performed on 5 sheep with nine different trajectories using an image guided robot. Impedance measurements were taken at 5 points around facial nerve. A monotonic decrease in impedance was recorded when the drill axis was intersecting the facial nerve. These results may be used in future to accurately approximate the proximity of the drill axis from the facial nerve based on tissue density and impedance correlation.

2.4 Machine Learning

Machine learning is a sub-field of artificial intelligence where algorithmic models are trained to learn from data to optimize the performance of a certain task rather than explicitly code to do so [75]. Machine learning provides new tools and techniques/methods to solve complex problems where classical statistical techniques fail. In the last decade, the explosion of large amount of data (due to advances in data collection/generation and data storage capabilities) and exponential growth in computing power led to unprecedented interest in

50 years old field of artificial intelligence. This interest saw innovative growth in machine learning algorithms during the last 2 decades. This also gives rise to growth of its subfield of deep learning which potentially needs more data to learn than classical machine learning but achieves highly accurate results. Most commonly, machine learning is used for classification, regression, clustering, dimensionality reduction of large datasets in the domain of image and speech recognition [76, 77], natural language processing, time series data analysis and so on. Machine learning can be categorized in to three main domains; supervised learning, unsupervised learning and reinforcement learning. Our work is mainly attributed to supervised learning (classification and regression) where algorithms are designed to predict or classify a desired outcome (discrete/category or continuous/real value).

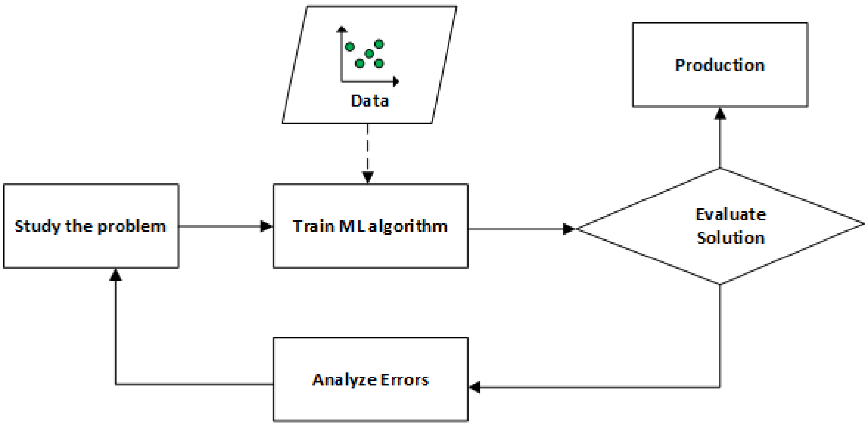


Fig. 2.27 Machine learning approach

2.4.1 Supervised Learning

Supervised learning extends the notion of standard curve fitting procedure where it tries to learn the unknown function which connects inputs to desired outputs. This is estimated by extrapolating the patterns found in the labelled training data [78]. Supervised machine learning is further categorized in to two groups; classification and regression depending on the output

2.4.1.1 Classification

The goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown classification problems in which the output of instances admits only discrete, unordered values. [79]. Lets we have a training set $S = \{x_{(1)}, x_{(2)}...x_{(n)}\}$ of feature vectors where $x_i \in X$ represented in a feature space X. Training set is sampled according to unknown probability distribution. A unique $x \in S$ is uniquely associated with a target label $y \in Y$. Here X is a real valued vector space and $Y = \{y_1, y_2...y_L\}$ is a set of L discrete (different) class labels. The classification learning algorithm C seeks to map $X \rightarrow Y$ keeping the loss function $P(x, f(x))$, that measures the error, to the minimum. In other words, algorithm tries to find a function f that minimizes the expected error on the unknown data.

2.4.1.2 Regression

In regression, the output targets are continuous values rather than discrete values as in classification. Let input X be a d-dimensional vector and output Y a q-dimensional vector, a regression algorithm maps input X to Y, $M : X \rightarrow Y$ on the basis of an input sample set $X = x^{(1)}, x^{(2)}...x^{(n)}$ and corresponding output target set $Y = y^{(1)}, y^{(2)}...y^{(n)}$ where $y^{(i)} = M(x^{(i)})$ and $y_{(i)}$ is real continuous value.

Linear Regression: Linear regression plays an important role in statistical learning. Take into account the regression issue where a continuous response Y is to be regressed on a number of variables x1,...,xp. It is well known that the simplest model to represent the regression function as a linear combination of predictors is linear regression. In instance, linear regression frequently offers a good approximation to the underlying regression function, especially when the sample size is small or the signal is very faint.

Simple/single-variant linear regression model has one independent variable x and its estimated regression function is represented as in 2.1

$$f(x) = b_0 + b_1 x \tag{2.1}$$

where b_0 is the intercept and b_1 is the slope. Multiple linear regression model is when there are two or more independent variables $x_1, x_2, ...x_p$ and its estimated regression function can be represented as in 2.2

$$f(x_1, x_2, ...x_p) = b_0 + b_1 x_1 + b_2 x_2 + ...b_p x_p \tag{2.2}$$

The objective of regression is to find the weights b_0, b_1, and b_2 values that provide the minimum SSR (the sum of the differences between the predicted value and the mean of the dependent variable) while bringing this plane as close as possible to the real responses. Model estimate entails estimating all of the model's parameters. For linear models, a variety of estimate techniques are available, such as least squares, maximum likelihood, Bayesian approach etc.

2.4.2 Unsupervised Learning

There is absence of the supervisor in this type of learning, therefore it is absolute error measures. Unsupervised learning methods aim to find relationships and patterns in a data structure without having a measured outcome (unlabeled data). This is called unsupervised because it lacks a response variable that can supervise the analysis. The aim of unsupervised learning is to find underlying dimensions, components, or clusters in an unlabeled data.

2.4.3 Reinforcement Learning

In Reinforcement Learning, the system learns from its interaction with the environment. The learning system (agent) observes the environment and perform actions, and in return get either the reward or penalty according to its good/bad actions. Its aim is to maximise the reward over time and learn the best strategy (policy).

2.4.4 Time Series Classification

Time Series Data: A time-series T is a sequence of measurements of size m taken over a period at regular intervals where $T = (t_1, t_2, \ldots, t_m)$. Since temporal information is present in a majority of everyday tasks, be it finance, healthcare, or sales – time series analysis is an active area of research. In machine learning, time series classification refers to the process of learning meaningful information from temporally ordered data and classify each example series into a class. This has led to it being studied extensively under various machine learning algorithms [80–83].

2.4.4.1 Classification Algorithms

Classification algorithms aimed to learn simple functions from the training data and output a hypothesis that perform the correct classification of the training data. Generalization is the term used for the ability of hypothesis to correctly classify the unknown data (test data) not present in the training data.

Support Vector Machines: Support Vector Machines (SVM) were initially designed as discriminative functions and were mainly used for large margin classification. Simply put, given a set of training examples, the SVM classifier learns the position of its support vectors and then subsequently the optimal hyperplane to categorize unseen samples. Owing to their impressive generalization ability, SVMs have been extensively studied to solve a number of classification tasks (see e.g. [81, 84, 85]). More recently, their uses cases have expanded to include function approximation, regression, and time series prediction by introducing an appropriate loss function [86]. Fig. 2.28 represents how support vectors (dotted lines) and decision boundary (solid line) maximally separates different classes in the case of linear SVMs. In SVMs, a separating decision surface is formed after the data is transformed in to higher dimensional space. This process is known as kernel function. There are two types of SVM, linear (for separable data) and non-linear (for non-linearly-separable data).

Lets take an example of linear SVM and its classification mechanism in terms of support vectors and hyperplane. We have training data $\{x_i, y_i\}$ where $i = 1...n$ and x is a d-dimensional real vector $x \in R^d$. The output $y \in \{-1, 1\}$ indicating the class x_i belongs to. The aim of the algorithm is to find a separating hyperplane with the highest margin that separates the input training samples belonging to each class, -1 and 1. A hyperplane can be defined as a set of points satisfying:

$$w^T x + b = 0 \tag{2.3}$$

where w is normal direction of the hyperplace and b is a form of threshold. Let's margin between the separating hyperplanes be defined as $d_+ + d_-$ where d_+ is the shortest distance between the separating hyperplane to the positive example whereas d_- is the shortest distance to the negative example. Margin can also be defined as $2/||w||$ where $||w||$ is the euclidean norm of w. The aim of the algorithm in this case (data is linearly separable) is to find largest margin between the separating hyperplanes [86]. Since the data is linear separable, we can have two parallel hyperplanes separating the 2 classes keeping the margin as large as possible as shown in Fig. 2.28. The maximum margin hyperplane is the one that is halfway between them and can be described by 2.3. Those x_i that lie closest to the max-margin hyperplanes are called support vectors as shown in Fig. 2.28. Hyperplanes separating the classes can be described as:

$$w^T x + b = 1 \tag{2.4}$$

$$w^T x + b = -1 \tag{2.5}$$

Each training example x_i need to hold the following constraint

$$w^T x_i + b \geq 1 \quad for\, y_i = 1 \tag{2.6}$$

$$w^T x_i + b \leq -1 \quad for\, y_i = -1 \tag{2.7}$$

Eqs 2.6 and 2.7 can be combined to one set of inequality

$$y_i(w^T x_i + b) - 1 \geq 0 \tag{2.8}$$

For the training data that is not linearly separable, we use a technique called kernel trick on max margin hyperplanes (originally proposed by Aizerman et al [87]) to develop non linear classifiers. It is generalization to the linear classifier technique except the dot product is replaced by a non linear kernel function.

Standard SVM is only used for binary classification, however, multiclass classification can be solved by decomposing the problem in to several multiple binary problems where standard SVM can be used [88]. Having said that, SVM model in python sklearn library deals with both binary and multiclass classification.

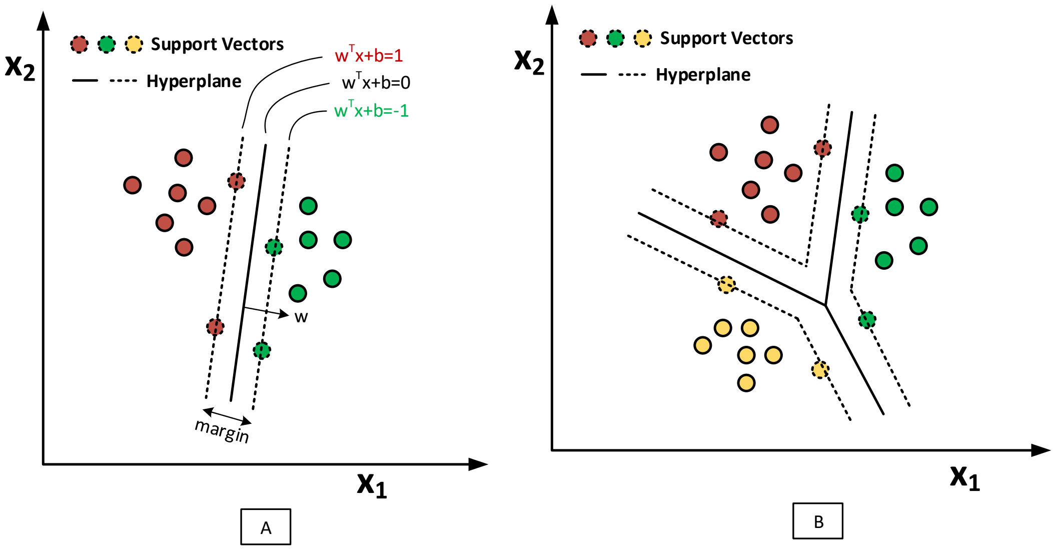


Fig. 2.28 Linear Support Vector machine as multiclass large margin classifier

***k*-Nearest Neighbours**: *k*-Nearest Neighbor (*k*NN) classifier is a fairly simple and straightforward supervised learning algorithm which classifies a data point based on how its neighbors are classified. *k*NN algorithm is based on feature similarity, that is, choosing the right k (number of neighbors) value is important for better accuracy. Another parameter is the distance measure that induces the relationship between neighbors [89], for example, Euclidean distance, Dynamic Time Warping (DTW), Manhatten distance etc. [90] found that a simple 1-NN classifier with DTW distance measure generally produces better results than more complex classification algorithms on time series data. DTW distance metric helps when two time series are fundamentally similar but has some degree of distortion or noise. It uncovers underlying similarities by stretching (warping) the time axis [91]. The DTW distance between two points q and x can be defined as

$$DTW(q,x) = \min_{\pi} \sqrt{\sum_{i,j \in \pi} d(q_i,x_i)^2} \tag{2.9}$$

Where $\pi = \{\pi_1, \pi_2, ...\pi_L\}$ is the optimum mapping with the following properties
$r = |q|$, $s = |x|$
$\pi_1 = (1,1)$ and $\pi_L = (r,s)$
$\pi_{l+1} - \pi_l = \{(1,0),(0,1),(1,1)\}$

Fig.2.29 presents the concept of $k$nn algorithm where distance from a new data point will be calculated to the nearest 3 neighbours (k=3), and votes for the most frequent nearest neighbour class. In this example, new data point belongs to class B.

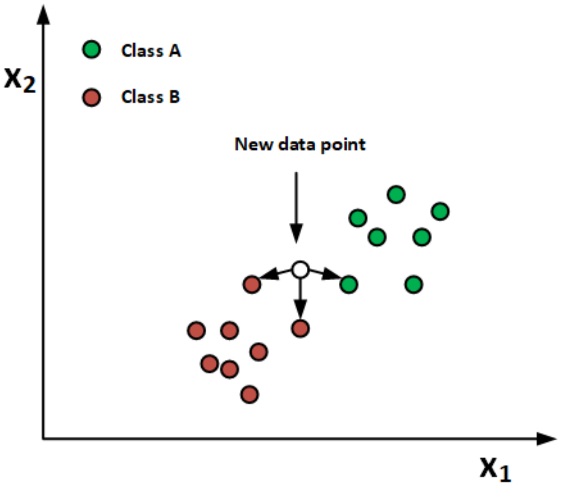


Fig. 2.29 k-Nearest Neighbour (k=3)

Artificial Neural Network:Artificial Neural Networks (ANN) are complex computational models that learn to solve a given task by extracting meaningful information from the training data. ANNs were first introduced in 1943 [92] and the authors presented a simplified version of computational model based on how neurons in the brain perform complex tasks. As opposed to SVM, which is a kernel-based algorithm, an ANN is a data-driven algorithm and typically performs better on large training sets.

An ANN is composed of simple computational nodes, called neurons, typically arranged in the form of layers as shown in (Fig. 2.30). Each layer performs a simple linear transformation on the incoming data, followed by a pointwise non-linear function. These seemingly simple models have proven to be immensely powerful in solving problems that have resisted the efforts of the scientific community for many years.

Fig. 2.30 shows an artificial neural network that consists of an input layer (left most) with N neurons, one hidden layer (center) with M neurons and output layer (right most) with K neurons. Input neurons are equal to the input dimension and output neurons are equal to the output dimensions (for example, number of classes in a classification problem). A neuron is connected to another neuron with a weight value that signify the importance of that connection. All neurons other than input neurons has a bias attached to it which caters for unforeseen factors. Hidden and output layer neurons perform a summation function and pass the result of summation to an activation function. There are different activation functions that we can use, for example, sigmoid activation, hyperbolic tangent (tanh), and rectified linear units (ReLU) to name a few. If we take top hidden layer neuron as an example, its output a_1 can be calculated as

$$a_1 = f(b + \sum_{i=1}^{N} x_i w_i) \tag{2.10}$$

where N is the number of inputs from the incoming layer. Each input is multiplied with its respective weight and aggregated. A bias term is added to the aggregated value. An activation function f would be then applied as shown in 2.10. A generalized equation for the hidden layer in the Fig. 2.30 can be given as

$$a_j = f(b_j + \sum_{i=1}^{N} x_i w_{ij}) \tag{2.11}$$

A generalized equation for the output layer can be formed on the same line as

$$o_k = g(b_k + \sum_{j=1}^{M} x_j w_{jk}) \tag{2.12}$$

The activation functions can be linear or non-linear, there would be no change if we choose a linear function, ie., $f(z) = z$. Non-linear activation functions can be defined as

- **Rectified Linear Activation (ReLU)**: The output of this function would be same as input if input is positive else 0, ie., $f(z) = max(0, z)$.
- **Sigmoid**: Its value ranges from 0 to 1. If the input value is positive, it outputs the value from 0.5 to 1 whereas if the input is negative, it outputs the value between 0 to 0.5. It has a S shaped curve. It can be defined by the formula in 2.13

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.13}$$

While training the neural networks, there are multiple iterations when the training examples are iterated through the network to minimize the loss function (This is called forward pass). Loss function is the difference of the predicted value and the actual target value. There are various loss functions such as maximum likelihood cross entropy, mean squared error. The weights are randomly initialized in the beginning and adjusted in subsequent iterations according to the loss. The gradient of the loss function is calculated with respect to each weight that has been set and multiplied with the learning rate and that specific weight is updated by that number. The same method is repeated for all weights (This is called back propagation). When the weights are updating, we say the network is learning.

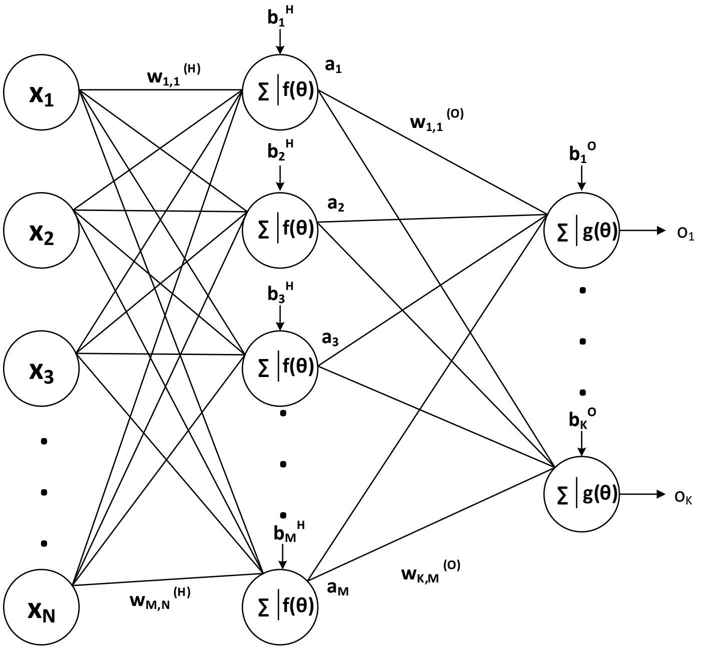


Fig. 2.30 Artificial Neural Network [Input-Hidden layer-output]

Random Forest Random Forest (RF) classifier first introduced by Breiman [93]. It is an ensemble of relatively uncorrelated decision trees where they vote for the most popular class and the class with most votes becomes prediction of the model for a particular sample.

A decision tree is a set of nodes and leaves that is constructed based on training set constituted of feature collection. The purpose is to divide the training set in to smaller subsets by carrying out sequence of tests. When a subset contains only the cases belonging to a single class, the process terminates. These type of tests are called the nodes and subsets are known as the leaves. Predicting the class for a new data sample, the process starts at the tree root and finishing up at one the leaves.

The generalization error of RF depends on the individual trees and their correlation. Each tree in the forest picks random set of features to split each node, which results in lower correlation among tress and the model becomes more robust to noise. RF classifier is also attributed to handling non linear tasks quite efficiently [94].

2.5 Deep Learning

Classical machine learning techniques are limited in processing data in its raw form. They need domain expertise to build feature extraction methods that transform the raw data into a feature vectors. These feature vectors are then used by machine learning algorithms for classification or regression. Deep learning is a representation learning based method consisting of multiple layer of representations where raw data is fed into the model that automatically detects representation for classification or regression [95]. Convolutional and recurrent neural networks are two widely used deep learning models discussed below.

2.5.1 Convolutional Neural Network

CNNs have been widely used for computer vision application but recently they have been utilized for other analyses as well such in the domain of natural language processing and time series analysis [96]. CNN is different from fully connected multilayer perceptron network in a way that it shares weights. It is a type of feed forward artificial neural network that detects pattern in the data and make sense of them. This type of network is usually employed to extract spatial features. This property makes it most favourable for 2 and 3 dimensional data such as images and video frames. CNNs started off with 2 dimensional architecture, however, most recently 1 dimensional CNNs are introduced for 1-D data in the biomedical field for classification and early diagnosis, structural health monitoring and for other sensor based systems [97]. Convolutional layers in the CNN are what that separates it from the standard fully connected neural network. There maybe other layers involved in the CNN depending on the application such as pooling layer, dropout layer and fully connected layer. The convolutional layer is made up of a stack of filters and these filters are actually backbone of detecting the patterns or extracting features from the input data. For example in case of computer vision problem, some filters would detect edges, others may detect different objects and textures. This property sets CNN apart from traditional machine learning algorithms where features are extracted manually (also known as hand crafted features). A CNN could have different convolutional layers with different set of filters, and deeper filters can detect more sophisticated features/patterns. We have to specify number of convolutional layers, number of filters in each layer, and size of those filters. In 2D CNN, a filter is relatively a small matrix where we choose the number of rows and columns and numbers with that matrix are initialized as random numbers. The filter slides across the 2D data according to the filter size until it does so over all blocks of the data. This sliding procedure is called convolving. It is based on convolution operator rather than general matrix multiplication. Convolving is the process of taking dot product of the filter values with the data and the resulting value from this dot product will be stored. This procedure is repeated for whole data matrix and we get a new matrix to be processed by the next layer. As mentioned earlier, there are different filters in a layer and these are called kernels. The data matrix is convolved with each kernel separately.

With the success of 2D and 3D CNNs, there are also 1D CNNs for time series or 1-dimensional data. So instead of 2D filters, there are 1D filters to slide over the 1D data and perform convolution. They produce 1D output for further processing in next layers. Time series exhibit one dimensional data (time) instead of 2-D (width and height). Filters provide non-linear transformation of 1D data. The convolution to the time series data can be presented as

$$C_t = f(w * X_{t-l/2:t+l/2} + b) \tag{2.14}$$

where C presents the outcome of convolution process (dot-product) resulting from application of a function f to a univariate time series X with filter w of length l. There is an additional bias factor b. f represents the final non-linear function applied on the convolution result for example a ReLU or sigmoid function. Convolving 1D data with multiple filters in a layer produces a multivariate output whose size is equal to number of filters. Fig 2.31 shows an example of 1D CNN filter of size 3 sliding over the input of size 6 and producing a feature map of size 4. Using different filters in a layer provides learning process of finding several discriminative features. Filter values are initialized randomly and they are learned automatically as the process goes according to the dataset. Pooling layer can be added after the convolution layer to reduce the size of the convolutional layer output by aggregating. Another type of layer called the normalization layer to help network converge faster. The biggest advantage of the CNN training is its speed compared to the conventional fully connected networks due to the weight sharing mechanism. Fig 2.31 shows an example of 1D CNN filter of size 3 sliding over the input of size 6 and producing a feature map of size 4.

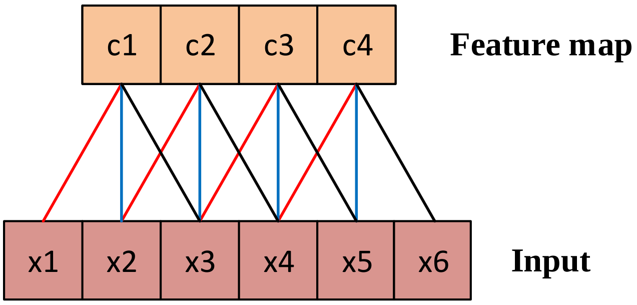


Fig. 2.31 Convolution operation of 1-dimensional CNN

2.5.2 Recurrent Neural Network

The data where temporal information is important, it is often advised to use RNN [95]. RNN processes sequential data better one element at a time and maintains history of the past elements of the sequence in its state vector. RNN has however one disadvantage during the training process, that is of vanishing or exploding gradient during back propagation (gradient value is the value that is needed to update the weights along with learning rate during packpropagation process). This is due to ever changing (growing or shrinking) of backpropagated gradients at each time step over a long sequence. This problem is solved by a different kind of architecture called Long Short Term Memory (LSTM) network that is a special type of RNN. It solves the problem with memory blocks in its hidden states along with three other units in its architecture namely input gate, output gate and forget gate. LSTM network provides short term memory to the network. The internal gates are used for data flow control in the memory blocks. These gates are intelligent in a way to decide which information is important to hold and other information to forget. Therefore, the network only learns from useful information and pass that information on to the long chain of sequences to make predictions. It works similar to human brain where it remembers the important points and throw away irrelevant data.

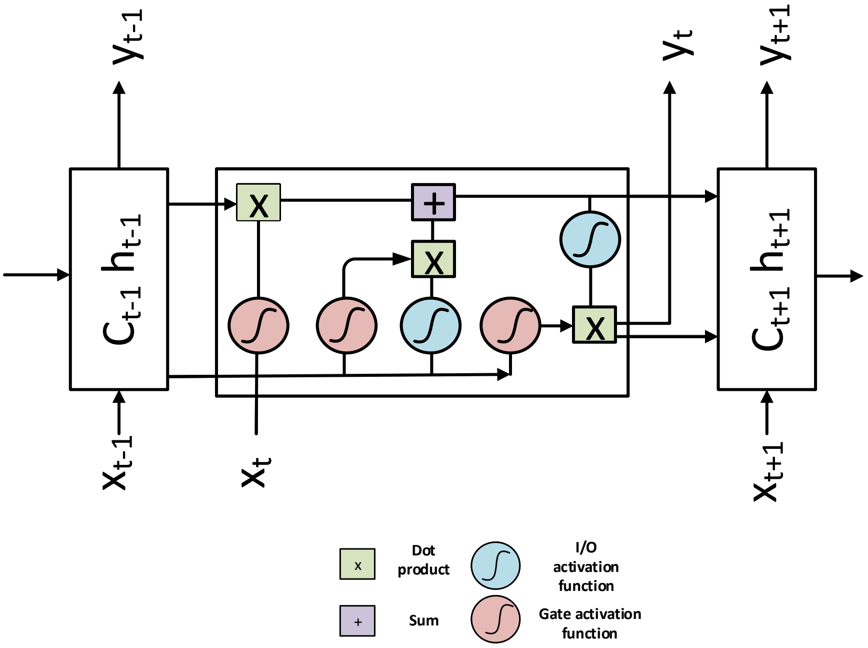


Fig. 2.32 Long Short Term Memory network block with key operations

Fig. 2.32 shows inner structure of an LSTM block/cell and operations involved in it. These operations allow LSTM network to keep or let information during the training process. There are three LSTM blocks in the figure and center one is elaborated. The other two blocks will help understand the flow of information from previous to next block. Memory block mentioned above is also called cell state. Cell state is like memory that carries the information along the sequence of data. Gates decide which information to keep in the cell state. In fig. 2.32, X is the input sequence, and C is cell state, h is the hidden state. We have three neurons with sigmoid activation functions in pink (from left to right) first sigmoid is the forget gate, second sigmoid is the input gate and third sigmoid is output gate. The sigmoid activation function outputs the value between 0 and 1. Any value close to 0 will be forgotten and values close to 1 will be kept in the cell. In the forget gate, current input X_t and previous hidden state h_{t-1} values are combined to form a vector and multiplied with the weights before passing on to the sigmoid function, the resulting value will decide whether to keep or forget it according to rule mentioned earlier. Output of the forget gate for time sample t can be presented as

$$f_t = \delta(w_f[h_{t-1}, X_t] + b_f) \tag{2.15}$$

where f_t is the output of the forget gate, δ is the sigmoid function, w_f and b_f are the weight matrix and bias associated with the forget gate neuron, h_{t-1} is the hidden state of the previous cell, and X_t is the input. Before getting in to the input gate, its important to mention another activation function involved that is tanh in the blue color neurons. These are input/output activation function. To update the cell state, h_{t-1} and X_t are combined to form a vector and multiplied with weight matrix associated with input gate before forwarded to the input gate sigmoid function and same information is passed on to tanh function. The input gate output and tanh function are presented as in 2.16 and 2.17

$$i_t = \delta(w_i[h_{t-1}, X_t] + b_i) \tag{2.16}$$

$$ti_t = tanh(w_{ti}[h_{t-1}, X_t] + b_{ti}) \tag{2.17}$$

Tanh function outputs the value between -1 and 1 to regulate the network. Dot product of the output from tanh and sigmoid function are taken and this product would help decide which information from tanh function to keep or left out. In order to find the cell state, forget state output is pointwise multiplied with previous cell's cell state and added to the input gate output. This process updates the current cell state value and is presented as in 2.18.

$$c_t = (i_t \cdot ti_t) + (f_t \cdot c_{t-1}) \tag{2.18}$$

In the output gate, it calculates the hidden state that keeps the information of the previous inputs. First, previous cell hidden state and current input are combined to form a vector, multiplied with the weight matrix associated with the output neuron and is fed in to the sigmoid function. It can be presented as in 2.19

$$o_t = \delta(w_o[h_{t-1}, X_t] + b_o) \tag{2.19}$$

On the other hand, cell state is processed through a tanh function. The dot product of output from the sigmoid function and the output from tanh output are taken to decide what information would be carried in the hidden state. In this way both the cell state and hidden state are carried to the next LSTM cell. The current cell hidden state and output can be presented as in 2.20 and 2.21

$$h_t = o_t \cdot tanh(c_t) \tag{2.20}$$

$$y_t = w_y h_t + by \tag{2.21}$$

2.6 Conclusion

Robotic systems and sensing methods for atraumatic insertion of cochlear implant have been thoroughly researched and presented in this literature review. Sensing methods for electrode array intra-cochlear localization are divided in to two categories; one category is sensing methods for existing commercially available electrode arrays and other type is custom designed electrode arrays with built in sensing capabilities. The former category includes force and impedance sensing capabilities. Force sensing can primarily judge the extent an electrode array is touching the intra-cochlear walls, however, it is not suitable for proximity approximation of electrode array from the walls. Impedance sensing is considered a tool to characterize scalar position and optimize intra-operative placement of electrode array. Different impedance measurement techniques such as bipolar and monopolar, give different set of information of EA locality during insertion and hence can be useful if both types are utilized together for real time impedance measurements. No such system exists and already developed impedance measurement systems have certain limitations (e.g., they only measure impedance magnitude). There is also no system to-date (to the best of my knowledge) that is using complex impedance measurement for real-time EA localization. It is also important to state that impedance values do not give any information about array tip, however, force sensing can be utilized to get tip information. Therefore, for accurate guided placement of the CI, both impedance measurements and force profiles may be required. Finally, machine learning algorithms are presented that has the potential to be used as prediction/estimation modalities for localization of EA during insertion process.

Chapter 3

Methodology

Interaoperative feedback systems for electrode array localization are limited. Those which are available in the form of diagnostic imaging come with radiation exposure problem and they are not well suited for intera-operative imaging capabilities. Works by [15] and [16] presented in 2.2.3 may provide real time feedback, however, a new technology (integrated sensing) requires plenty of resources to reach clinical acceptability. The experiments performed by [12] and [14] are for perimodiolar electrode arrays (as discussed in 2.3.2.3), only uses impedance magnitude as a sensing element and looking at difference of impedance magnitude difference of electrodes proximal to lateral or modiolar walls.

Recently electrical impedance is used to assess facial nerve proximity during robotic drilling procedure of cochlear implantation [74]. Electrical impedance measurements can also be employed to optimize EA position and eventually reduce insertion trauma. The EA of cochlear implant has number of electrodes which stimulate auditory nerve. Most cochlear implant (CI) systems have built in impedance measurement system and they may employ different methods to measure it. These methods are different based on the electrode pairs across which impedance is measured; monopolar impedance measurements are between individual electrodes and an extracochlear ground electrode, bipolar measurement is between two consecutive electrodes (one as a source and other as a sink), and some CI systems have tripolar measurement which employ varying excitation between three consecutive electrodes as shown in Figure 2.21.Various non-CI systems also have tetrapolar systems to measure impedance which employ four consecutive electrodes. This work focuses on mono and bipolar impedance measurements (both magnitude and phase) of electrodes along with their real and imaginary parts to develop a data driven model for robotic feeder control.

This chapter presents methods and techniques used in this work. These include hardware/software design, their interface and data analysis/machine learning techniques. The hardware consists of impedance meter, actuation system with its controllers, force sensor and electrode array from Oticon Medicals. Impedance meter and actuation system are interfaced to MATLAB via USB port and they are controlled with a Graphical User Interface (GUI). Machine learning pipeline is also integrated in to this system for prediction. The chapter is organized as follows: electrode array used throughout experiments are introduced first, followed by the novel impedance sensing mechanism, actuation system is explained in the third section and machine learning techniques are presented in the fourth section. Preliminary results of electrode array insertion in the phantom models will be presented in the last section [1].

3.1 Electrode Array

Oticon Medical's soft straight EVO® electrode array was used. It is a long (Insertion Length: 25 mm, Active Length: 24 mm), thin (proximal diameter = 0.5 mm; distal diameter = 0.4 mm), flexible array with a smooth silicone surface carrying 20 micro-machined platinum-iridium electrodes as shown in Fig. 3.1. Each electrode length is 0.47 mm and the gap between two consecutive electrodes is 0.73 mm. Electrodes are numbered from E1 to E20 where former is the apical and latter is the basal electrode. Surface area of electrode ranges from 0.46-0.66 mm^2, smaller being at the apex and increasing in size towards base. Each electrode is connected to a wire, and wires from 20 contacts are soldered and glued with dual row male connector (as shown in Fig. 3.2). Electrode contacts are stimulated and controlled through these wires. The soft and flexible silicon carrier makes sure that forces imparted on inner ear are bare minimum. Although the electrode array is straight, however, silicon carrier has shape memory.

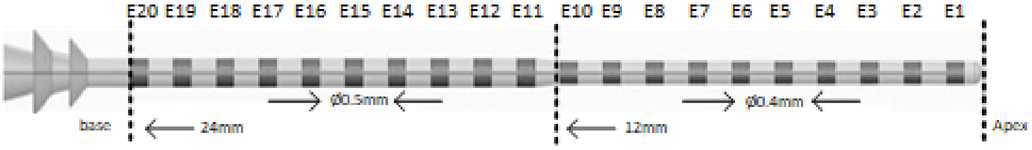

E20 E19 E18 E17 E16 E15 E14 E13 E12 E11 E10 E9 E8 E7 E6 E5 E4 E3 E2 E1
ø0.5mm
ø0.4mm
base
24mm
12mm
Apex

Fig. 3.1 Electrode array with 20 platinum-iridium electrode contacts numbered from E1-E20 from apex to base. (image from www.oticonmedical.com)

A scaled up 2:1 plastic cochlea model is used. The reason for using scaled up model is due small size of the real human cochlea and is hard to perform laboratory experiments on that size. The cochlea model is filled with saline solution with 0.9% concentration as a surrogate to perilymph fluid. According to Suesserman [**?**], the conductivity of perilymp fluid in scala tympani is 1.43 $(\Omega m)^{-1}$ and according to Weast [99] and Wolf [100], 0.9g/100ml NaCl is

[1] Part of this chapter has appeared in Hafeez et al. [98]

required to obtain this conductivity at 20 °C. The conductivity was checked before each experiment with a conductivity meter.

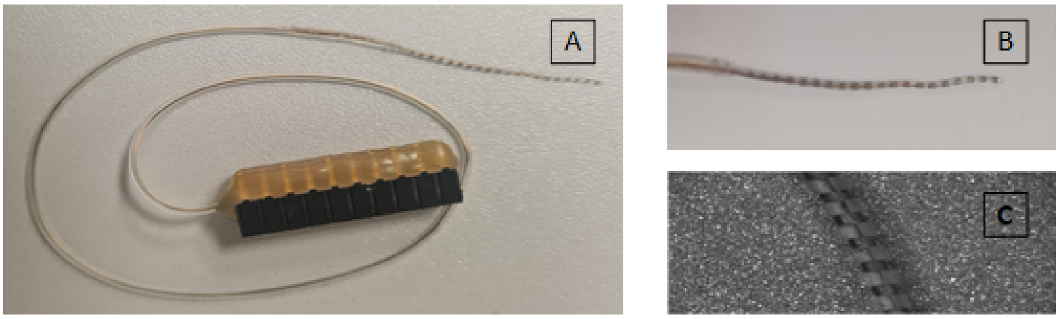


Fig. 3.2 (A) Electrode array with associated wiring (wraped in silicon carrier) soldered and glued to a female connector, (B) close-up version of 20 electrode contacts, (C) part of electrode array under microscope (www.oticonmedical.com)

3.2 Impedance Measurement

3.2.1 Impedance

According to Ohm's law [101], when an electric current (I) flows through a conductive material, voltage (V) across two points of that material is equal to product of current and resistance (R) of that conductor or resistance of the conductor is equal to ratio of voltage and current.

$$V = IR \quad \Rightarrow \quad R = \frac{V}{I} \tag{3.1}$$

This is true when a direct current (DC) is applied between cylindrical shaped conductor such as a wire. When an alternating current (AC) is applied, a generalization of the Ohm's law gives rise to the concept of impedance. Impedance at a given frequency is defined as a ratio of voltage phasor to the current phasor [102]. Electrical impedance, Z, is a complex value and comprised of magnitude (which is the ratio of voltage and current) and phase angle (which is difference of phases of voltage and current) and is often given by 3.2

$$Z\angle\theta = \frac{V\angle\theta_1}{I\angle\theta_2} \tag{3.2}$$

Electrical impedance, Z, of an electric circuit is considered to include two components, resistance R (in Ω) which is the real part and reactance X (in Ω) which is imaginary and is given by 3.3 where j is unit imaginary number. |Z| is impedance magnitude and θ is the phase angle. Geometrically, impedance is a vector composed of resistance and reactance where phase angle θ suggests position of impedance vector Z and depends on resistance and reactance values as shown in fig. 3.3.

$$Z = R + jX = |Z|e^{j\theta} \tag{3.3}$$

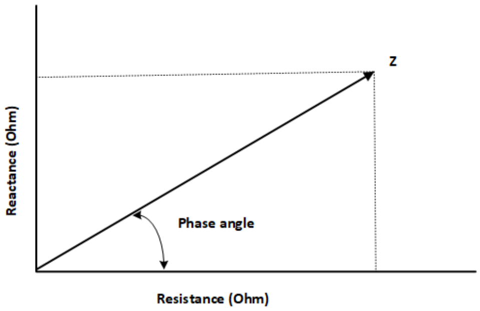


Fig. 3.3 Geometric relationship between impedance (Z), reactance (X), resistance (R) and phase angle

3.2.2 Impedance meter using DAQ device

Impedance meter was designed using National Instrument (NI) Data AcQuisition (DAQ) device [103]. A known resistance is placed in series with Design Under Test (DUT), an unknown system for which impedance measurement is required. An analog output (AO1) of the DAQ device is connected to one end of DUT alongwith the positive end of a differential input (AI1). The negative end of the differential input (AI2) is connected with the junction of DUT and known resistance as well as positive end of another differential input (AI3). Finally, other end of known resistance is connected to negative end of the second differential input (AI4) and differential GND. The analog output is required to apply an analog input to the series circuit, whereas two differential inputs are used for measuring voltage drops across DUT and known resistance. The current through the circuit is calculated using ohm's law with known resistance and current through it. Impedance magnitude across DUT can then be measured by the current in the circuit and the voltage drop across it. Impedance phase is measured by taking the difference of voltage and current phases.

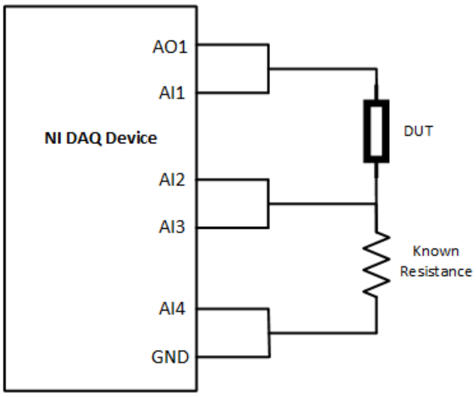


Fig. 3.4 Basic design concept of impedance meter with DAQ device

3.2.3 DAQ Devices Specification

Two DAQ devices have been used in this work; one to develop impedance meter and other to control the multiplexers. Their specifications are as follows:

3.2.3.1 NI DAQ 6211

There are two types of Input/Output ports; analog and digital.

Analog Input: There are 8 differential or 16 single ended channels (it is upto the user how to use them) with analog-to-digital conversion (ADC) resolution of 16 bits. Single channel maximum/multichannel maximum (aggregate) sampling rate is 250 kS/s. Its timing accuracy is 50 ppm (parts/million) with timing resolution of 50 ns. Input range could be $\pm 0.2V$, $\pm 1V$, $\pm 5V$, $\pm 10V$ with maximum working voltage for analog inputs is $\pm 10.4V$ of AI GND. All ports marked with AI in Fig. **??** (B) are analog input ports. These ports are used to read voltage levels.

Analog Output: There are two analog output channels with Digital-to-Analog Conversion (DAC) resolution of 16 bits. 250 kS/s is maximum update rate for each channel. Time accuracy and resolution are 50 ppm of sample rate and 50 ns respectively. There is $\pm 10V$ output range with output current drive of $\pm 0.2mA$. There are multiple analog output waveform modes (periodic and non-periodic). All ports marked with AO in Fig. 3.5 (B) are analog output ports. These ports are used to generate voltage signals.

Digital I/O The device has 4 digital input and output channels each with pull down resistor of 47 $k\Omega \pm 1\%$. They have functionality of static digital input/output and timing input/output. Digital input channels are marked as PFI<0-3>PO<0-3> whereas digital output channels are marked as PFI<4-7>P1<0-3> as in Fig. 3.5 (B).

It also has 2 counters/timers with the resolution of 32 bits. USB signal stream and programmed I/O are the two ways to transfer data using this device.

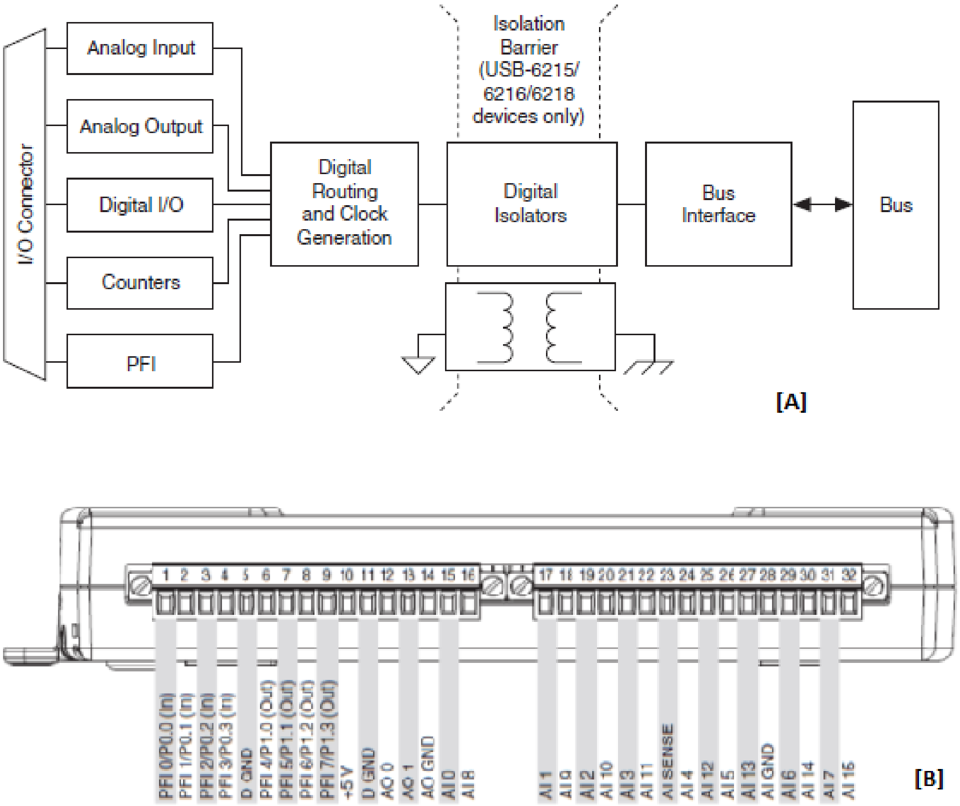


Fig. 3.5 (A) NI DAQ 6211 Device function block diagram, (B) Pin configuration

3.2.3.2 NI DAQ 6009

This device has 4 differential or 8 single ended analog input channels and 2 analog output channels with sampling rate of 48 kS/s. There are 8 digital output channels and 4 digital input channels as shown in Fig. 3.6(B).

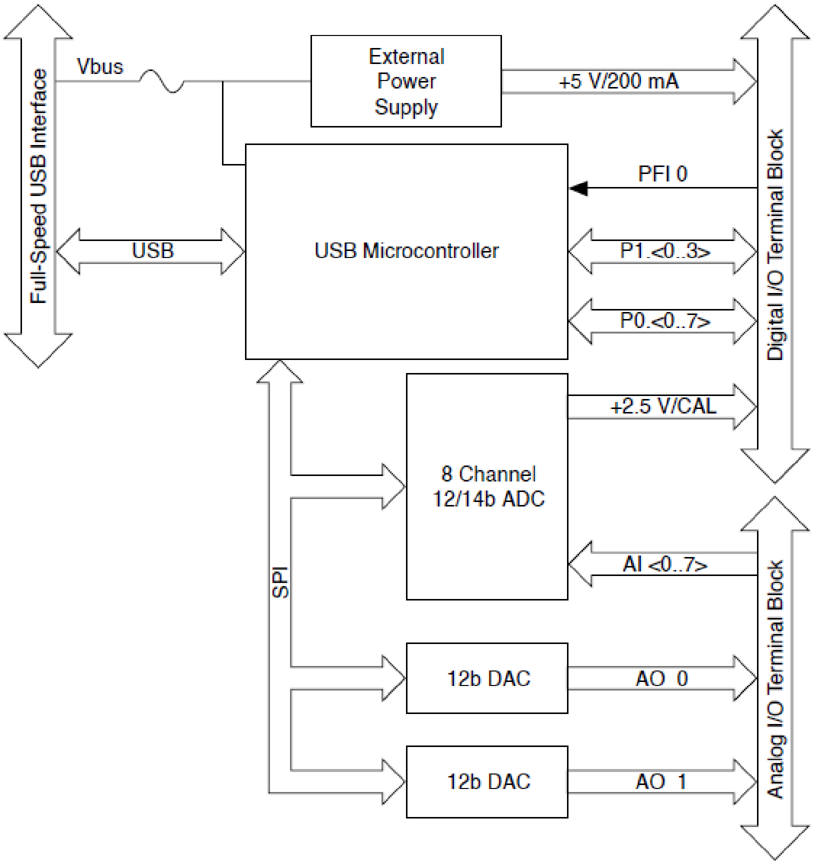


Fig. 3.6 NI DAQ 6009 Device function block diagram with I/O pin detail

3.2.4 Multiplexer

As there are multiple electrode contacts/channels on electrode array, impedance recordings are required to be done sequentially and multiplexers are used for the same purpose. CD74HC4051 is an analog multiplexer/demultiplexer 3.7. Its a 16-pin integrated circuit device. It has 8 I/O channels (A0-A7) with 3 select lines (S0, S1, S2). Each combination of select lines directs a particular I/O data to the output for multiplexer or vice versa for demultiplexer. There is an active low triggered enable input to activate or deactivate the device. V_{cc} is power input level for outgoing channel whereas V_{EE} is power input level for incoming channel. There is also a GND port as shown in Fig. 3.7.

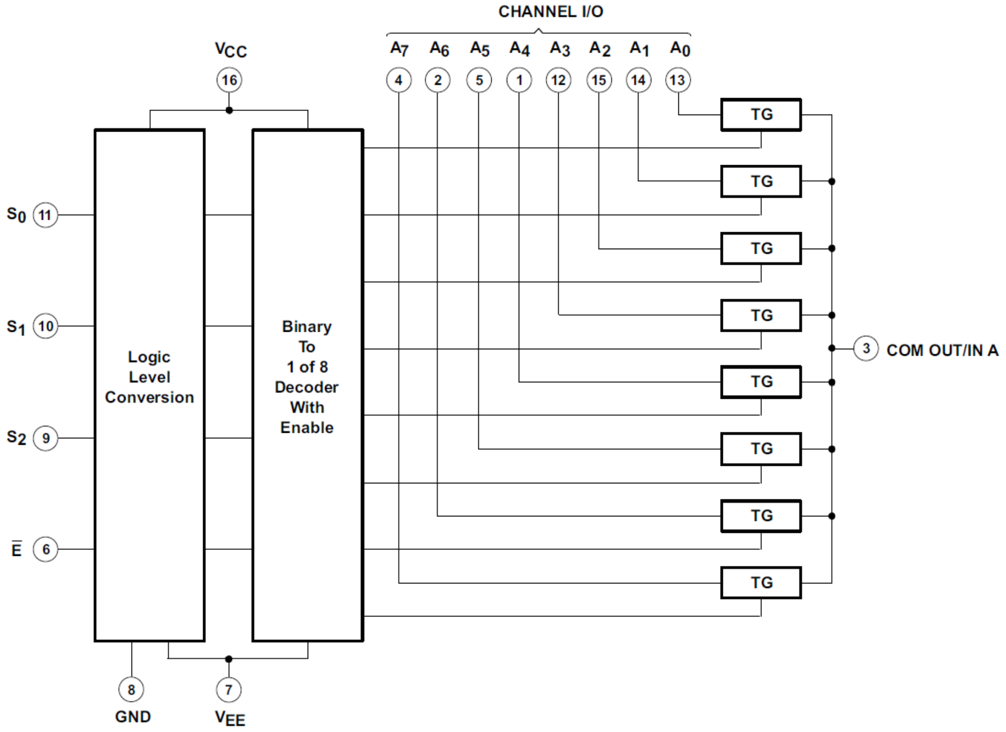


Fig. 3.7 Multiplexer CD74HC4051 functional block diagram with pin numbers

3.2.5 Impedance Meter

Impedance meter with measurement switching capabilities was developed using National Instrument (NI) (Austin, TX USA) data acquisition (DAQ) devices (DAQ6211 and DAQ6009) and a simple voltage divider circuit. The system's setup configuration chain is presented in Fig. 3.8. All analog and digital signals were generated and read by the DAQ devices and these devices were controlled by custom software written in MATLAB ® (Fig. 3.9). The voltage divider circuit was constructed with a known resistance (R) of 3.3Ω and electrode pair (EP) as its series component. A sinusoidal voltage signal V_{in} of 1V (p-p) with 1 kHz frequency was applied to the circuit and voltage across known resistance (V_R) and EP (V_{EP}) were measured by the DAQ device. Voltage input signal to the circuit V_{in} was generated by DAQ's voltage output port A_o and measured voltages (V_R and V_{EP}) were read by DAQ's input ports A_{i1} and A_{i2} respectively. Current I through the circuit can be calculated with (3.4)

$$I = \frac{V_R}{R} \tag{3.4}$$

Impedance is a complex quantity that possesses both magnitude and phase and can be given as (3.5). Impedance magnitude is the opposition to the current when an AC voltage is applied. Bipolar impedance magnitude $|Z|$ of selected electrode pair is just voltage across electrode pair divided by the current through it as in (3.6)

$$Z = |Z|e^{j\theta} \tag{3.5}$$

$$|Z| = \frac{V_{EP}}{I} \tag{3.6}$$

The impedance phase ($arg(Z)$ or θ) gives information about the type and amount of reactance in overall impedance. It is calculated by taking the phase difference between voltage and current as in (3.7)

$$\angle\theta = \angle\theta_{V_{EP}} - \angle\theta_I \tag{3.7}$$

The resistive (R) and capacitive reactance (X) parts of the impedance can be calculated as

$$R = |Z|cos\theta \tag{3.8}$$

$$X = |Z|sin\theta \tag{3.9}$$

Two 74HC4051 8x1 multiplexers were used to sequentially switch from one pair to another and measure bipolar complex impedance between different electrode pairs. Each multiplexer has 8 data inputs D1-D8, 3 select lines C1-C3, and enable input EN and an output Q. Electrode contacts were connected to data input pins of multiplexers such that an electrode contact was selected by each multiplexer to make a pair. Eight electrode contacts (E1, E3, ...E15) are connected to multiplexer1 inputs and electrode contacts (E2, E4, ...E16) are connected to multiplexer2 inputs. Electrode pairs are organized as shown in table 3.1. Electrode pairs are chosen to acquire sensing information as local as possible. The last 4 electrodes of the array are not included as they do not touch the ST curve during the insertion and have minimal effect on damaging the ST membrane. Multiplexer1 and multiplexer2 were controlled by 3 digital output signals S1-S3 generated by DAQ6009 device digital output ports ($D_0 - D_2$) and S4-S6 generated by DAQ6011 digital output ports ($D_0 - D_2$) , respectively. These select lines vary from 000-111 as digital signals to select appropriate contact to form an electrode pair for impedance measurement. When an electrode contact was selected, a connection was established between the contact and the output Q. The enable signal EN was connected to +5V Vcc signal from the DAQ6009 device to activate the multiplexers. When multiplexers were enabled and an EP was selected, signals were applied and read from series circuit to measure outputs |Z|, θ, R and X according to the above equations.

Electrode Pairs	Electrode Contacts
EP1	E1-E2
EP2	E3-E4
EP3	E5-E6
EP4	E7-E8
EP5	E9-E10
EP6	E11-E12
EP7	E13-E14
EP8	E15-E16

Table 3.1 Electrode pairs organization

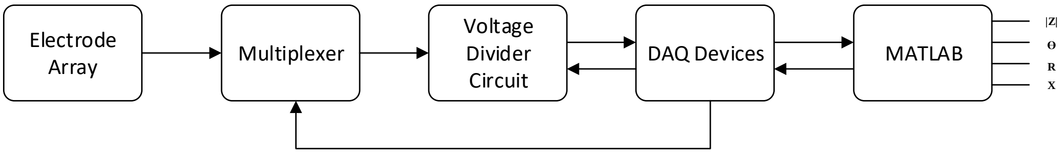


Fig. 3.8 Setup configuration chain

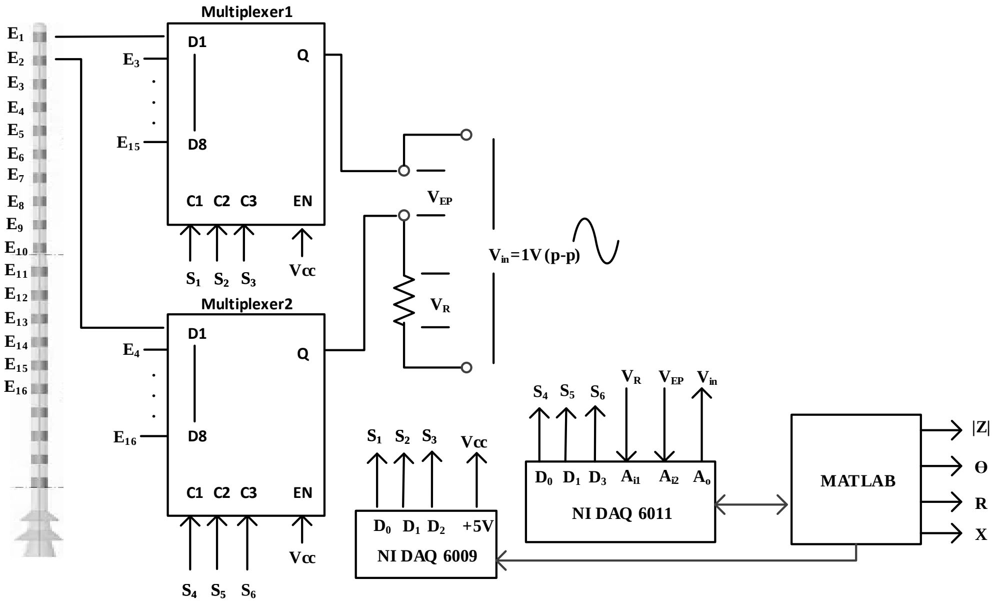


Fig. 3.9 Impedance meter block diagram with multiplexing

3.3 Actuation System

Two types of actuators are used in the development of the actuation system. Physik Instrumente (PI) M-404.4PD is a precise, inexpensive cost-optimized linear stage with ballscrew and rotary encoder as an integrated sensor. The width of 80 mm and 100 mm travel range with active drive DC motor. It has an operative voltage of 24 V. and motor power of 26 Watts. Its design resolution and minimum incremental motion is 0.25 μm and maximum velocity of 50 mm/s. The actuator is shown in Fig. 3.10 (B). There will be 2 such actuators used in the proposed system.

The other actuator is PI's M-061.1PD precise rotation stage (as shown in Fig. 3.10) (A) with unlimited travel range (>360°) with design resolution of 17.5 µrad (°). The maximum velocity it supports is 90°/s. It has a gear ratio of 90:1. The max torque it can handle is $\pm 6Nm$ at any axis. It has an active drive dc motor as in the linear stage. It has similar operative voltage and motor power as in the linear stage described above.

The controller used to control above mentioned PI actuators is C-863 (as in Fig. 3.10(C)). Each actuator is connected to this controller in a daisy chain network setting. The function of the controller is to control the DC motor that drives the actuator either linearly or rotation. It supports communication interfaces such as USB, RS-232 and D-sub 9 (m). Although it has user software (PIMikroMove), however, it also provides application programming interfaces (APIs) in most common languages C/C++, C#, MATLAB, Python and LabView.

The overall actuation is shown in Fig. 3.12 (A). One linear stage M-404.4PD is placed horizontally and screwed with the flat wooden board. Another similar linear stage is attached vertically to the horizontal stage. The rotational stage M-061.1PD is attached to the vertical linear stage. The complete system is connected to the PC via C-863 daisy chain controller network.

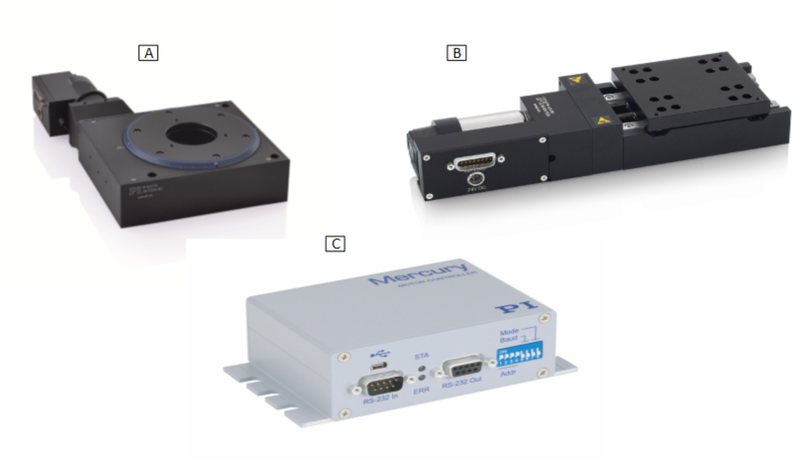


Fig. 3.10 PI Actuators and controller

3.4 Experiment and Analysis Pipeline

An overall sketch of the experimental and analysis pipeline used in this work is presented in Fig. 3.11. EA insertion experiments are performed on 2:1 scaled-up ST model filled with saline solution. EA is inserted using a 3-DoF actuation system connected to the PC via a controller. Complex impedance measurements of electrode pairs are collected using a custom impedance meter connected to the PC. A custom GUI developed in MATLAB ® is running on the PC and controls actuation and impedance meter. The data collected during the experiments are then used to train and test the machine/deep learning models to predict insertion path and depth in real time.

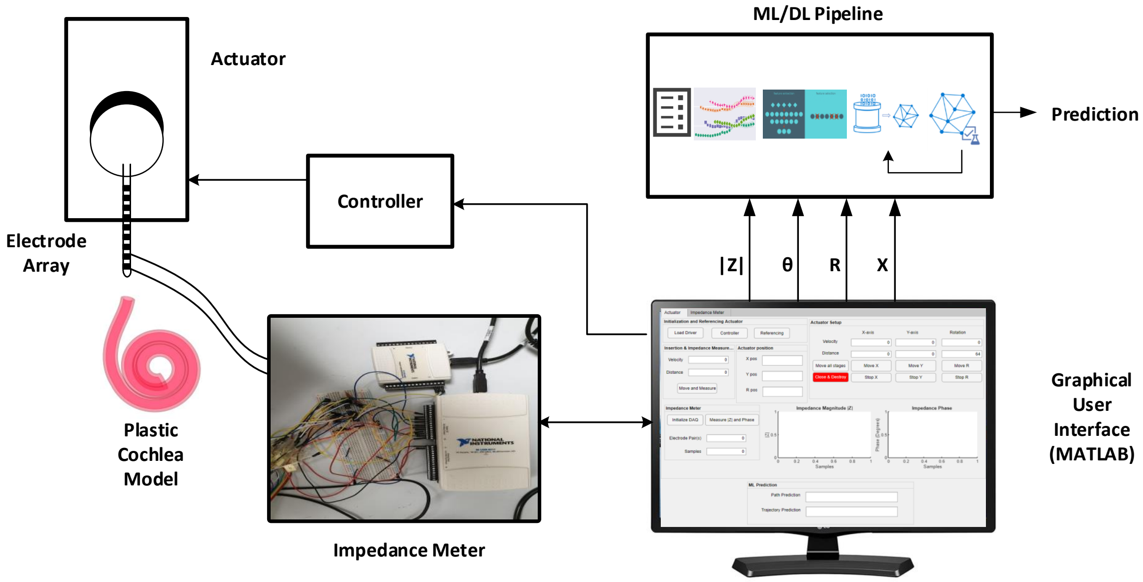


Fig. 3.11 Overall experimental and analysis pipeline

3.4.1 Experimental Setup

The experimental setup for EA insertion is shown in Fig. 3.12. A 2D plastic cochlear model glued on a support was placed inside a glass filled with saline. The electrode array was placed on a holder attached to the rotational actuation of the actuation system. Electrodes of the array are connected to the impedance meter via a connector glued to the EA feeder support as shown in Fig. 3.12 A. Impedance meter setup is shown in Fig.3.12 B, where we have a breadboard housing the voltage divider circuit, multiplexers and connections with the electrodes. There are two DAQ devices for analog/digital I/O signals to the circuit and multiplexers and are connected to the PC via USB cables. In Fig. 3.12 C and D, we have a portion of the EA under the microscope and zoomed version of ST model during the EA insertion process.

Fig. 3.13 shows Graphical User Interface (GUI) to integrate and control the system. There are three main sections in the GUI, 1) the actuation ,2) impedance meter and 3) machine learning prediction. On the top left, we have a section where actuation system driver is loaded and the controller is initialized. This section also reference all the actuators as well before we intend to use them for EA insertion. In top right section, we have all the controls where either we can adjust the actuators individually or all at once by choosing the custom velocity and distance to move. We can also stop moving actuators by Stop buttons. Actuator position section displays final position of each actuator after the move; mm for linear guide and angle for rotation actuator. The DAQ devices are initialized from the impedance meter section and takes in number of EPs and samples we want to measure. Insertion and Impedance Measurement section is used for EA insertion at a certain velocity and distance, and impedance measurements during the insertion process. The graphical section shows plots of impedance magnitude and phase during the insertion process in log scale due to potentially very high impedance values. Finally, ML prediction section displays real time predictions of our insertion path (straight or curved) and prediction trajectory (medial, middle or lateral).

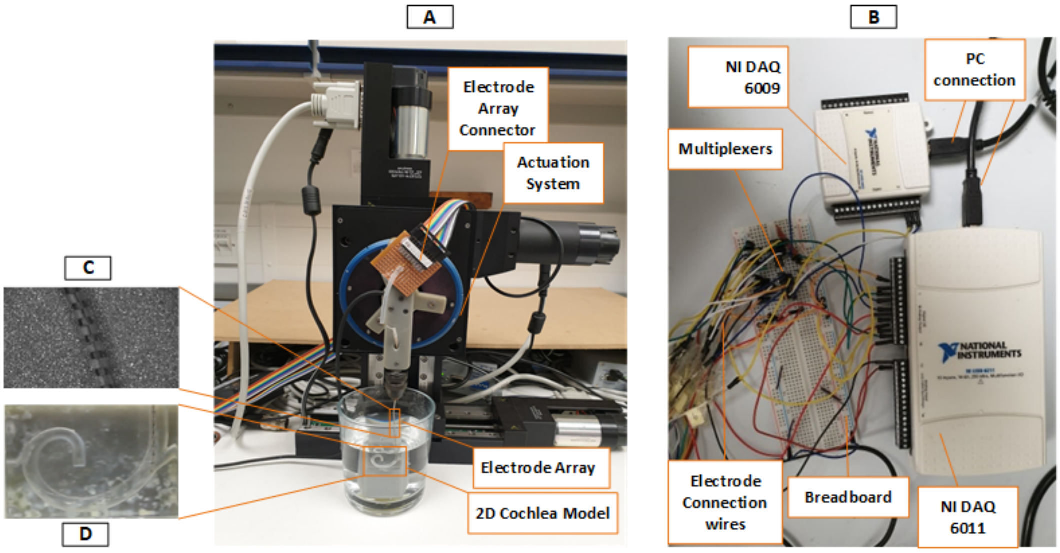
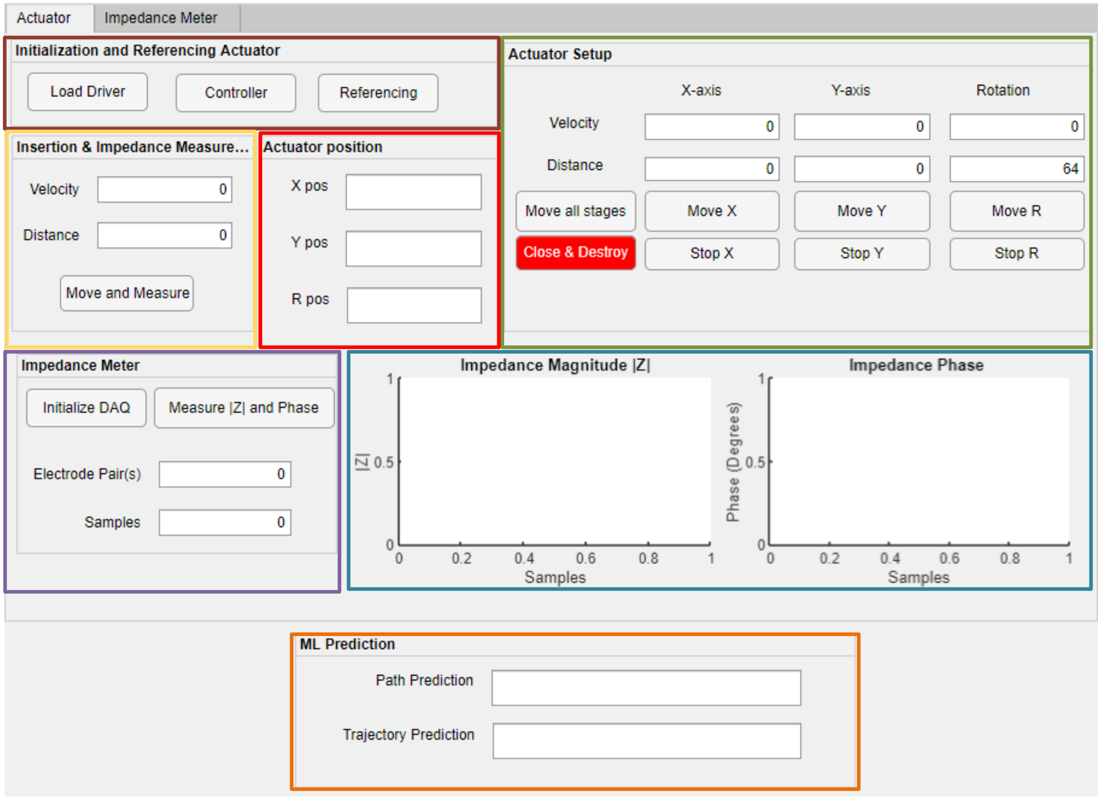


Fig. 3.12 Experimental setup comprising of actuator, impedance meter, electrode array and plastic ST model.



Insertion & Impedance Measure...

Velocity 0

Distance 0

Actuator position

X pos

Y pos

R pos

Actuator Setup

	X-axis	Y-axis	Rotation
Velocity	0	0	0
Distance	0	0	64

Impedance Meter

Electrode Pair(s) 0

Samples 0

ML Prediction

Path Prediction

Trajectory Prediction

Fig. 3.13 Custom Graphical User Interface (GUI) created in MATLAB for system integration

3.4.2 Preliminary Results

The impedance model is based on Tykocinski [72], and consists of three components, 1) resistance between electrodes due to bulk medium (saline), 2) polarization resistance and 3) capacitive reactance due to electrode-electrolyte interface. Components (2) and (3) combined are called polarization impedance and can be modeled as a parallel circuit with (1) in series with them. According to this model, we not only need to measure impedance magnitude but also the phase, and impedance resistive and reactive components to fully comprehend the model and use these features for prediction. Since we are measuring bipolar impedance, the total impedance includes resistive and reactive components as shown in Fig. 3.14. R_3 is the bulk resistance between the 2 electrodes. On either side of it, we have polarization circuits for each electrode due to electrochemical reaction.

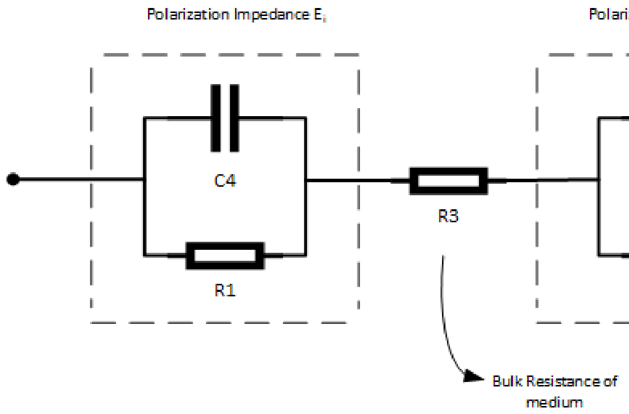

Polarization Impedance E_i
Polarization Impedance E_{i+1}
C4
R1
R3
C2
R2
Bulk Resistance of medium

Fig. 3.14 Electrode/Electrolyte Impedance equivalent circuit [Electrode array in saline solution]

Experiments were performed by inserting the electrode array at a speed of 0.05 mm/s for 25 mm depth and bipolar impedance magnitude and phase measurements of each pair were recorded sequentially during the insertion. It takes 500 seconds to complete the insertion. Impedance meter samples a pair in 1.5 sec, so 41 samples of each electrode pair were taken during the insertion process and analysed offline using Python 3.6. Bipolar complex impedance of apical 16 electrodes in pairs of 8 were recorded sequentially during the insertion of the electrode array in a plastic ST model. The reason for using nearest two electrodes is to acquire small local sensing information. Fig. 3.15 shows four instances during insertion; A) 4 electrodes inserted, B) 8 electrodes inserted, C) 16 electrodes inserted, D) 20 electrodes inserted. The arrows show the location of the tip of EA. Fig. 3.16 shows impedance magnitude ($|Z|$), phase (θ), real (R) and imaginary (X_c) parts during insertion. |Z|, R and X_c have their units in Ω and θ is in degree (°) units. The recordings not only depicts the changes in values when EA is closely placed to the wall (compared to when it is not) but also shows changes when a specific electrode is rubbing (exerted pressure/force) along the wall. It is important to note that when an electrode pair is not in the saline solution (not entered ST model), it gave a high open-circuit impedance value and these values are ignored and not included in the graphs.

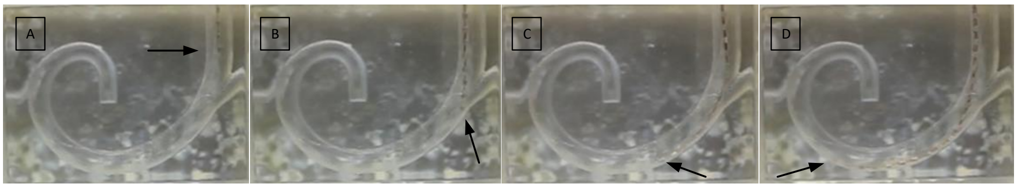


Fig. 3.15 Electrode array insertion at different instances inside ST phantom

Comparing Fig. 3.15 and Fig. 3.16, EP1 (E1-E2) and EP2 (E3-E3) were inside the ST model and away from the wall at instance A. At instance B, E1 (most apical electrode) was starting to touch the lateral wall for the first time and all measurements of EP1 were starting to change. $|Z|$ from 2.5 kΩ to 3.2 kΩ, θ is getting less negative from -36° to -33°, impedance real part (resistance R) increased from 2 kΩ to 2.7 kΩ, and impedance imaginary part (reactance X_c) from 1.5 kΩ to 1.8 kΩ. In the same way, after 4 samples (between instance B and C) second pair EP2 came in to contact with the wall and its electrical properties starting to change in the same way. A significant change in measured values were also observed in subsequent electrode pairs EP3 (E5-E6), EP4 (E7-E8), EP5 (E9-E10) just before instance C till instance D, however, there is no significant change in complex impedance of EP6 (E11-E12), EP7 (E13-E14) and EP8 (E15-E16) as they did not come in to contact with the wall during the insertion process. Fig. 3.17 shows percent change of values during whole insertion (when a particular pair is in saline filled cochlear model) in $|Z|$, θ, R, and X_c of EP1 (apical), EP4 (middle) and EP7 (basal) which clearly shows EP1 has more pronounced change than EP4 due to more contact pressure on it whereas EP7 has no significant change as this pair did not come in to contact with the outer wall.

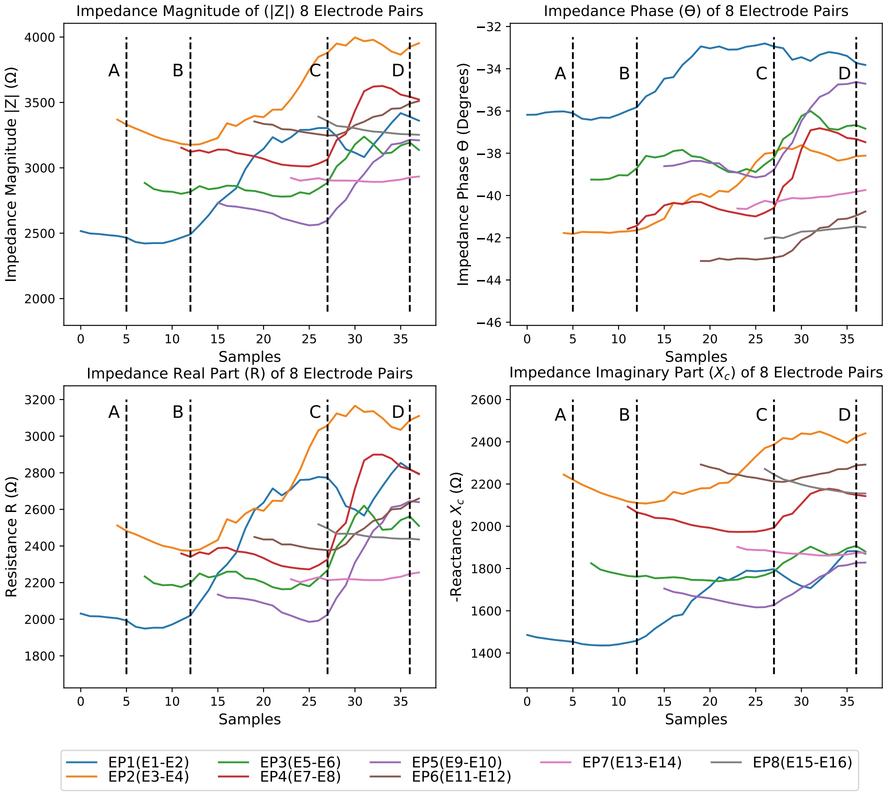


Fig. 3.16 Impedance magnitude, phase, real and imaginary parts of 8 electrode pairs during insertion

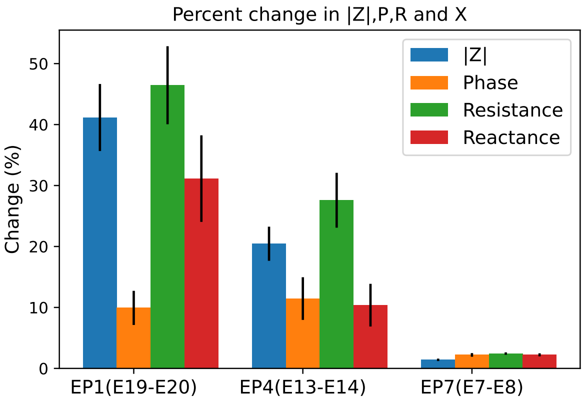


Fig. 3.17 Percent change (from minimum to maximum values) with standard deviation in impedance magnitude |Z|, phase θ, resistance R and capacitive reactance X_c of electrode pair 1, 4 and 7

An increase in $|Z|$ due to wall contact suggested that there is more resistance path between the electrodes. The more negative Phase θ implies that impedance is more resistive than capacitive and this phenomenon can also be seen in R and X_c graphs where the change in R is more pronounced than in reactance. Also, an increase in reactance means a decrease in polarization capacitance according to relation $X_c = 1/2\pi fC$.

These results are mainly due to three reasons; 1) disturbance in the chemical reaction at the electrode-electrolyte interface due to wall contact pressure/force, 2) when array gets closer to the wall, there would be a decrease in reacting electrolyte with electrodes and between electrodes, 3) impedance of the plastic material is higher than saline.

3.5 Conclusion

According to current research, there is enough evidence of a relation between EA placement procedure with hearing outcomes and bipolar complex impedance measurements may be used as a sensing mechanism for localization of EA during cochlear implant surgery. Impedance change due to electrode proximity to different material and application of pressure/force can have application in other electrode implants. These results may be used as feedback control for the actuation system for EA insertion in implantation to manoeuvre it precisely.

Chapter 4

Insertion Trajectory Prediction

One of the primary focus of minimally invasive cochlear implantation is trauma-less electrode array insertion. Therefore, electrode array placement accuracy during cochlear implant surgery is vital to ensure efficacy, safety and functional outcomes. Although imaging based trajectory planning has been proposed [104], [105], however, it is an expensive approach as it requires high resolution CT images, it comes with exposure to harmful radiations and electrodes may also deflect from the planned trajectory due to electrode design and tissue properties. Skinner et al. [106] suggested that electrode array insertion in scala tympani has better hearing outcomes, however, according to [107] nearly 25% fails proper insertion into scala tympani. Therefore, it is of paramount importance to insert the array in to tympanic duct properly and without damage to the internal structure. This process posses several challenges;the size of cochlea/ST, position of facial nerve with respect to cochlea opening, the insertion axis and associated trajectory and inconsistency of mental representation of insertion axis among surgeons. To overcome some of the issues and optimize the insertion process, automated systems along with a sensing system will help reduce variability.

The final placement of electrode array into ST mainly depends on the insertion angle or trajectory, however, there is limited intra-operated information regarding the optimum trajectory. This chapter proposes data driven approach to the trajectory prediction of electrode array during insertion. Bipolar impedance data of multiple electrode pairs were recorded, features are extracted from pre-processed data, train and tested on machine learning algorithms for trajectory prediction (medial, middle or lateral).

The chapter is organized as follows: 1) The acquisition process of complex impedance of electrodes during the EA insertion is presented 2) Then offline prediction workflow using machine learning algorithms is presentation, including data presentation and visualisation, preprocessing and model evaluation, 4) finally the results, discussion and conclusion[1].

4.1 Data Collection

The electrode array is inserted from three different angles (medial, middle, and lateral) where the middle position was the center of the plastic model and medial and right at 0.5° to either side from the center as shown in (Fig. 4.1). Initially electrode array was out of saline (and out of ST model) except first pair EP1 (E1-E2). A high impedance measurement was recorded if a pair was out of saline as an open circuit value because there is no conductive material (saline) around electrodes. The electrode array is inserted into the ST model for 25 mm at a speed of 0.08 mm/s by moving the vertical linear actuator (EA is placed on it with a holder). Recordings of 8 electrode pairs (EP1:E1-E2, EP2:E3-E4, … EP8:E15-E16) were taken with a sampling time of 1.5s. Therefore, each pair has 25 samples during complete insertion (1 sample/pair/mm).

Electrode array was inserted for 25mm in ST model 137 times from three different directions; 41 times each from medial and lateral directions and 55 times from the middle of ST model. For each insertion, two features ($|Z|$, θ) of an electrode pair were recorded sequentially during insertion and impedance real and imaginary parts were calculated from its magnitude and phase later after the experiments. The reason for including impedance's real and imaginary parts in the analysis due to their non-linear relation with the impedance magnitude and phase. During one complete insertion, each electrode pair records a vector $\mathbf{P}^{(i)} = [\mathbf{Z}^{(i)}, \Theta^{(i)}, \mathbf{R}^{(i)}, \mathbf{X}^{(i)}]$ containing m samples of impedence magnitude $\mathbf{Z}^{(i)} = [|Z|_1^{(i)}, |Z|_2^{(i)}, \ldots, |Z|_m^{(i)}]$, phase $\Theta^{(i)} = [\theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_m^{(i)}]$, and both real and imaginary parts of the impedence given by $\mathbf{R}^{(i)} = [R_1^{(i)}, R_2^{(i)}, \ldots, R_m^{(i)}]$ and $\mathbf{X}^{(i)} = [X_1^{(i)}, X_2^{(i)}, \ldots, X_m^{(i)}]$ respectively. Here $m = 25 - 2(i-1)$ for all $i \in \{1, 2 \ldots, 8\}$. The reason for decrease in samples in subsequent pairs was due to discarding of high impedance measurements when a pair was out of saline solution during the insertion process.

[1] Part of this chapter has appeared in Hafeez et al. [108]

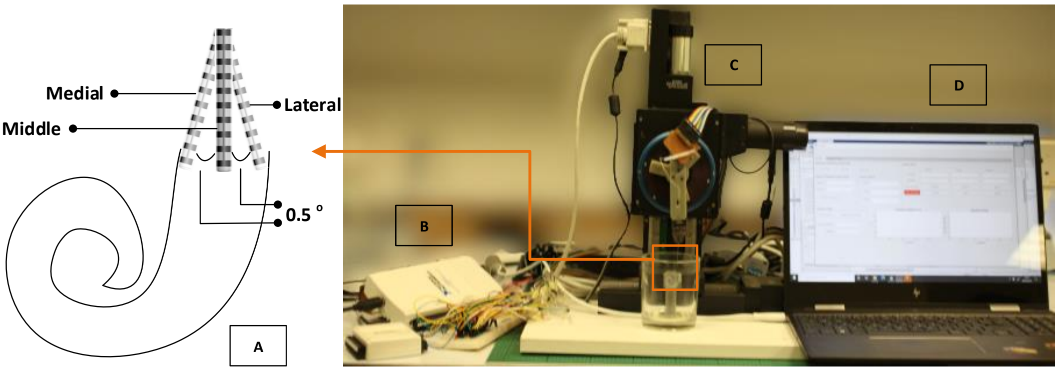


Fig. 4.1 Experimental Setup., (A) Insertion from three different directions (medial, middle and lateral) of ST model, (B) Impedance meter using 2 DAQ devices (6011 and 6009) and breadboard to house circuit, multiplexers and their interconnections with electrode contacts via wires from EA,(C) Actuation system with electrode array feeder, ST plastic model and connectivity circuit board of EA, (D) Laptop with MATLAB GUI to control both impedance meter and actuation system.

4.2 Offline Prediction Workflow

The workflow for this experimental study is given in Fig. (4.2). Starting off with the problem definition i.e., classification of three different classes of insertion trajectory using machine learning algorithms. Dataset assembling and visualization is the next step to set up data in a certain manner and visualize it for analysis. The dataset compromises of electrical features of 8 electrode pairs collected during insertion experiments in the lab. Preprocessing is one of the important steps [109] which includes dumping any high impedance value or averaging out any high impedance/short circuit value for our specific problem. It also involves the normalization/standardization of the data with different techniques. There are two options before dividing the dataset for training and testing a machine learning model; either using the normalized raw data or extract the features from the normalized time series data and select the most discriminant features. In this chapter we are using most common method by extracting the time/frequence domain features from the normalized raw time series data. In model selection and training, we choose machine learning algorithms that best suit for a specific problem and trained on the dataset. We can either split the dataset into different ratio train and test sets, or train the model with k-fold cross validation to avoid overfitting. In our case we choose 5-fold cross validation and we choose *k*NN/SVM/ANN machine learning models to be trained and tested for offline trajectory classification. While doing the model validation, model's hyperparameters are tuned to get the best performance. Once hyperparamaters are tuned for all models, their accuracies are compared, and an optimal model is selected for further validation and utilization as an online predictor.

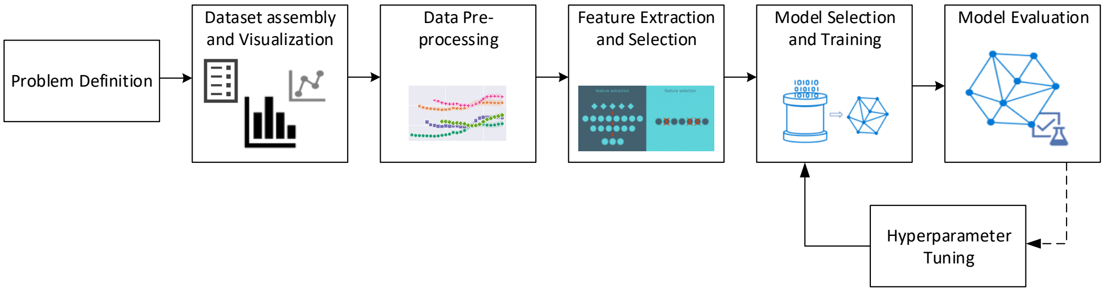

Problem Definition
Dataset assembly and Visualization
Data Pre-processing
Feature Extraction and Selection
Model Selection and Training
Model Evaluation
Hyperparameter Tuning

Fig. 4.2 Supervised learning (Classification) workflow

4.2.1 Dataset Assembly and Visualization

Figure (4.3) shows line plots of 8 electrode pairs covering 4 features ($|Z|, \theta, R, X$) and three insertions directions (left, middle, right). These plots are average of the respective class samples with their standard deviation. Each time step corresponds to 1 mm of insertion into the plastic ST model. As can be seen in the Fig. 4.3, the features are represented as the rows and trajectories are represented as the columns. Electrode pairs (1-8) are represented as different markers where 1 is the most apical and 8 the basal as shown on the right side of Fig. 4.3.

When EA is inserted from the middle of the ST model's diameter, it does not touch any wall and remains in the middle of the structure until it touches the lateral wall after 8-10 mm insertion. At that time only the electrode tip touches the lateral wall and it bounces on and off until it rests along the lateral wall and first electrode pair starts sliding along it. After that, it slides along the lateral wall for the rest of the insertion. It can be seen an increase in impedance magnitude as the respect electrode pair touches the lateral wall and slides along with it due to force exerted on EA and proximity of electrodes to the plastic which is a high impedance material. The impedance phase is becoming less negative during this period which exhibits less capacitive behavior. It may be due to disturbance in the electrochemical reaction between electrode and saline solution when it hits the lateral wall and slides along with it. For the left direction, EA slides along the modiolar wall and touches the lateral wall when the wall starts to curve. On the other hand, EA slides along the lateral wall during the right direction insertion and gets an impact when curving along the wall. Data visualization was carried out using Python's seaborn library [110] after converting the dataset in to a pandas' dataframe.

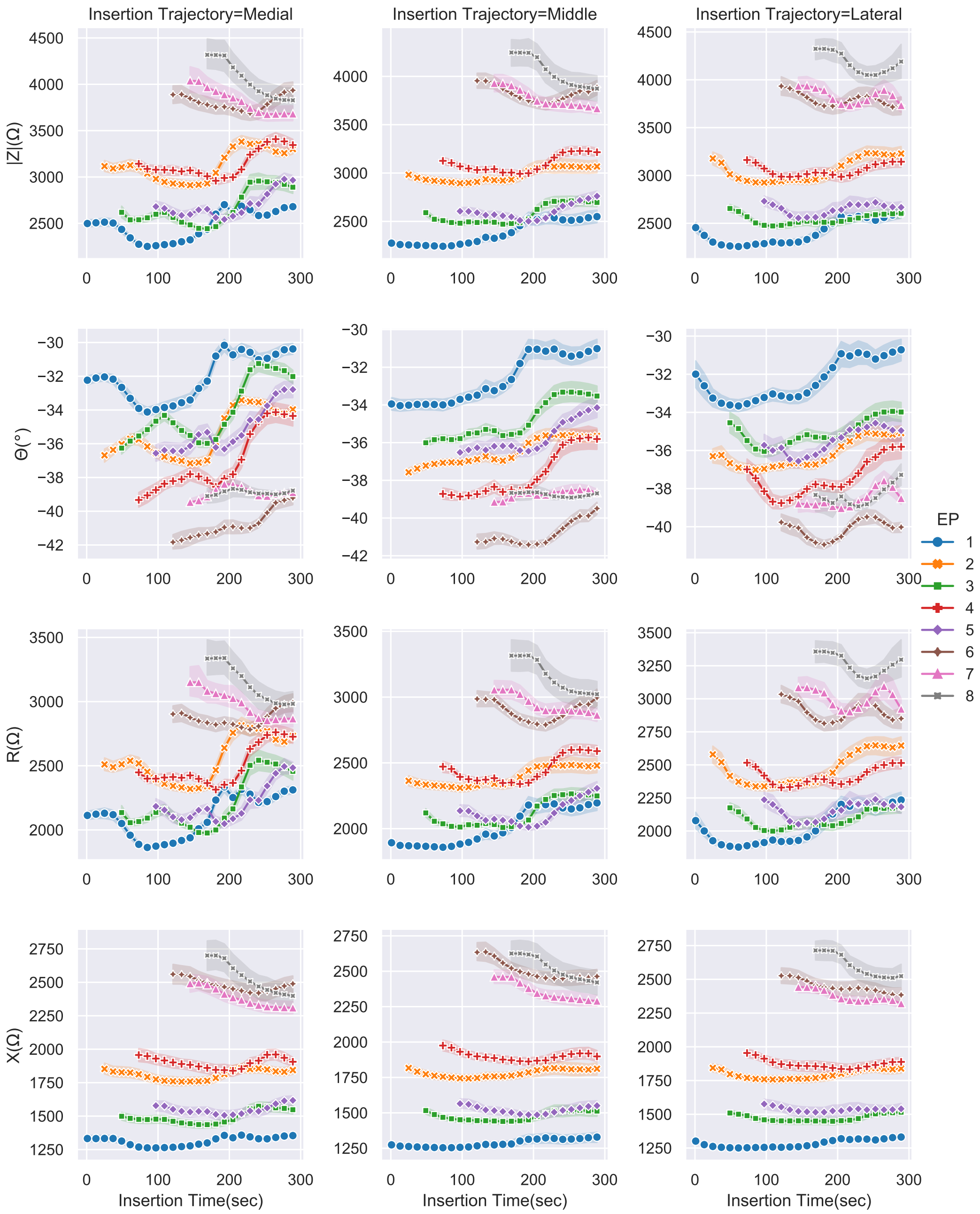


Fig. 4.3 Complex impedance of eight electrode pairs. Rows in the overall figure represent features (From top to bottom: Impedance magnitude (Z(Ω)), phase (Θ(°)), resistance (R(Ω)), reactance (X(Ω)).Columns represent insertion trajectory (From left to right: Medial-Middle-Lateral). High Impedance values are ignored in respective electrode pair's plot.

4.2.2 Preprocessing

This chapter is focused on supervised classification problem and it is understood that preprocessing step do tend to effect the classifier's predictive behaviour [111]. Therefore, preprocessing stage is of paramount importance in the machine learning pipeline. The preprocessing stage may consists of removing NaN/infinite/null values from dataset, handling missing values, dataset balancing in terms of number of classes, normalization/scaling/standardization of input data before feature extraction etc. For our dataset, there was a need to look into high impedance values due to open circuit and very low impedance values due to short circuit between measuring electrodes. There was no short circuit values, however, open circuit (high impedance) values were removed from the EP2-EP8 recordings. These high impedance recordings were due to their presence outside of saline solution during the insertion process. If there is no conductive material around electrodes, high impedance value is recorded. We have not noticed any other data that needs to be removed or imputed. The dataset is almost balanced with 3 classes (medial, middle and lateral) representing 41,55 and 41 classes. Therefore, no dataset balancing was required. We, however, did standardized our input data such that the mean of the values is 0 and standard deviation is 1 (as in 4.1) of each training/testing example.

$$z = \frac{x_i - \mu}{\sigma} \tag{4.1}$$

4.2.3 Feature Extraction

TSFRESH (Time Series FeatuRe Extraction on the basis of Scalable Hypothesis test) [112] has been used for large scale feature extraction and important features are selected for training the model. The reason for using this technique is to avoid hand-crafted feature extraction. This technique has the ability to provide vast array of features for further selection. TSFRESH extracts 794 time series features that belong to time domain, frequency domain and statistical features. Limited list of extracted features are given in table 4.1. Full list of features can be found at [113]. With 63 time series characterization methods (stated in [113]), it computes a total of 794 time series features and they are selected on basis of automatically configured hypothesis tests. However, we have not used its selection method for feature selection as there was no way to select fixed number of features. It will be discussed in next section.

This method was opted to compare the results with normalized raw data results. It was also used to see which electrical measure (|Z|, θ, R, X_c) has more importance in the prediction. Standardized data was converted in to a Pandas dataframe as required by the TSFRESH

Table 4.1 Limited List of Extracted Features using TSFRESH python package

Time Domain	Frequency Domain
Mean	Power Spectral Density
Standard Deviation	Fast Fourier Transform (FFT) Coefficients
Skewness and Kurtosis	Continuous Wavelet Transform (CWT) coefficients
Autocorrelation	FFT aggregated
Autoregressor Coefficient	Number of CWT peaks

package. Four columns of the dataframe are time series data (|Z|, θ, R, X_c), other columns include class labels, timestamps of time series data, sample number and respective electrode pair. Classification is performed with either uni-variate (single EA data) or multivariate (more than one EA data) time series data. After the features are extracted, we have to impute the features dataframe for any non-finite values(i.e. NaN/inf) and then apply the feature selection method to select most relevant features. The imputation is carried out by the mean strategy i.e., replacing the non-infinite value with the mean of the rest of the vector values.

4.2.4 Feature Selection

Feature selection or dimensionality reduction is mainly carried out to reduce the computational complexity but it also improves the performance of the machine learning algorithms [114]. There are four types of feature selection methods, 1) filter model, 2) wrapper model, 3) hybrid model and 4) embedded model. Filter based method has been chosen for this study due to lower complexity as its an unsupervised method and relatively fast. It's a ranked based technique where features are ranked according to information theoretic measures (Pearson correlation and Chi-squared) and top ranked features are selected for training. SelectKBest method from python sklearn library with $f - classif$ parameter has been used in this work that uses Analysis of Variance (ANOVA) f-value (f-value is calculated by dividing two mean squares) to rank the features. 25 features have been selected for each example out of 794 features extracted by TSFRESH and used for training and testing the model.

4.2.5 Evaluation Criteria

There are number of classification performance metrics such as accuracy, sensitivity, specificity, receiver operating characteristics (ROC) curve. In this work, accuracy will be used as our performance metric as the dataset at hand is balanced. Simply put, accuracy reflects how frequently the classifier predicts correctly. Accuracy is determined by dividing the total number of correct predictions by the proportion of true predictions as in 4.2. In the present study, there are three classes presented as $\{medial, middle, lateral\} \equiv \{1,2,3\}$.

$$Accuracy = \frac{Total\ number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4.2}$$

4.3 Results

Our machine learning pipeline was trained and tested on features extracted and selected from different time series (|Z|, θ, R, X) recorded during experimentation and their different combinations. For each case, different electrode pair(s) are used. This way, it was possible to see impact of different time series and electrode pair on the performance of machine learning algorithms.

First look at the case of univarite time series when only one time series is involved in analysis. For the data preparation, we have to take certain steps. The dataset was formulated taking a single time series. Our dataset input is defined as $D = \{x^{(1)}, x^{(2)}, ...x^{(i)}\}$ where $i \in \{1,2,...137\}$. For instance, for the case of impedance magnitude |Z| time series, the input vector $x^{(i)} = \{Z^{(i,1)}, Z^{(i,2)}, ...Z^{(i,j)}\}$ would only contain impedance magnitude recordings where $Z^{(i,j)} = \{|Z|_1^{(i,j)}, |Z|_2^{(i,j)}....|Z|_m^{(i,j)}\}$ and $m = 25 - 2(j-1)$ for $j \in \{1,2,...8\}$. Now the dataset is prepared for feature extraction procedure. The same procedure for impedance phase, resistance and reactance recordings. The input datasets are prepared to include different number of electrode pairs. There are 3 classes/labels for our dataset so $Y \in \{0,1,2\}$ for $\{medial, middle, lateral\}$. The output labels remain the same for any input dataset.

Once the input dataset was prepared, it is standardized and converted in to pandas dataframe so that features are extracted from it using TSFRESH. Now for every input vector of a specific electrode pair in an input dataset, there are 794 features extracted i.e., $Z^{(i,j)} = \{f_1^{(i,j)}, f_2^{(i,j)}, ...f_{794}^{(i,j)}\}$. For example, if we take univarite time series as an example with 3 electrode pairs, then we will have 137x3=411 input vectors for which features are extracted. Before the feature selection, there maybe a need for cleaning the feature set by removing any NAN/infinite values. Once we have cleaned feature set of the dataset, top ranked 25 features were selected for each input vector using SelectKBest feature selection method such that $Z^{(i,j)} = \{f_1^{(i,j)}, f_2^{(i,j)}, ...f_{25}^{(i,j)}\}$. Now the features of all electrode pairs of a specific input example would be flattened as a single vector. For example, for the above case of univariate time series with 3 electrodes, there would be 137x75 (rowsxcolumns) feature set to be trained and tested.

k-fold cross validation method is employed for training and testing the datasets to avoid any over-fitting and k=5 is choosen. The model was trained and tested 5 times in our case and produced the accuracy of the model in each iteration. During each iteration, the dataset was divided in to 5 folds (portions); 4 folds are assigned as training set and one fold as a test set so that training and test sets are different in each iteration as shown in Fig. 4.4. The final accuracy was the average of accuracies of all iterations.

There are three models (SVM, SNN, kNN) that are cross validated on specific and optmized hyperparameters. For SNN, we used a multilevel perceptron (MLP) with two hidden layers of size layer1=100 neurons and layer2=10 neurons, and finally a classification layer with 3 neurons. The hidden layers had an associated non-linearity of rectified linear units (ReLU). A softmax layer was used to convert the output of the MLP into probabilities of the respective classes. The network was trained for 1000 epochs using an Adam optimizer at a learning rate of 0.001. For SVM, Radial Basis Function(RBF) kernel function was used to train the data and Dynamic Time Warping (DTW) was used as a distance metric and 5 number of neighbors for the *k*NN classifier model. We have not used any weight class metric as our dataset is balanced.

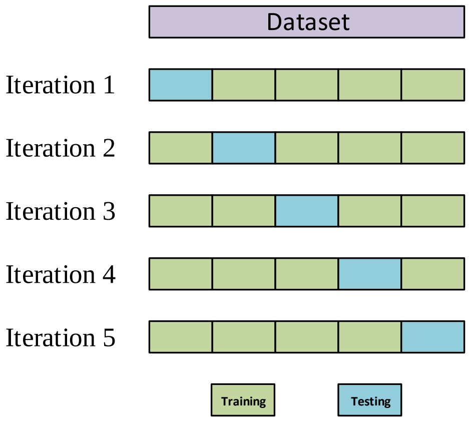


Fig. 4.4 5-fold cross validation mechanism for model evaluation

Fig. 4.5 presents accuracies when features extracted from univarite time series (|Z|, θ, R, and X) are cross validated using three machine learning models. The models were cross validated with different number of EPs, for instance, 3 on the x-axis means three apical EPs were used and so on with other numbers. The plot on the upper left of Fig. 4.5 shows the cross validated accuracies of features extracted from impedance magnitude recordings only. We can see, even with just one EP accuracies from SVM and SNN models are high i.e. >90%. SNN gave the highest accuracy with just one EP, however, SVM accuracy was highest when 3 EPs were included which is closer to using one EP in that case. SNN gave reasonably closer accuracy when 7 EPs were used compared to when one EP was utilized. In case of kNN, there is an increasing trajectory with regards to the EPs included in the validation except when all 8 EPs were used. The minimum accuracy started from 75% reaching the peak when 7 EPs were included in the analysis at 86%. Overall, SNN gave the best accuracy at 92.5%, then SVM at 92%.

The top right plot (Fig. 4.5) gives the cross validated accuracies when features from impedance phase θ was used for training and testing the models. As seen, SNN and SVM have almost identical accuracy line plots with respect to the number of EPs used in the cross validation. There is also upward trend in accuracy percentage when using more EPs. SNN has however better accuracy at 94% when using 6 EPs whereas SVM generated highest accuracy (93%) when features from 6 EPs were included. Again in case of kNN, accuracy was much worse compared to other two models highest being at 88% with 4 EPs data.

If we analyse lower two plots (Fig. 4.5) i.e., accuracies generated with features extracted from impedance resistive (R) and reactive (X_c) components, SNN model again outsmarted the other two models. SVM and SNN models in case of feature extracted from R time series gave highest accuracy (92% and 94% respectively) when only one EP's data was used. They also have reasonably decent accuracy when 4 EPs and 6 EPs were used respectively. For kNN model, the accuracy is relatively low compared to other two models except when 6 EPs data were used that is at 90%. On the other hand, when features extracted from X time series with different EP combinations, the only difference is SVM accuracy relatively lower than SNN compared to other time series. This is also noticeable that better accuracy is achieved when more than 1 EPs are used. SNN and SVM generated highest accuracy when 6 EPs were used, however, SNN accuracy 10% higher than SVM at 94%. In this case, SVM and kNN accuracies are relatively comparable where kNN giving highest accuracy when 4 EPs were used at 81%. Comparing the performance of different models against the time series and different number of EPs used, features extracted from impedance phase θ and reactance X gave the highest accuracy of 94%. Both of these highest accuracies came when 6 EPs were used that gave us the intuition that using high number of electrodes give high accuracy, however, the relation is not linear. There is also noticeable point that even though highest accuracy was achieved when using 6 EPs but we can still get reasonable accuracy even with one EP.

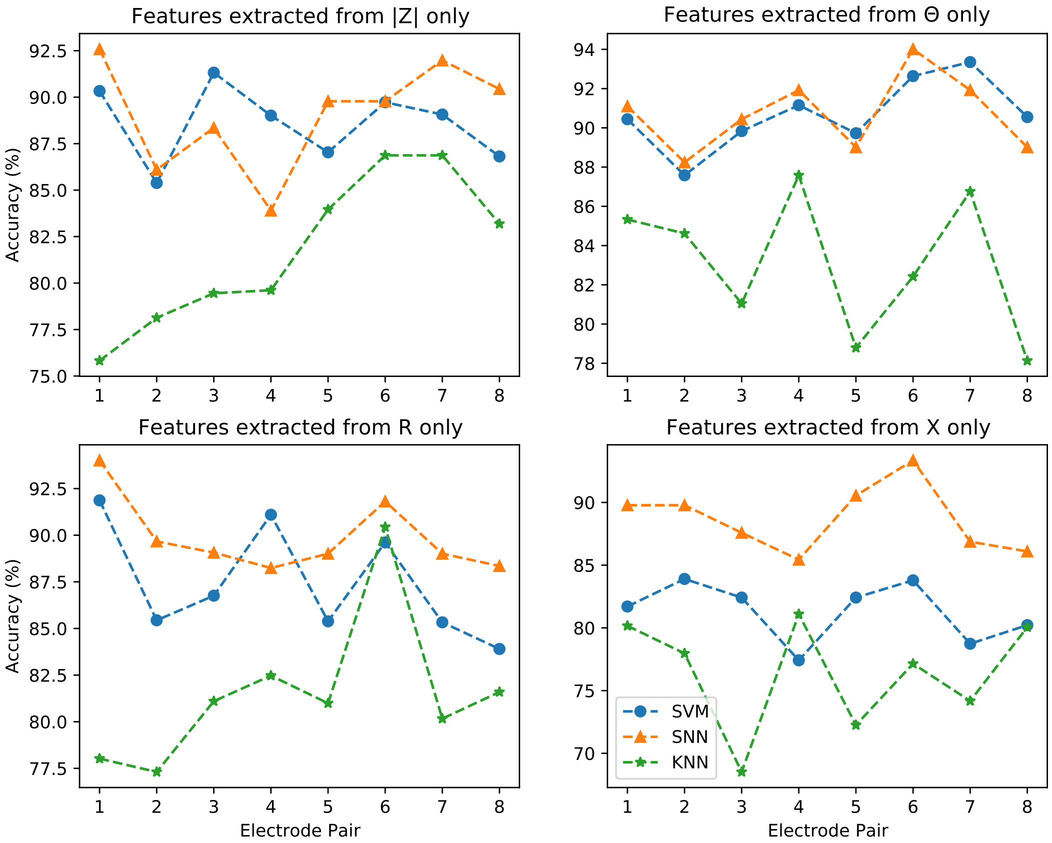


Fig. 4.5 Model (SVM, SNN, kNN) prediction test accuracies when models are trained with features extracted from: [Upper left] impedance magnitude (|Z|) time series only, [upper right] impedance phase (θ) time series only, [lower left] resistance (R) time series only, [lower right] capacitive reactance (X_c) time series only

Now we will look in to the performance of the models if we combine the features extracted from different time series. Three different combinations were cross validated (|Z| & θ, |Z| & X, θ & R) their accuracies using different combination of EPs are presented in Fig. 4.6. Now input vector to the TSFRESH would be 2 time series as multivariate inputs and separate feature vector was extracted from each time series and combined as an input example. After that, 25 features are selected from combined features extracted from 2 time series. This features set was then cross validated on three models and accuracies were evaluated.

The top plot in the Fig. 4.6 shows the accuracies when features extracted and selected from multivaite time series of |Z| and θ. We achieved highest accuracy of 96% with SNN model that is 2% higher than when only θ time series was used and 3.5% higher when |Z| time series was used alone. This accuracy was achieved when 6 EPS were included in the analysis. There were also improvements in the performance of other two models especially the kNN. The highest accuracy with SVM and kNN were found to be 96% and 95% when 6 and 7 EPs are used respectively. Again, accuracy with just one EP was reasonable, however, a bit lower than the highest accuracy for SVM and SNN models. Looking in to middle plot when models were trained and tested with features from multivariate time series ($|Z|$ and X), there is not much difference in accuracies compared to when impedance magnitude was alone used for analysis. Likewise, there are comparable performance of SVM and SNN models, however, kNN performed badly in this case. The lower most plot shows model performances in terms of accuracy when the features were extracted from impedance phase θ and impedance real part R. We have achieved the highest accuracy in our analysis so far at 97.5% and for the first time SVM outsmarted the neural network model by 0.5%. This high performance is achieved when features are extracted from 6 EPs data. kNN again performed poorly and achieved highest accuracy of 92% with the same number of EPs. From this analysis, we can say that we have sufficient evidence of better model performance with more one time series and high number of EPs.

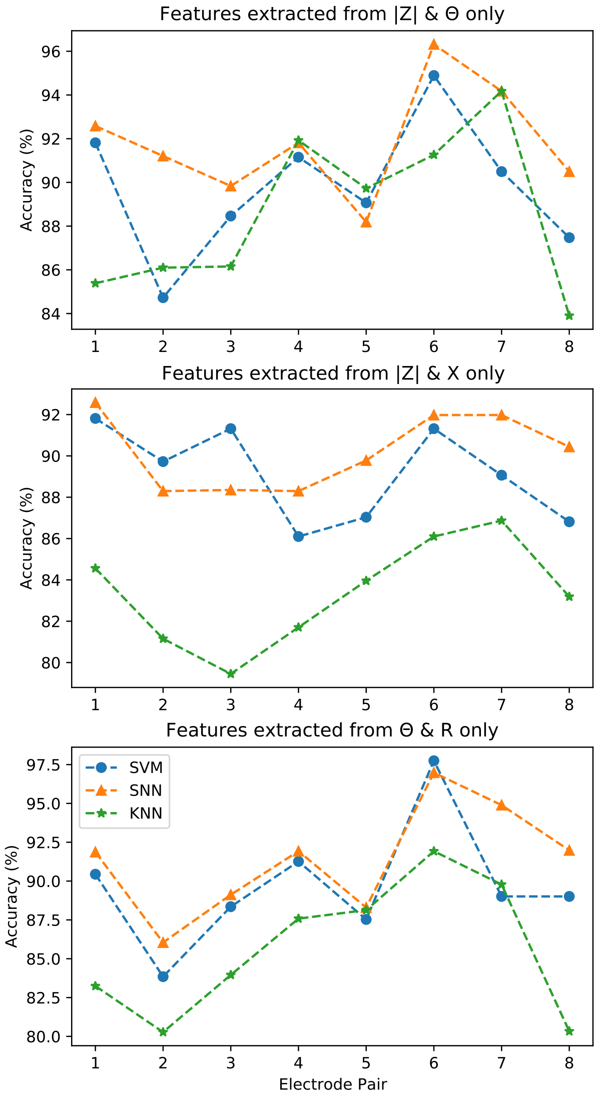


Fig. 4.6 Model (SVM, SNN, kNN) prediction test accuracies when model is trained with features extracted from: [Upper] |Z| and θ, [Middle] |Z| and X_c, [Lower] θ and R

Further to the analysis, all 4 time series data were used to extract the features and models were trained and tested. Separate feature set was extracted from individual time series for a specific data example and then 25 features were selected from all those features combined for each example. All three models were cross validated on these selected features and models' performance were evaluated. Fig. 4.7 shows model accuracies for different number of EPs used in the analysis. As we can see, SNN achieved the highest accuracy of 97.5% when 6 EPs were used. The same model, however, achieved reasonable accuracy of 95% when only one EP was used. The same percent accuracy was achieve with SVM when 6 EPs were used and 92.5% was when one EP was used for that model analysis. kNN again, achieved the lowest performance of all with highest accuracy of 92% when 7 EPs were included in the analysis.

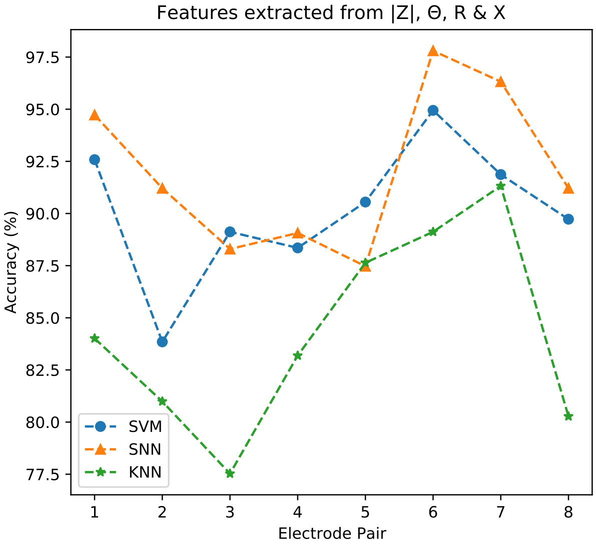


Fig. 4.7 Model (SVM, SNN, kNN) prediction test accuracies when model is trained with features extracted from all available time series data (|Z|, θ, R and X_c)

4.4 Discussion

The insertion angle of the cochlear approach or EA insertion trajectory has gained attention recently in the research community as it has been observed that certain trajectories are optimal for atraumatic insertion of electrode array [115, 116]. Array insertion from certain angles effectively has different force profiles both on cochlea walls and basilar membrane which eventually leads to intra-cochlear trauma [117]. Insertion trajectory also influence insertion depth (number of electrodes inserted successfully in the ST) of the electrode array. A bad trajectory could also lead to array translocation into scala vestibuli [118]. The ideal axis for the insertion is the centerline of the ST, however, it is not possible to insert from ideal axis due to facial canal position. For this, optimal axis has to be defined before the surgery to avoid facial canal. The mental representation of the optimal axis depends on the surgeon's experience in the manual surgery [119]. Wimmer et al. [104] proposed a method to find this optimal trajectory using two planes (inplane and outplane angle) for robotic surgeries. However, this is a pre-operative method and there is no information of EA direction when its inside the ST. This is the first step in our work that is vital to give information about the EA insertion direction inside ST and it could be further extended to adjust the direction during the insertion process with the robotic feed system in case it gets away from optimal path (next chapter).

4.5 Conclusion

A machine learning pipeline has been developed for offline analysis and prediction of electrode array insertion trajectories. In-vitro experiments were performed where electrode array was inserted from different angles (medial, middle, lateral) 137 times in total and bipolar complex impedance data were recorded scanning 8 electrode pairs during the insertion process. Three machine learning models were cross validated on different set of features extracted from uni-variate and multivariate time series complex impedance data. Also, inclusion of different set of EPs were also evaluated in terms of models' performance. We can conclude that model performance increased when we used features from multiple time series data and also there is enough evidence that higher accuracy is associated with higher number of electrode pairs used in the feature extraction process. However, there is no linear relationship between the model performance and number of EPs involved in the analysis. In terms of three models evaluated, shallow neural network achieved better accuracy than support vector machines and k-nearest neighbours which is inline with usually neural network better performance credentials with non-linear data.

Chapter 5

Online Prediction and Correction

Recently, advancements in machine learning techniques, sensing technologies and computation capabilities in the control systems, have led to increased interest in data-driven and learning-based control strategies [120]. The vast generation of data by complex systems carry important information about the structure and operation of these systems, for example, data collected from brain, autonomous cars, unmanned aerial vehicles and satellite navigation to name a few. When first principle model is difficult to obtain and complex, these data recordings can help scientists to understand, predict, classify and ultimately control the behaviour of the system [121]. Neural networks have been employed in modern control theory to approximate unknown functions for both system identification and control design. Reinforcement learning is another domain that is used to design the control systems where the agent tries to tune the policy to maximise the rewards to accomplish a target. However, reinforcement learning (RL) requires a large number of samples to train and tune the policy and its training in real time on actual hardware instead on a simulator sometimes may not be feasible.

Real time tracking of the electrode array trajectories during insertion procedure and avoidance of deviation from centre line path can help minimize the placement errors. An automated system can apply the necessary corrections to reach the target position if deviations from the planned path are quantified and available at all times. Previously, analytical methods and kinematics models were used for path planning of the insertion process and there has not been a lot of work done for real time trajectory prediction and correction for electrode array insertion without adding an additional sensor on the array. Data-driven models represent an alternative approach to estimate the electrode array position in scala tympani.

Under actuated dynamical systems can be found in a number of domains, for example, in engineering and biology. There are several advances made for controlling such system especially in robotics and healthcare. For instance, in healthcare such control systems provide treatment solutions to number of diseases such as Parkinson's, autism spectrum disorder, and cardiac arrhythmias. These techniques are mostly based on traditional mathematical models from control and optimal theory. Such control methods work well when system dynamics are known and it is possible to model the dynamics accurately. However, it gets difficult for complicated dynamical systems such as neural activity networks. It may be possible to model the systems using traditional methods but designing control strategies of such under-actuated systems is challenging. Artificial intelligence (AI) models can be used to control dynamical systems using deep neural networks. However, such AI models are always dependent on large set of data and their black box nature makes analysis difficult.

The main focus of this chapter is to present application of an ensemble of machine learning algorithms for real time closed loop feedback control of in vitro insertion process of electrode array in cochlear implantation. It is the first step in this domain to develop intelligent controllers for robotic surgical procedure that can learn from its past experience and acquire the relationship between operating conditions and optimum actions.

For feedback control, one of the approaches utilizing sensor measurement data is to classify it into one of the finite set of classes (situation) and each class corresponds to a known control action [122]. In one of such research project, image data was used to design a controller using this technique to control a quadrotor to navigate a forest trail [123]. Images were captured from three sides (left, center and right) of real world hiking trails and multi class (3) deep neural network classifier was trained and tested on these images. Real time feedback controller was designed according to the 3 classes, for example, when an image is classified as right side of trail, the quadrotor will turn left and so on. Based on this work, we have implemented our system using time series data acquired from impedance sensing unit for EA optimized steering during cochlear implantation. A plastic ST perception technique based on machine learning algorithms has been developed that bypasses the problem of determining the underlying characteristics of ST.

We are considering supervised learning to design a closed loop feedback control for EA insertion. For this purpose, we showed how to train an ensemble of machine learning algorithms for classification of trajectories based on performance related to learning from data. Supervised learning in essence entails solution to an optimization problem where objective function is actually a cost function that we want to minimize. Classifiers adopted in this scenario make decisions on sequence of features that are dependent. Therefore, classifier's performance is evaluated on the basis of generated closed-loop behaviour.

This chapter deals with the online prediction and correction of the electrode array insertion angles/direction and trajectory using complex impedance data and data driven closed loop control system. It starts off with the preliminaries of the problem at hand followed by the problem formulation. Since we are using raw complex impedance data recorded during experiments, the next section presents machine learning algorithms' performance in an offline setting. This section is divided in to full insertion data as well as subsequence data evaluation for insertion direction and path classification. The chapter then moves on to presenting an online closed loop system for insertion correction using ensemble learning approach. Lastly, online prediction and correction results are presented followed by the conclusion [1].

5.1 Preliminaries

Lets consider a dynamical system shown in Fig. 5.1 where the dynamics of the system can be expressed as

$$x' = f(x,u) \tag{5.1}$$

In Eq 5.1, state is represented as $x \in X$ and control input is $u \in U$. x' represents change in the state x. f is the function that defines the drift (shift/relocation). From the Fig. 5.1, the sensor generates the feature $s \in S$ and is dependent on the state x and their relationship is unknown. The classifier maps the feature s to a unique label $b \in B$. The aim of the classifier is to learn the weights $w \in R^n$ to minimize the error between the predicted label and the actual label associated with a certain feature vector. The set of labels $B = \{b_L, b_M, b_R\}$ corresponding to the position of electrode array on the left, middle or right of the ST model's horizontal plane. The three labels are associated with three different control actions that carrier has to implement for electrode array to stay in the middle of the ST (away from walls). These are:

1. **Move Left**, if the array is touching the right wall
2. **Move Right**, if the array is touching the left wall
3. **Go Straight**, if the array is in the middle

The data set consists of the features and the labels (x^j, s^j, b^j) where j represents index of an example in the data set. The classifier tries to find the relationship between the features and the labels. The feature vector s depends on the state x, so we can say that the classifier indirectly learns the mapping from state to the control. This way, classifier generates the control input u to the system and results in a closed loop feedback system as shown in Fig.5.1.

[1] Part of this chapter has appeared in Hafeez et al. [108]

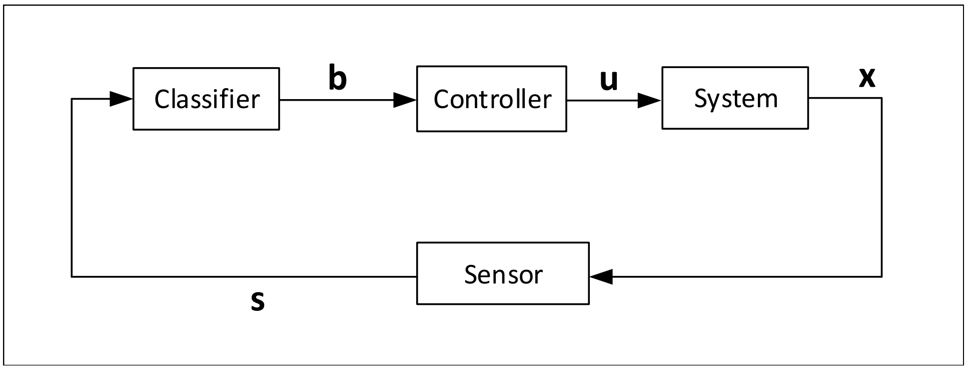


Fig. 5.1 A classifier in a loop feedback control system

5.2 Problem Formulation

The insertion of the electrode array is configured such that it is presented as (x, y, θ) in the plane where (x, y) points are considered as the location of array tip and θ angle represent heading of the array with respect to centre line of the ST model towards either of the modular/lateral walls (as depicted in Fig. 5.2 (A)). Let $\vec{d}$ be the center line trajectory of the electrode array and $\vec{v}$ is defined as direction of array insertion away from the center line direction. The goal of the insertion is to keep the electrode array in the center of straight path of ST model until it touches the curved path from where it slides along the curved path. The motivation of keeping electrode array on the center line trajectory is two fold; (1) it would not damage the delicate structure of the ST walls, (2) the center line insertion imposes less overall force when the array touches the curved path and slides along it.

For this problem, we are considering ST model as two paths; one is the straight path and other the curved path as shown in Fig.5.2(B). One of the classification models would predict whether the position of the electrode array is located in straight path or curved path at a certain time. In this work, we will be generating the control signals for only straight path as we were unable to find any mechanism to control array in the curved path.

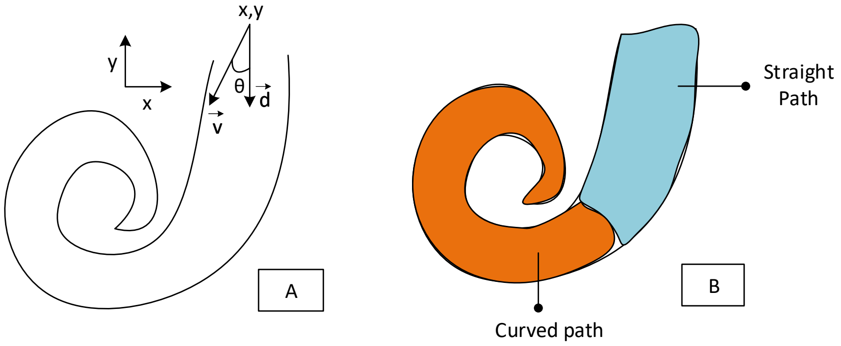


Fig. 5.2 (A) Electrode array insertion configuration (B) Division of insertion path in to straight and curved sections

5.3 Offline learning with raw signals

We have demonstrated classification on full and partial insertion data using standard machine learning pipeline that includes major feature extraction and selection process in section 4.2.3. This process takes a lot of resources and time (for example, 4 EPs data take 5.4 seconds to extract and select features) and is not fit for our application. There is another method to train the models with raw data or normalized raw data. This method fits better for our application to skip time consuming feature extraction/selection as time delay is critical for real-time control application. We will first look into the generalization of machine learning algorithms on full insertion data to classify different trajectory insertion. Further to that different evaluation metrics will be used to test those algorithms on partial insertion data to make sure the data is reliable enough to train a model to be used as an online predictor. There would, however, be limitation on the generalization of the model due to different environmental conditions (e.g., room/saline solution temperature) between two sets of data (data collected for training and testing and the data collected during online/real-time testing).

5.3.1 Full Insertion Data Evaluation

Electrode array is inserted from three different angles and takes three different trajectories during the insertion process. In this analysis, generalization of three classification algorithms (SNN, KNN, SVM) are looked into using the full insertion data of 4 electrical features (|Z|, θ, R and X) of electrode pairs. Moreover, combination of different electrode pairs data are analysed to see their impact on the classification accuracy. This analysis would help in establishing whether its possible to detect which trajectory electrode array took during the insertion using the complex impedance data. Also, this analysis is carried out on standardized raw data. After the data is organized for each combination of electrode pairs, machine learning models are trained and tested using 5-fold cross validation procedure. Hyperparameters of algorithms and evaluation criteria remained the same as in the analysis presented in Chapter 4. This will help to make comparisons of algorithms generalization ability with and without feature extraction/selection process.

Classification accuracy achieved by each learning model on different combination of electrode pairs data is presented in Table 5.1 where EP1 is the most apical pair. For example, row 4 represents a dataset comprising of four feature time series of the first four electrode pairs (EP-1,2,3 and 4) and test accuracies of three machine learning (ML) models are presented when trained and tested on the said dataset. The last row presents the test set accuracies when all 8 electrode pairs are considered in the modeling.

It is evident from the test accuracies of the three classifiers, higher test accuracy was achieved when we used four electrode pairs. It is potentially due to the fact that first 4 electrode pairs observe most of the force from the lateral wall after EA touches the wall and rub against it. Most of the time, these 4 electrode pairs are only touching the lateral wall after the full insertion. Also, high impedance change is observed after EA touches the lateral wall when it is curving to slide along the wall. It is also observed that a single electrode could also have high predictive information, for example, the first-row accuracy results where only the first pair was used. Relatively high accuracy of 94.2% is achieved using SNN. Two ML models (SNN and SVM) gave higher than 90% accuracy to predict three classes (medial, middle, and lateral direction insertions) except on four occasions (*kNN* when 1, 5 and 7 EPs are used, SVM when 6 EPs are used) considering all selection of pairs. SNN and SVM performed better than KNN for most of the EP data combinations. Overall SVM model gave the highest accuracy of 97.1% when we used four electrode pairs. With the same EP combination, SNN gave a test accuracy of 92.3%.

5.3.2 Partial Insertion Data Evaluation

Although high accuracy results are achieved to predict different insertion trajectories using complete insertion measurement data, it is important to know how well the selected models predict on different subsequence datasets. There are two ways the models are tested to classify the subsequence data in this analysis. First, the classification of trajectories on small window of data from the insertion depth. Secondly, classification of insertion path whether the electrode array is in the straight path or on the curved path at a particular instance

Table 5.1 Test accuracies of full insertion data with 4 raw features to classify insertion direction/trajectory

	EP	Accuracy (%)		
		SNN	SVM	KNN
$\lvert Z\rvert,\theta,R,X$	1	94.2	90.9	88.0
	$1-2$	92.3	94.7	92.8
	$1-3$	90.4	91.9	92.8
	$1-4$	92.3	**97.1**	92.8
	$1-5$	93.3	94.7	85.7
	$1-6$	94.2	89.5	90.4
	$1-7$	92.8	93.3	88.0
	$1-8$	93.3	90.9	90.4

as shown in Fig. 5.2(B) where blue part is considered as straight path and orange as curved path according to the ST model structure.

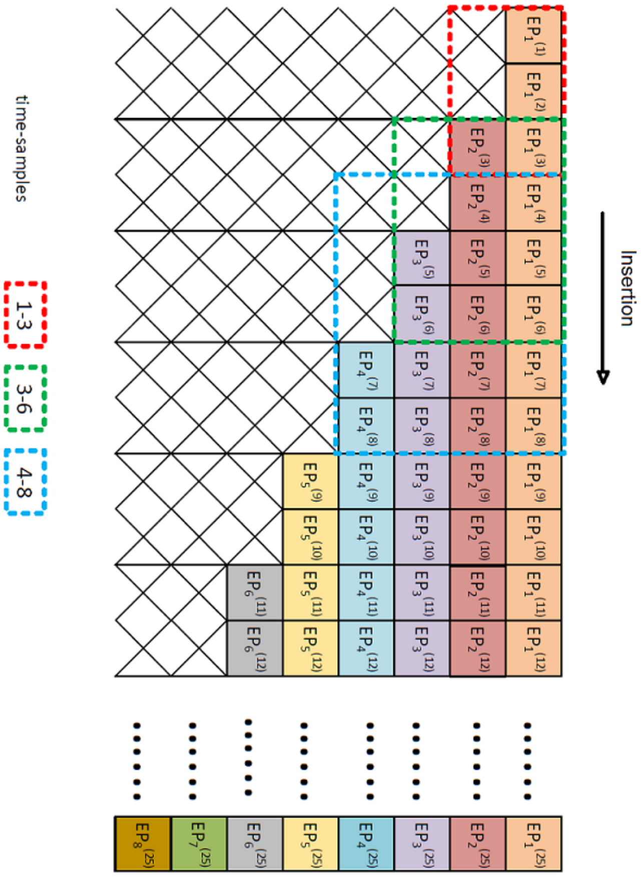


Fig. 5.3 subsequences (window) selection from the insertion data for the short-term trajectory prediction

Classification accuracy for trajectory (medial, middle, lateral) is tested by two subsequence selection methods. First, subsequences are extracted from the complete time series data as shown in the Fig. 5.3. The crossed boxes are high impedance values (when a certain EP is out of saline solution during insertion) that are excluded from the analysis. In this method, subsequences are selected with respect to the time samples and include multiple electrode pairs in the analysis. For example, timesamples 1-3 in the first row of the Table 5.2 includes EP1 and EP2 data (3 samples of EP1 and 1 sample of EP2 from begining of insertion) as shown in Fig. 5.3 red dotted box. The samples from both pairs are then combined to form a vector for training and testing of the algorithms. In the same way timesamples 3-6 contains samples from EP1-3 starting from sample 3 as in Fig. 5.3 green dotted box and they formed a vector for further analysis.

The first column of Table 5.2 presents different time samples combination to form subsequences and the other three columns depict test accuracies at three ML models. With these subsequences, SNN overall gave the highest accuracy and SVM came in second. The highest accuracy was achieved (93.8%) when time samples 2-8 of each feature and relevant electrode pairs (which are inside the saline-filled model) were trained and tested. However, it takes time to wait for eight-time samples before we know the insertion direction. The sweet spot is when 1-4 time samples were used and 86.1% accuracy is achieved. This could work well for online prediction of EA insertion direction and the actuation system could be utilized to correct the direction during run time based on the prediction.

Table 5.2 Classification test accuracy achieved with three machine learning algorithms on $|Z|, \theta$, R, and X values using different sub-sequences (different EP data combined)

timesamples	Accuracy (%)		
	SNN	SVM	KNN
$1-3$	85.2	78.5	80.0
$1-4$	86.1	82.3	80.9
$1-5$	82.8	78.0	71.9
$1-6$	87.1	83.8	83.3
$1-7$	86.1	81.4	76.6
$1-8$	86.6	89.5	74.7
$2-4$	86.1	79.5	81.9
$2-5$	87.6	79.0	76.6
$2-6$	86.6	82.8	86.1
$2-7$	87.6	79.0	80.0
$2-8$	93.8	81.9	78.5
$3-5$	84.2	76.6	73.8
$3-6$	92.3	78.0	80.4
$3-7$	92.8	81.9	80.0
$3-8$	91.9	85.2	84.2
$4-6$	82.8	75.7	73.8
$4-7$	90.9	76.6	72.3
$4-8$	91.4	80.9	72.8
$5-7$	87.6	77.1	77.6
$5-8$	85.7	79.0	79.0

For the online prediction of trajectory, data from the most apical electrode pair (EP1) is used to train and test the algorithms first and then use the trained model for online prediction with the new collected data during the insertion process. Different length subsequences of EP1 data are used, that is, subsequence data collected during 4 mm, 3mm, 2mm and 1mm insertion. For these subsequences different set of data is assembled for analysis and only straight path is considered for prediction and further correction. For example, for 4 mm insertion depth, 2 subsequences of 4 time samples each are extracted from each insertion example. These 2 subsequences will make 2 separate examples in the new dataset. That means now there would be 137x2 (274) examples in this dataset with respective targets (medial, middle, lateral). In the same way, for 3mm subsequence, there will be 3 subsequences of 3 time samples each and each subsequence will make an example in the new dataset (137x3=411 examples) with respective target label. Moreover, new dataset for 2mm and 1mm subsequences have 137x4=548 and 137x8=1096 examples according to the split of the straight path data. It is important to mention that we are using all 4 features (|Z|, θ, R and X) in this analysis.

Table 5.4 shows the machine learning algorithms performance in terms of their cross-validated accuracies. It is noteworthy to mention that due to low accuracy of SVM algorithm, random forest algorithm is trained and tested and included in the analysis due to their ensemble learning approach. As can be seen in the table, high accuracy is achieved when 4mm subsequences are used and second-high accuracy is achieved when 2mm subsequences are used. 1mm subsequences achieved the lowest accuracy. This behaviour is attributed to SNN and KNN algorithms whereas the accuracy decreases as the subsequence length is decreased from 4mm-1mm. In terms of best performing algorithms, SNN gave the best accuracy among the four algorithms cross validated while the SVM algorithm performs the worst.

Table 5.3 Cross validated multi-class accuracies for trajectory prediction using EP specific sub-sequences during straight path insertion

	Sub-Seq	Accuracy (%)			
		SNN	SVM	KNN	RF
EP1	4mm-2	**86.8**	69	79.1	82.5
	3mm-3	78.8	64.2	72.0	73.5
	2mm-4	80.3	65.3	77.0	72.8
	1mm-8	75.2	66.7	76.3	71.2

For trajectory prediction, two methods of subsequencing have been used and tested. Further analysis is carried out to predict whether the electrode array is in the straight path or the curved path during the insertion. For this, we are using the full insertion data and splitting them in to straight path and curved path data and labelling them accordingly. Since now we only have 2 classes (straight or curved), we termed such problem as binary classification. Different data sets are formed according to the subsequence length, labelled accordingly and cross validated on four machine learning models. For example, for 4mm subsequence there will be 2 subsequence examples of straight path and 3 subsequence examples of curved path. That is, there will be 5 examples in the new dataset out of one example from the original dataset. The examples in new dataset will be given target labels according to the path (straight or curved). 4mm-2C/3S dataset will have 137x5=685 examples to trained and tested using the algorithms. In the same way, new datasets namely 3mm-3S/3C, 2mm-4S/5C, 1mm-8S/10C have 137x6=822, 137x9=1233, 137x18=2466 , examples respectively.

Table 5.4 presents the accuracy results for the path prediction during the insertion process using different subsequence length datasets. These results show that as the subsequence length is decreasing, the accuracy of all algorithms also decreases. This is obvious because of the shorted length vector, we have less information available for prediction. However, the decrease is not linear due to increased number of examples on which the model is trained and test that makes generalization easier. As show in Table 5.4, again the highest accuracy of 87.4% is achieve by SNN on 4mm subsequences. Overall, SNN also outperformed other machine learning models in terms of accuracy. The lowest performer was random forest model with accuracy of 65.9% when 1mm subsequences are used.

Table 5.4 Cross validated binary class accuracies for straight/curved path prediction using EP specific sub-sequences

	Sub-Seq	Accuracy (%)			
		SNN	SVM	KNN	RF
EP1	4mm-2S/3C	**87.4**	81.3	83.7	83.2
	3mm-3S/3C	82.4	76.4	76.9	78.3
	2mm-4S/5C	78.7	73.2	76.6	74.4
	1mm-8S/10C	64.2	67.1	68.2	65.9

5.4 Real-time classifier in the loop system

Now there is enough evidence from our offline predictive analysis that machine learning algorithms can be used as an online/real-time classifier for electrode array steering during insertion. By this we can achieve low force profiles during EA array insertion that could lead less traumatic intraoperative behaviour. This online method is used for correction of the trajectory of the electrode array during the insertion based on the prediction of machine learning models. For online prediction, we have used two classifiers; 1) Classifier A for trajectory prediction, 2) Classifier B for path prediction. Both classifiers would be trained on 4mm subsequence data of the most apical electrode pair because this is where most of the information about insertion is.

Although a reasonable accuracy has been achieved by our individual model, however, ensemble approach (applying the voting algorithm on three selected models) has been used to increase the classification accuracy for both our classifiers. According to [124], ensemble learning is the method in which multiple classifiers are combined to take their weighted vote to make predictions on new data. Each classifier makes its prediction and then votes for the final prediction as shown in Fig. 5.4. This figure explains one of the ensemble learning process that is called majority voting ensemble. In this kind of scheme, final predictor would be predictor that gets the majority vote from the multiple classifiers. In our setup, we have three classifiers (RF, SNN and KNN) that achieved the maximum accuracy. Although it is possible to give uneven weights to the individual classifier vote, however, equal weight is given to each classifier in our system.

All the offline classification is carried out using Python 3.6 but in order to keep a homogeneous system along with actuation and sensing, machine learning system is developed again in MATLAB. All the hyper-parameters of learning models remain the same and individual models are trained and tested again to make sure performance are kept the same as offline models developed in Python. The ensemble majority voting algorithm is also developed in MATLAB.

Once we have ensemble based trained models for both classifier A and B, we can setup the system for online prediction and correction. Fig. 5.5 shows the control flow graph for online prediction and correction of insertion trajectory. As mentioned above, this control strategy is for straight path only to keep electrode array at the centerline of the ST model. For this, there is need to first predict whether the electrode array is in the straight path or the curved path.

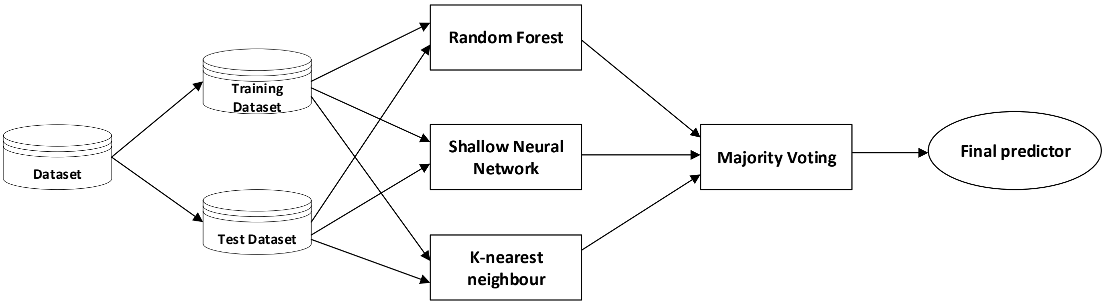


Fig. 5.4 Ensemble of classifier for real time classification

According to Fig. 5.5, online prediction and correction starts with electrode array insertion and complex impedance data recording. First, the electrode array is inserted for 4mm and complex impedance (|Z|, θ) are recorded during insertion and impedance resistive (R) and capacitive reatance (X) are calculated with |Z| and θ. As earlier mentioned, our system record 1 sample/mm/feature, therefore, a vector is formed of size 1x16 (4 samples for each |Z|, θ, R and X). This vector is first fed as a test input example to the classifier A that would predict the path of the electrode array during 4 mm insertion. The path could be straight or the curved path as shown in Fig. 5.2. That means, there are two options and if the model predicts EA to be in the straight path (label represented as bi), the recorded data vector should be fed into the classifier B. This classifier predicts the trajectory of the EA whether it's sledded along the lateral or medial wall or it's in the middle of the ST model. Classifier B is an ensemble classifier that is trained with 4mm subsequence data for trajectory classification. Once the recorded data is fed, this classifier makes prediction and decides whether the correction of trajectory is needed or not. If the trajectory is towards medial or lateral wall, the correction is done by moving the EA towards the middle of the ST model. For example, when the trajectory is predicted to be *medial*, controller moves the rotational axis 0.5° towards the right. When the trajectory is predicted as *lateral*, the controller moves the rotational axis towards the left to the factor of 0.5°. If the trajectory is in the middle, no action is taken. To confirm the correction, complex impedance data is recorded again but this without any further insertion. The flow is moved back to the data measurement block and similar vector is generated with new collected data as EA is still in the ST model. Classifier A and B would again predict the path and trajectory. The controller will again make the correction according to the classifier B prediction. On the third iteration, EA is inserted further for 4mm and the same procedure is repeated again and controller makes the prediction according to the prediction. We can set the number of iterations *n* that we want to make during the insertion process.

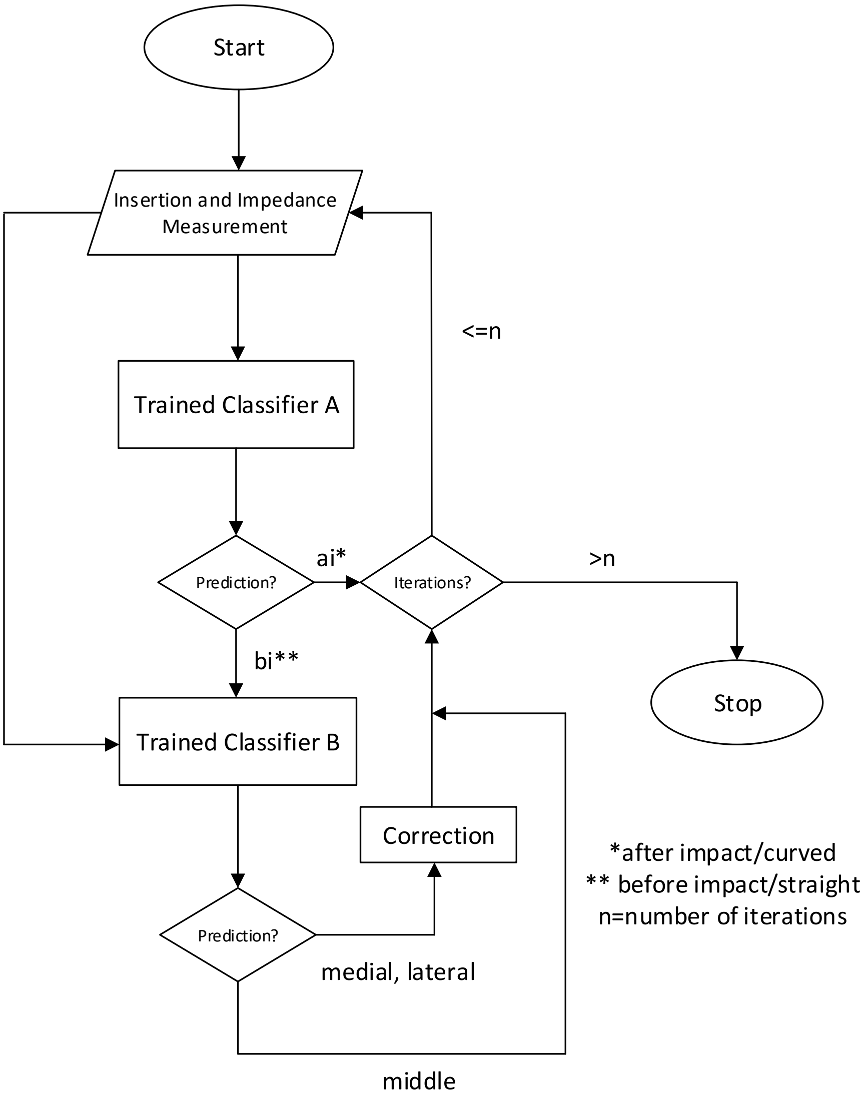


Fig. 5.5 Flow chart of real time prediction and correction

Figure 5.6 presents the pictorial representation of one of the online experiments. In Fig. 5.6 (A), the experimental setup is shown as the EA is inserted for 4mm and data is collected, EA trajectory is towards medial wall as EA can be seen inside saline filled ST model. Close up of this trjectory is shown in (C). Once the recorded data vector is passed on to classifier A and classifier B, controller makes the correction by moving the rotational stage of actuator towards right by 0.5° as shown in Fig. 5.6 (B) and close up in (D). The trajectory could be towards lateral wall as shown in (E) and is corrected as shown in (F).

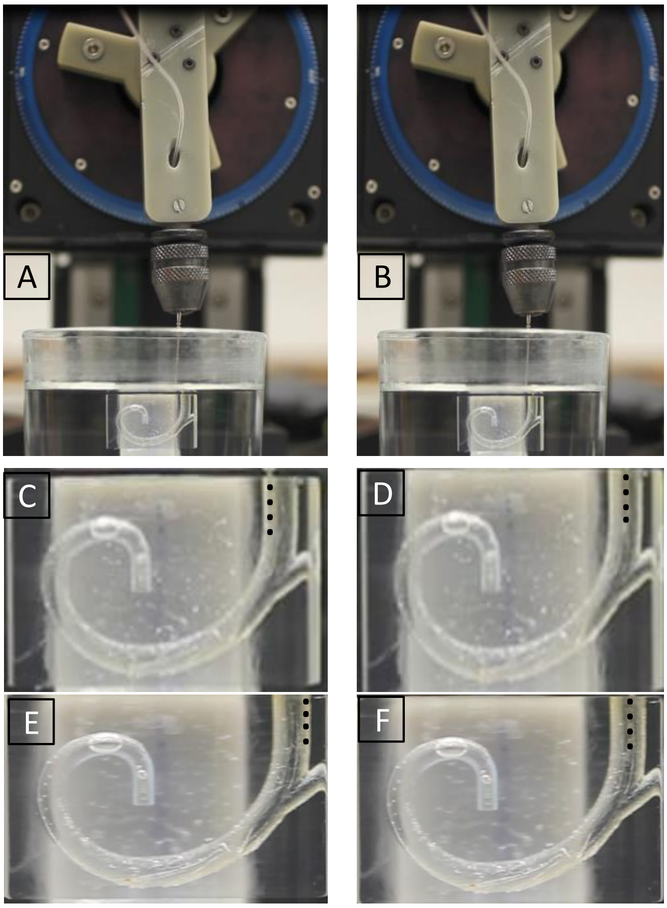


Fig. 5.6 online correction of insertion trajectory. (A) experimental setup with medial trajectory insertion, (B) experimental setup with corrected to middle trajectory, (C) closeup of medial trajectory insertion, (D) correction of medial trajectory, (E) lateral trajectory, (F) correction of lateral trajectory

Table 5.5 shows the ensemble learning classification results for both offline and online predictions. Ensemble learning models (classifier A and classifier B) are trained and tested first using three machine learning algorithms. Classifier A ensemble model achieved accuracy of 89.05% that is an increase from 87.4% when the highest performing individual model is used for path prediction. For classifier B, cross validated accuracy of 86.86% is achieved using ensemble learning that is slightly higher than the highest performing individual SNN model. For online prediction, 30 iterations are performed to test both classifier ensemble models. Classifier A achieved 83.33% accuracy whereas classifier B achieved slightly less accuracy of 80% when tested for online prediction and correction.

Table 5.5 Ensemble classification results.

Ensemble Classification Model	Offline Accuracy (%)	Real-time accuracy (%)
Classifier A	89.05	83.33
Classifier B	86.86	80.00

5.5 Conclusion

This chapter presents an important aspect and procedure for robotic system aimed to be used for electrode array insertion in cochlear implantation to avoid trauma. A novel system is developed for online prediction and correction of trajectory of electrode array during electrode array insertion. This system is open to further development and can be beneficial for automatic steering of EA during insertion. The system is tested using the most apical EP and in the straight path only. It can be extended to other EPs and to the curved path as well given a novel actuation scheme is developed alongside to steer the EA in curved path. The system presented can also find application in other electrode insertion application and also in needle steering.

Chapter 6

Electrode Array Insertion Depth Estimation

Electrode array insertion depth can be assessed by medical imaging modalities e.g., CT, either at the end of the procedure or sometimes during the surgery if imaging device is available in the operating room [9]. However, those imaging techniques come with the risk of radiation exposure and mostly avoided especially during the insertion process [125]. It also increases the time and cost of the surgical procedure.

Several studies have used different methods to predict EA insertion depth prior to the implantation as part of the surgical planning process [126]. In one of the studies, basal turn diameter and width were used to predict the angular insertion depth using a planning software [127]. The results were compared with the routine prediction method using 2D CT images. [128], [129] also presented methods to predict insertion depth.

In other biomedical applications, an ultrasound based percutaneous needle insertion guidance robotic system has been developed which is used for real time monitoring of positioning and orientation [130], [131]. Real-time monitoring of insertion depth of a optical fiber based needle array was carried out for cancer treatment [132].

We have seen complex impedance measurements vary with respect to the EA position in the cochlear model during insertion. Giardina et al. [13] briefly touched upon the relationship of response magnitude with insertion depth in their research with evidence from the in-vitro experiments. It was concluded that magnitude response (in mV) increases with the insertion depth, however, there was no further comment on if it's possible to estimate insertion depth with impedance measurements. Magnitude response of most apical electrode was compared at different insertion depths as well as magnitude response of most deep and shallowest electrodes were also compared after the full insertion. Electrocochleography (ECochG) has also been used intraoperatively for monitoring of EA insertion and there is a strong correlation between response to ECochG and EA scalar position during implantation [133], [134], [135]. Therefore, ECochG has the potential to estimate insertion depth.

Aebischer et al. [136] recently found out strong correlation of intra-operative impedances and insertion depth, however, insertion depths were only estimated after the full insertion. Tissue resistances were extrapolated from transimpedance recordings and their relationship with insertion depth were statistically validated. Final electrode positions were acquired from CT images of 20 implanted subjects. All subjects were inserted the same lateral wall electrode array. The proposed model can estimated the linear insertion depths by a margin of 0.76 ± 0.53 mm (mean and variance). However, the shortcoming of this approach is that it only estimates insertion depth after the full insertion.

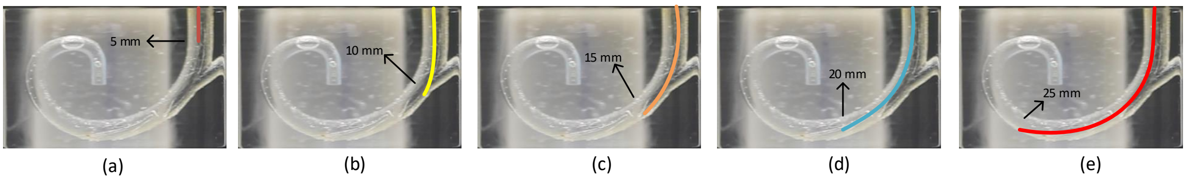
Post surgery hearing outcomes dependency on EA insertion depth is very much debatable among researchers. For example, according to [137], it is concluded that deep electrode array insertion gives better speech perception whereas in another study it is the opposite due to potential effects of trauma because of deeper insertion[138]. However, CI surgery needs objective measures to improve insertion process and assess surgical outcomes. These measures could also be used for real time monitoring for EA insertion process.

Our focus in this chapter is to introduce a method where it's possible to estimate insertion depth during the insertion process using complex impedance data. Analysis is divided into two sections as a classification and regression problems for estimation of linear insertion depths of electrode array. The classification part deals with estimating final insertion depth and regression part corresponds to depth estimation at each time step during insertion. For classification problem, four algorithms (SVM, KNN, SNN, and RF) are used to train and test the dataset. Moreover, a hybrid learning model of one dimensional convolution neural network (CNN) and long short term memory (LSTM) networks is employed to estimate linear insertion depth at every time-step.

6.1 Classification of Insertion Depths

6.1.1 Dataset Formation and Visualization

Apart from using the same dataset as in Chapters 4 and 5, we are going to introduce another dataset with complex impedance measurements at different insertion depths. Electrode array was inserted 10 times for each insertion depth. Insertion depths we tried were 5 mm, 10 mm, 15 mm, 20 mm, 25 mm as shown in Fig. 6.4. Therefore, we have 50 examples in this dataset. Each example has 4 time series features (Z, θ, R, X) as the other dataset for insertion trajectory. However, we collected 25 samples for a time series each even if the EA is inserted for less than 25 mm. For example, the system would measure 5 samples each for a time series for 5 mm and then the rest of 20 samples would be measured when EA is static at 5 mm depth. This is to ensure, we have equal length time series for training and testing. Electrode array is inserted from the middle of ST model in all experiments of this dataset. For regression analysis, dataset with examples of insertions from different directions will be used for maximum coverage of the ST model.



(a) (b) (c) (d) (e)

Fig. 6.1 Electrode array insertion at different insertion depths. (a)5 mm, (b)10 mm, (c)15 mm, (d)20 mm, (e)25 mm

Fig. 6.2 shows graphical representation of the complex impedance measurements taken during the experiments. The graph shows impedance magnitude, phase, resistive and reactive parts of first electrode pair EP1 at different insertion depths. We have electrical measurements row-wise and different insertion depths column-wise. Graphs are plotted for 25 samples, however, measurements are taken both while electrode array is continuously inserted and at rest for depths lower than 25mm. Each time sample corresponds to 1mm insertion depth for 25mm depth. For other insertion depths, partial measurements are taken when the array is static after inserted to the corresponding depth. The graphs shown in the figure are mean and standard deviation of the respective insertion depth experimental data. Looking at the first column, when EA is inserted for 5 mm, impedance magnitude decreases even though EA remains at the center of the ST model. However, the decrease is small and decrease is continuous during both instances when EA is moving (inserted) for 5mm and then static for rest of 20 time samples. On the other hand impedance phase is getting less negative but the most change is observed when the electrode array is moving and not static. The change is also small in the region of less than 0.5°. In the same way decrease in resistance and reactance can also be seen. In the second column, more change in all electrical properties are observed when the EA has touched the lateral wall and rests there. This is due to proximity of EP1 to the plastic material that is high impedance material than the saline solution. We can see there is 100Ω increase in |Z| and 1° change in phase θ. Similarly, for insertion depths of 15mm, 20mm and 25mm, there are more pronounced changes in electrode pair complex impedance after it touches the lateral and slides along it or rests there.

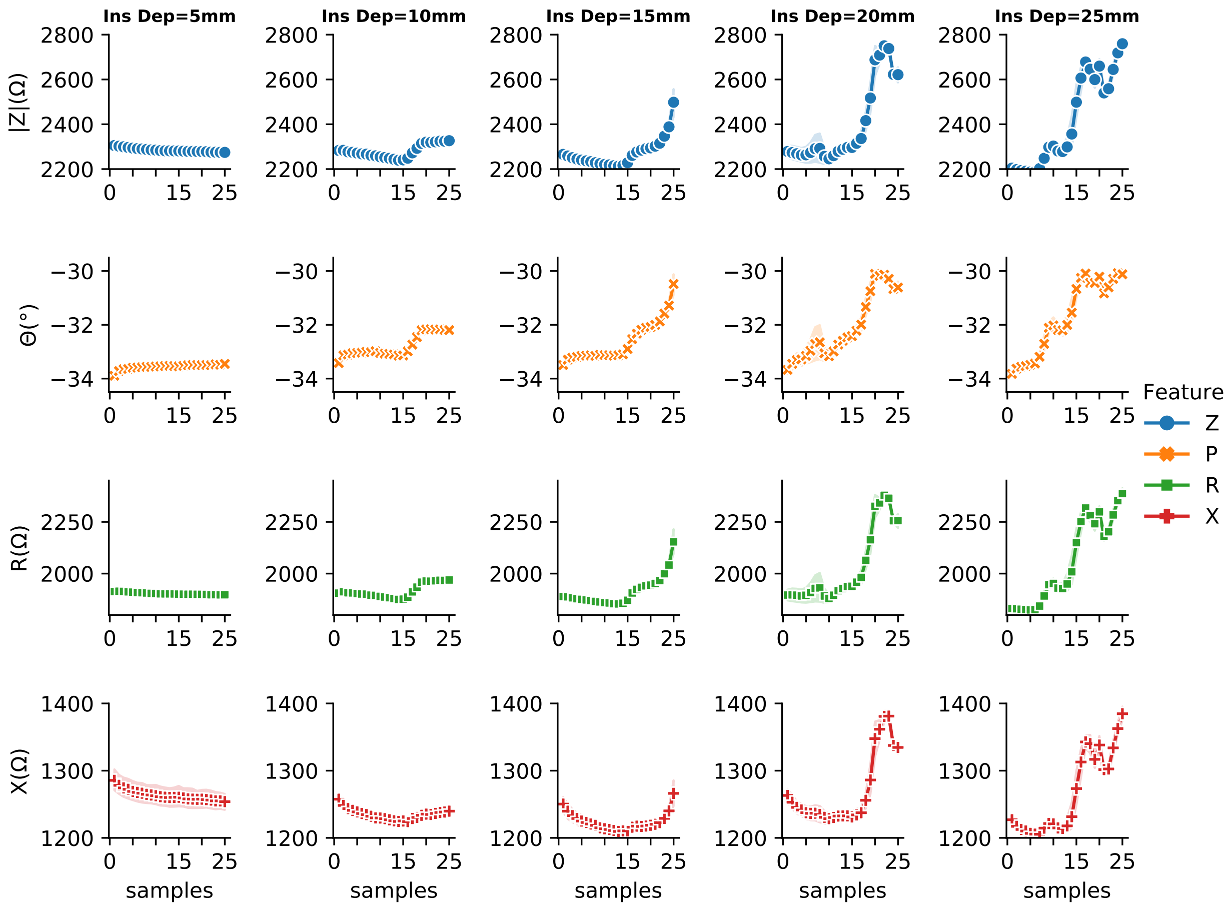


Fig. 6.2 Electrical impedance visualization of EP1 at different insertion depths. Columns[left-right]: 5 mm, 10 mm, 15 mm, 20 mm, 25 mm. Rows[top-bottom]: Impedance magnitude, phase, resistance and reactance

6.1.2 Classification Results

This dataset is defined as $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)})...(x^{(i)}, y^{(i)})\}$ where $i = 50$, x is the input vector and y is the target label. The input x is presented as $x = \{\mathbf{Z}, \Theta, \mathbf{R}, \mathbf{X}\}$ where $\mathbf{Z} = \{|Z|_1, |Z|_2, ...|Z|_m\}$, $\Theta = \{\theta_1, \theta_2, ...\theta_m\}$, $\mathbf{R} = \{R_1, R_2, ...R_m\}$, and $\mathbf{X} = \{X_1, X_2, ...X_m\}$ where $m = 25$. The target label $y \in \{1, 2, 3, 4, 5\}$ representing 5 classes for different insertions depths (5mm, 10mm, 15mm, 20mm, and 25mm respectively). The data is first standardized by subtracting the mean and dividing by the standard deviation. Then the machine learning models are trained and tested using 5 fold cross validation scheme. The four machine learning models are cross validated with certain hyper-parameters that are similar to the ones used in previous analysis.

Cross validated accuracies of four machine learning algorithms are given in Table 6.1 with their standard deviation. Support vector machine gave the highest accuracy with 98%

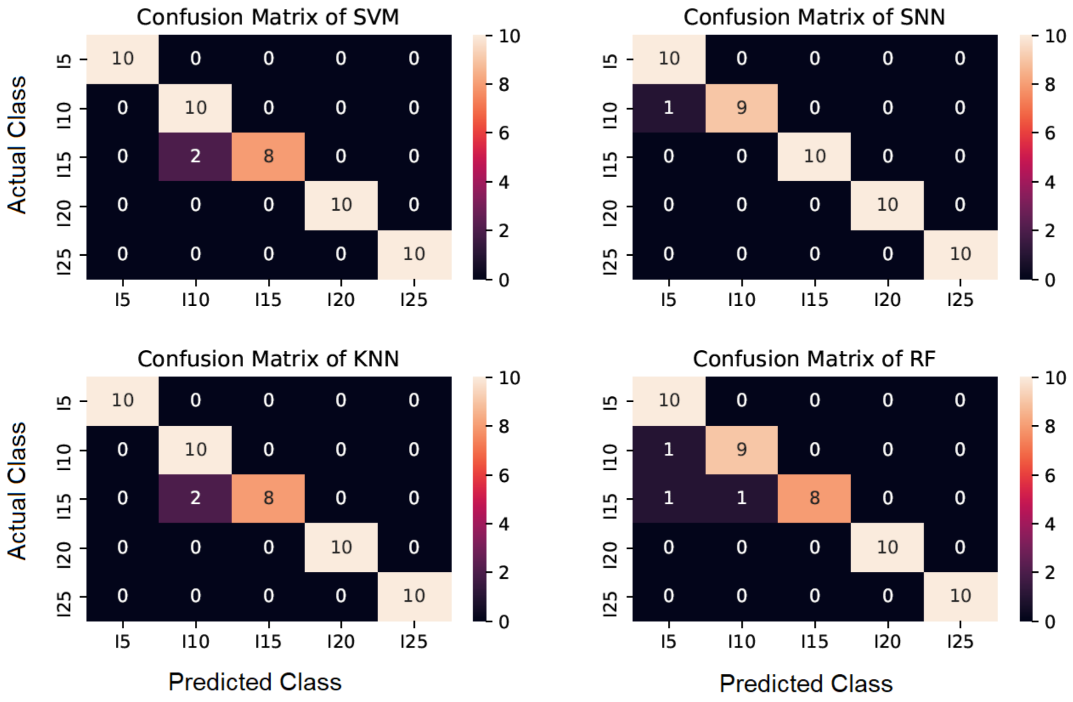
Table 6.1 Test accuracies of five class insertion depth dataset

EP	Accuracy % (SD)			
	SNN	SVM	KNN	RF
1	96.0(4.9)	98.0(4.0)	96.0(4.9)	92.0(9.8)

with SD of 4%, whereas Shallow neural network and k-nearest neighbours were equally good at giving the same accuracy of 96% with SD of 4.9%. Random forest algorithm not only gave the lowest accuracy of 92% but also having the highest SD of 9.8%. It is important to mention that insertion depths are estimated/classified using the most apical EP data. The same procedure can be repeated with other EPs data or can also use combination of EPs data.

Further analysis is carried out by looking in to the confusion matrix that gives us summary of the predictions of a classification problem. It gives information about the true positive, true negative, false positive and false negative predictions with count values and broken down by each class. Fig. 6.3 shows confusion matrix of all 4 learning models. In each matrix, x-labels and y-labels are representing the five classes. Confusion matrix for SVM is at top left corner, where we have only 2 incorrect predictions of insertion depth 15mm as 10mm insertion depth. All other insertion depth classes have 100% true positive rate. With SNN case on right top corner, only one incorrect prediction of 10mm insertion depth class is seen as 5mm insertion depth. All other classes have all true positive predictions. Confusion matrix of kNN (left down) is similar to the matrix of SVM. For random forest machine learning model, we can see three incorrect precitions; 10mm insertion depth is incorrectly predicted as 5mm and 15mm insertion is once incorrectly prediction as 10mm and once as 5mm. All other 3 classes have 100% true positive rate.

With confusion matrix analysis, it is evident that the insertion depths of 10mm and 15mm have incorrect predictions whereas the rest of classes have no incorrect predictions in all trained models. The reason may be explained that these insertion depths are at the crossroads of straight and curved insertion paths especially 10mm depth. Having said that, with such a small dataset, the models are trained well and predictions accuracies are above 95%.



	I5	I10	I15	I20	I25
I5	10	0	0	0	0
I10	0	10	0	0	0
I15	0	2	8	0	0
I20	0	0	0	10	0
I25	0	0	0	0	10

	I5	I10	I15	I20	I25
I5	10	0	0	0	0
I10	1	9	0	0	0
I15	0	0	10	0	0
I20	0	0	0	10	0
I25	0	0	0	0	10

	I5	I10	I15	I20	I25
I5	10	0	0	0	0
I10	0	10	0	0	0
I15	0	2	8	0	0
I20	0	0	0	10	0
I25	0	0	0	0	10

	I5	I10	I15	I20	I25
I5	10	0	0	0	0
I10	1	9	0	0	0
I15	1	1	8	0	0
I20	0	0	0	10	0
I25	0	0	0	0	10

Fig. 6.3 Confusion matrix of classification

6.2 Insertion Depth Estimation using Regression

In the regression problem, the input X is mapped to the output Y by the algorithm $M : X \rightarrow Y$ where X is an input sample set $X = \{x^{(1)}, x^{(2)}, ...x^{(m)}\}$ and corresponding output target set $Y = \{y^{(1)}, y^{(2)}, ...y^{(m)}\}$ where m in our dataset is 3425. One of the examples of input x corresponds to $\{|Z|, \theta, R, X\}$ having time sample each for impedance magnitude, phase, resistance and reactance. The output target set Y values range from 1-25 mm as shown in Fig. 6.4. In this, we have used our insertion trajectory dataset to cover dynamics of the ST model and different insertion paths. Linear regression algorithm is used to fit the data, however, after getting unsatisfactory results advanced convolution and recurrent neural networks are applied for this regression problem.

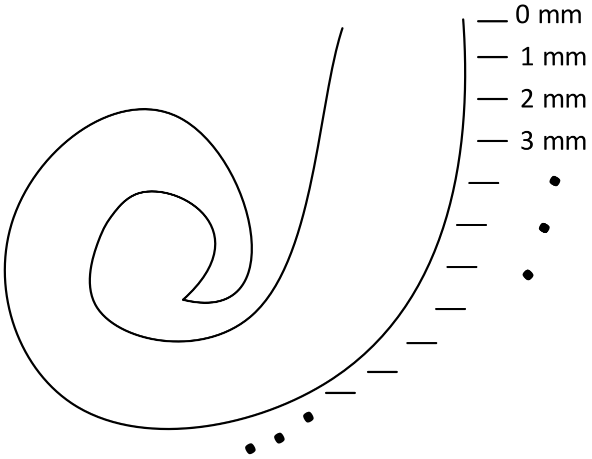


Fig. 6.4 Insertion depth blueprint of ST model

6.2.1 Performance Metrics

To access the performance of regression, there are different performance metrics than classification such as mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and r-squared value (R^2).

MAE is expressed as in (6.1) where y_i is the i^{th} ground truth numeric value and $\hat{y}_i$ is the i^{th} predicted numeric value. MAE is calculated by taking the average of the absolute prediction errors. Absolute is the mathematical function that converts any negative value to the positive value. Any negative errors will be forced to positive value while calculating MAE. Unlike MSE, MAE gives equal weightage to any error and does not penalize large errors, that is, it increases linearly.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6.1}$$

MSE is the most popular performance metric for regression and it is also used as a loss function in learning algorithms for regression as its been used in our model fitting/optimizing. MSE is calculated as the average of the squared differences of the ground truth target values and the predicted targets. The square operation removes any negative sign from difference value. The squared error value also contributes to penalizing the larger errors. It is expressed as in (6.2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6.2}$$

RMSE is just the square root of the MSE but it has significant in terms of the unit it has. MSE will have the squared units whereas RMSE has the same unit as the target/predicted value. Therefore, MSE is normally used as an optimization parameter for model training whereas RMSE gives better indication of the performance of regression model. It is given as in (6.3)

$$RMSE = \sqrt{MSE} \tag{6.3}$$

R-Squared(R^2) is also known as coefficient of determination and gives the indication of how close the predictive values are to the fitted regression line. Higher R^2 value indicates how good a model is to explain the variability of data around its mean. It is defined as the ratio of explained variance to the total variance and is represented mathematically as in (6.4)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{6.4}$$

where $\bar{y}_i$ is given by

$$\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6.5}$$

6.2.2 Linear Regression Results

Insertion depth estimation is first estimated with a basic multivariate regression model i.e., Linear Regression. There are four independent variables |Z|, θ, R and X and a dependent variable depth d. The estimated linear regression function can be represented as in 6.6

$$d = b_0 + b_1|Z| + b_2\theta + b_3R + b_4X \tag{6.6}$$

Linear regression model tries to determine the weights (b_0, b_1, b_2, b_3, b_4) such that the prediction d is as close as to its actual value. The linear regression model from Python's sklearn library is trained and test. The estimation function used by the linear regression models is ordinary least squares which minimizes the distance from the observed response to the predicted values. The data is split into 85/15 train test split. After fitting the model with the train data, model is used to predict insertion depth from the test data set. The performance metrics MAE, MSE, RMSE, and R^2 are calculated using the predicted and true values of insertion depth.

The performance metrics of the linear regression model is given in 6.2. The model predicted the test data set with mean absolute error of 4.54 mm, mean squared error and root mean squared value of 33.76 mm^2 and 5.81 mm respectively. The R^2 metric turned out to be 0.35. It is evident that errors are substantially high when total linear insertion depth is 25 mm. The model also happen to have very low R^2 value that shows the model was unable to fit the data. The reason could be that there is no linear relation between independent variable and dependent variable. The solution to improve the performance metrics is to use non-linear deep learning model as discussed in next section.

Table 6.2 Regression performance metrics using Linear Regression Model

EP	Model: Linear Regression			
	MAE	*MSE*	*RMSE*	R^2
1	4.54	33.76	5.81	0.35

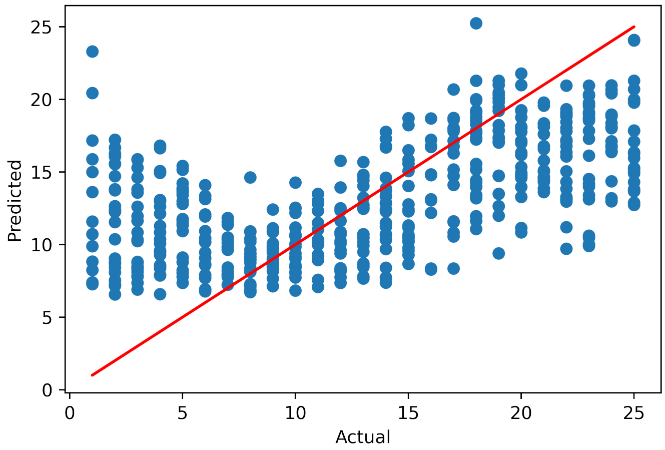


Fig. 6.5 Linear Regression with actual/predicted values distribution in the test set

6.2.3 CNN-LSTM Hybrid Model Results

A hybrid model of 1-dimensional convolutional neural network (CNN or Conv nets) and a variety of recurrent neural network (RNN) known as long short term memory (LSTM) network is used as a regression model. As shown in Fig. 6.6, a learning network is developed to train and test our insertion data to predict linear insertion depth using complex bipolar impedance features. There are five layers in this model; input layer, convolutional layer, LSTM layer, dense or full connected layer and finally the output layer. The convolution layer is responsible for spatial feature extraction whereas the LSTM layer is useful for temporal fearures. The input layer has all the data arranged according to the requirement of the subsequent convolutional layer. Since we are doing regression, each time sample in the dataset will have a target label according to the depth of the most apical electrode. EP1 data has been used in this analysis, however, the same process can be applied to other electrode pairs. At each time step, there are four recorded values |Z|,θ, R and X, and a vector is formed with these values as shown in input layer. The value of k is 3425 that represent the number of examples we have in the data set that has been created. The target values are set from 1-25 mm according to the depth to most apical electrode during the insertion process.

Once we have the input layer setup, there is 1-D convolutional layer. The reason for using 1-D CNN layer is its low conputational complexity compared to 2-D CNN and also 1-D CNN even with shallow architectures can learn from complex 1-D signals [97]. There are 12 1D filters of size 1x4. The selection of number of filters is carried out as hyperparamter tuning start from 4 filters. Each filter learns different features from the input 1-D signal. Size of the filter has been kept as the size of the input signal. The convolution layer produces the output (feature map) of the size 4x12 for each input training example. The convolution process for each kernel is carried out according to the Fig. 2.31. The feature map is fed in to the LSTM layer with 12 cells. The number of LSTM cells are selected according to the size of its input from the CNN layer. Each cell works as explained in Fig.2.32 and carry the information to the next LSTM cell. The LSTM layer produces an output of the size 1x12 that is fed into the fully connected layer with 25 neurons and onto a single neuron output layer. The output layer finally gives the insertion depth prediction. There are total 1281 trainable parameters in the model.

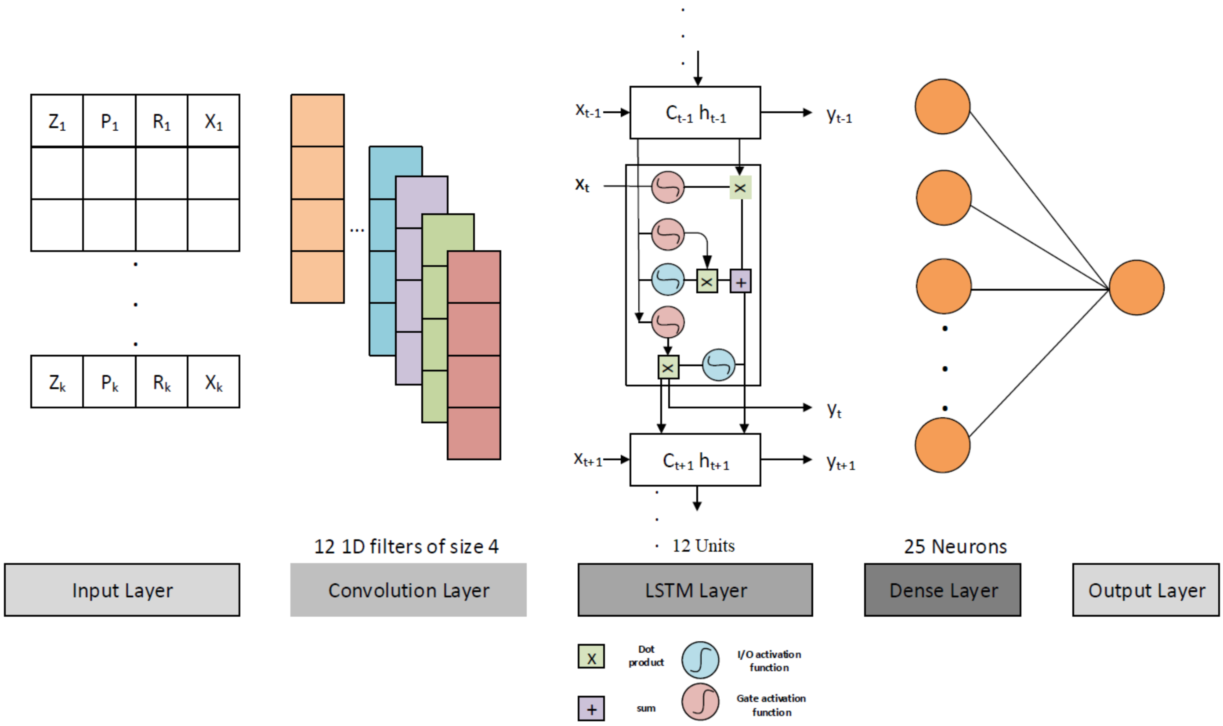


Fig. 6.6 Hybrid CNN-LSTM model for insertion depth prediction

6.2.3.1 Model Training

Learning model is developed in Python 3.6 using Keras library with tensorflow on the backend. The dataset is divided in to 70/15 ratio for training and validating the regression model. The rest of 15% dataset is set aside to be used as test set and will be unseen to the model during the training. The activation function for convolutional layer and dense layer is Rectified Linear Unit (ReLU) that supresses all negative values and its a linear function for values above 0. The activation function for LSTM layer is tanh and recurrent activation is sigmoid function.

The model is trained for 1000 epochs with the batch size of 30. Adaptive moment estimation (ADAM) optimizer is used with learning rate of 0.001 and mean squared error (MSE) as the loss function. To avoid overfitting, earlystopping has been used with patience value of 200. This helps to stop the training process if the model is overfitting. Fig. 6.7 shows the progress of model training where x-axis is number of epochs and y-axis is MSE value. It is evident that model is not overfitting as MSE of validation set is a little higher than the MSE of train set and there is not a big difference between those RMSE values.

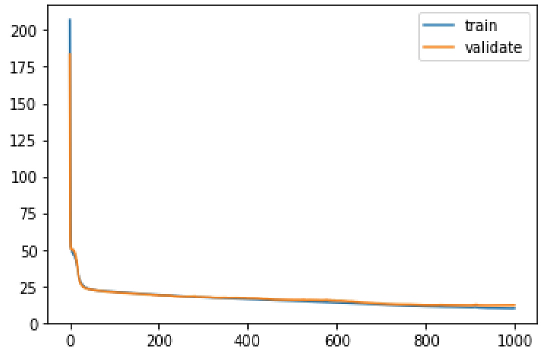


Fig. 6.7 CNN-LSTM hybrid model training for insertion depth regression

6.2.3.2 Performance Scores

Linear insertion depth estimation was carried out by a hybrid learning model. The model is trained and tested on a dataset collected in-vitro in a plastic ST model. Out of 3425 data examples, 2397 are used for training the model while the rest of 1028 are equally divided into validate and test set. The performance measures of our model is depicted in Table 6.3. The table shows performance metrics for both validate and test sets. The test set remained unseen to the model during the training process. Mean absolute error for the validate and test sets are 2.66 *mm* and 2.57 *mm* (10% variation) which indicates good model performance. Mean squared error is 11.93 mm^2 for the validation set and 11.32 mm^2 when depth is predicted on the test set. MSE is a bit high with respect to our data range of 1-25 mm. The reason for this high value is the inherit property of MSE to penalize large errors compared to small errors. Better performance measures of regression models are RMSE and R^2 value that turned out to be 3.45 *mm* and 0.76 (76%) respectively for the validation set. The same performance metrics are almost similar when insertion depth is predicted using the unseen test set.

Table 6.3 Regression performance metrics for validation dataset using CNN-LSTM Model

EP	Dataset	Model: CNN+LSTM			
		MAE	*MSE*	*RMSE*	R^2
1	Validate	2.66	11.93	3.45	0.76
1	Test	2.57	11.32	3.36	0.76

To get clearer visual representation of the predictive performance of the model, Fig. 6.8 presents fitted linear regression line. Actual targets are along x-axis and model predicted values are along y-axis. This graph also shows distribution of the actual and predicted insertion depths on the top and right side of it in green color. It can be seen, there is even distribution of our test set in terms of linear insertion depth 1-25mm data samples. If we see the distribution of the predicted depths, the performance is good in the beginning when the EA is inserted for 6-10 *mm* whereas prediction performance deteriorates in the middle of insertion depth, for example, from 12-20 mm depths. The reason for this bad performance could be the fact that it is the region when when EA is touching the lateral wall and sliding along with it. The insertion depth estimation is also not up to the mark at 1 mm. The reason for this maybe stabilization of impedance values in the beginning of the insertion.

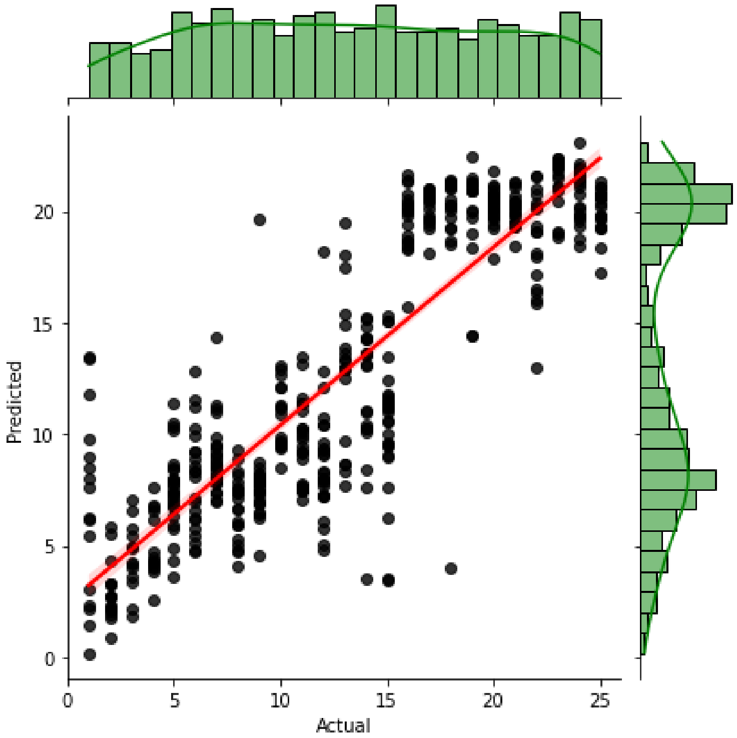


Fig. 6.8 Regression line with actual/predicted values distribution in the test set

In Fig. 6.9, randomly selected 25 examples of test set are taken and bar graph are plotted for their ground truth target values and estimated targets by the model. This graph better portray the difference in the estimation performance in terms of insertion depths. There are very few instances in this subset of data where real insertion depth has substantial difference with respect to the estimated or predicted insertion depth. For example, data example 3 has big difference between the two values i.e., 8 mm. Most of the data examples have nominal difference and few of them have very small difference, for example, data example number 5, 6, 11, 19 and 20.

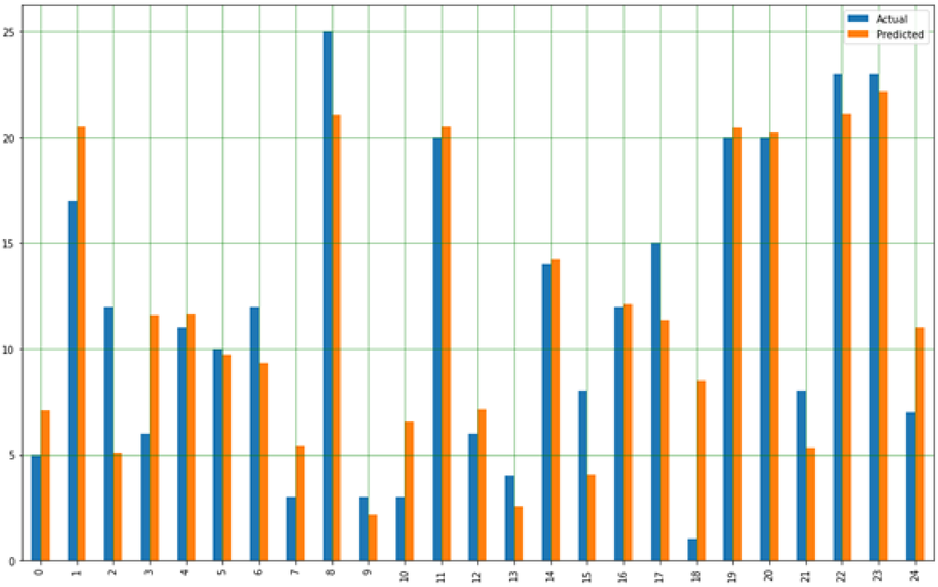


Fig. 6.9 Randomly selected 25 sample comparison of actual and predicted values of test set

6.3 Conclusion

It has been shown in this chapter that there is correlation between the complex impedance measurements and the linear insertion depth during electrode array insertion. For partial insertion depth estimation, multi-class (5) classification achieved results of above 95% accuracy when using SNN, SVM and KNN machine learning models. Whereas, for each mm time-step insertion depth estimation, a meager 0.35 R^2 value is achieved using linear regression model. For that reason an advances CNN-LSTM deep learning model is used to increase the prediction of mm-linear insertion depth. A relatively better 0.76 R^2 value is achieved partially due to the reason of non-linearity properties of the data at hand. It may not be possble to use such model in clinical practice, however, there is a room for improving the model performance by collecting more data and use data augmentation to generate synthetic data. With the results at hand, it is clear that complex impedance is not only beneficial for trajectory/path classification but also for insertion depth estimation. In the same way, this method can also be used for postoperative array management and intraoperative EA translocation. Although, angular insertion depth is mostly mentioned in the hearing research community, however, linear insertion depth is easily converted to angular insertion depth according to the ST size as mentioned in [136]. The analysis in this chapter is carried out on the insertion data of the most apical electrode pair but it can be replicated to other electrode pairs. Complex impedance data of multiple electrode pairs can also be used in a single analysis.

Chapter 7

Conclusion

7.1 Summary

The objective of this work was to develop a system for automated and atraumatic electrode array insertion in cochlear implantation. For that system, a complex impedance measurement system was developed and an actuation system with 3 degree of freedom was developed. To achieve the pursuit of atraumatic electrode array insertion, a data driven model was chosen to detect the trajectory and insertion depth of array during the insertion.

First of all, an impedance meter was developed that could not only measure bipolar impedance magnitude but also its phase, resistive and reactive parts. The reason going for complex impedance was the impedance equivalent circuit that defines the interaction of electrodes with the conductive solution and tissues in the cochlea. This impedance meter was designed for sequentially recording eight electrode pairs in bipolar stimulation setting during the insertion. Initial experiments showed that complex impedance components change when electrode is touching the ST walls.

Insertion trajectory is crucial for trauma-less insertion of the electrode array into the cochlea. It is also of vital importance that surgeons know of the EA directions once it's in the ST. This research focused on classifying EA insertion from three different angles/positions of the ST using machine learning algorithms. It's been successfully show-cased that it is possible to make trajectory prediction using complex impedance. It has been shown that high classification accuracy is achieved when all components of complex impedance are used and higher number of EPs are taken in to account. Learning models are developed with/without feature extraction and selection algorithm from the time series data. The models were trained and tested on both full insertion data and windowed/partial subsequence data.

Data driven machine learning models are trained and tested for trajectory (medial, lateral and middle) and path (straight or curved) predictions during the insertion process. Overall shallow neural network achieved the highest performance in terms of accuracy. However, in order to improve performance, an ensemble learning approach was adopted with majority voting scheme. Ensemble learning model was developed with three machine learning algorithms namely shallow neural network, k-nearest neighbours and random forest.

Ensemble model trained on subsequence data was employed to develop data driven feedback control loop. A complete robotic system was developed by integrating complex impedance sensing, 3 DoF actuation, machine learning algorithms and feedback control loop. This system was able to reasonably correctly predict the electrode array insertion trajectory (above 80% classification accuracy) and path (above 95% classification accuracy in partial insertion experiments and 0.85 R^2 value for mm time-step estimation) in online setting. This system has the potential to reduce the trauma induced during the implantation procedure.

Last but not least, an advance learning scheme was developed to estimate the insertion depth of electrode array during insertion. A hybrid model consisting of convolutional and recurrent neural network has been trained and tested on complex impedance data to estimate insertion depth both in classification (partial insertion depth estimation) and regression (mm time-step estimation) models.

This thesis offers a new methodology to measure and use complex impedance of electrodes in cochlear implant; for the localization of electrode array during the insertion process. It provides evidence of using machine learning algorithms for efficient prediction of different electrode array trajectories during the insertion. It also shows that a deep learning model can be used for linear insertion depth estimation that can be used on run-time. Overall, this thesis provides a framework for better electrode array insertion that may cause less trauma than the manual insertion process.

7.2 Research Limitations

The experiments are performed on scaled up 2D scala tympani model. However, tympanic membrane is a 3D structure, therefore further experiments need to be performed on 3D ST model. Having said that, only the first turn of the ST where EA experiences the most impact was considered. That turn is almost identical in 2D and 3D models. A 3D model was designed, however, failed to 3D print the model using the clear material due to non-availability of material in house and ordering to a non-professional manufacturer. The 2D used model was also scaled up 2:1 and I believe it is important to test the system for original size ST model.

As mentioned earlier, plastic model is used which is different from real cochlea in terms of internal structure. The plastic does not incorporate non-linearity introduced by the tissues in real animal or human cochlea. However, we believe that our system may perform similarly or better in real cochlea as medial wall has tissue structure and lateral wall has bony structure that may help better to discriminate the trajectory. This needs to be further investigated by first testing the system on animal cadaver and then on human cadaver to introduce the system to real environment of actual size and internal structure.

An important step to optimize the EA insertion process is the design of an automatic feeder system with an option of variable insertion speed adjustment. Such a system reduces the variability caused by the manual procedure in speed, tremor, and insertion duration. The proposed sensing system works best with such a robotic tool. The limitation of the feeder system we used is bulkiness and it might be difficult to attach to a robotic arm for easy maneuvering in the operating theatre. The second limitation is EA insertion duration due to the limitation of impedance meter sampling time. It is important to record at least 1 sample/EP/mm of insertion. However, lower insertion speed was associated with lower insertion trauma and better post-surgery hearing outcomes [11, 47].

It is understandable that all cochlear implants measure impedance differently (current stimulation first and then voltage measurement) than the proposed method and only measure impedance magnitude. Due to this reason, the system is not directly translational, however, we consider our system as a standalone robotic insertion platform with a novel sensing mechanism that requires access to the electrode contacts of CI to measure the proposed feature set during insertion to make a prediction.

One of the limitations of the proposed study is the dependence of bipolar impedance measurement on conductive fluid. Any measurement outside saline solution is a high impedance value and barely useful. There needs to be a mechanism for the automatic exclusion of such measurements from the feature set. There is another limitation in terms of variability in discriminative measurements due to different electrode arrays or measurement conditions which may limit the generalization of the machine learning algorithms. This limitation can be addressed by characterization and calibration of the implant prior to use. Lastly, the proposed work has used 3 trajectories which may not be representative of several other insertion possibilities. Further investigation needs to be done by incorporating additional trajectory variants, for example, similar to the work by Martins et al. [139] where they are accessing trauma when EA is inserted from different round window quadrants. Nonetheless, the proposed work is a step forward towards more accurate predicted implantation which may also have application in other electrode implants.

7.3 Future Directions

It has been demonstrated successfully in this work that complex impedance has merit as a sensing modality for atraumatic steering of electrode array. However, other sensing mechanism introduced in this domain such as force sensing and electrocochleography (EcochG) can be integrated in to a single unit along with complex impedance. Data generated by these sensing systems would carry more information about the EA insertion in real time than a single sensing system. This is what termed as data fusion where data from different sources (sensors) are combined to reduce uncertainties, improve reliability and increase the performance of the machine learning models [140]. It would also be interesting to correlate complex impedance with the insertion forces and train the ML models to stop the insertion when a certain threshold exceeds to avoid intraoperative damage.

As we have used classical machine learning algorithms for our trajectory prediction, there is need to collect more data and test deep learning models that are believed to have more generalization ability and high performance. However, these deep learning models are data hungry and they tend to overfit with few data examples. To record large datasets, we have to look at the reliability of the electrode array as well since it is not made for the purpose of insertion hundred and thousands of times to collect large amount of data. This problem could be overcome if a number of electrode arrays are available from the same manufacturer.

The impedance meter designed in this study has a very slow sampling rate due to limitations of the DAQ device used. Its sampling rate could be increased by using high sampling rate DAQ device to collect more data per unit time. NI DAQ 6211 has been used in our work that has sampling rate of 250 kS/s, NI DAQ 6341 or 6366 could be used with sample rate of 500 kS/s and 2 MS/s respectively to increase data collection per unit time. It is also possible to increase data collection rate by designing impedance meter using AD5933 integrated circuit interfaced with a microcontroller that has high sampling rate. More data recorded per unit time will help machine learning algorithms have higher performance.

This system is only tested on the electrode array supplied by Oticon Medical that is one of electrode types that they produce. Our proposed system should be further tested on different electrode types from different manufacturers such as from Cochlear Ltd, Med-el and advanced bionics.

We have proposed solution for the trajectory, path and insertion depth prediction of electrode array using bipolar complex impedance measurements. This system may be helpful in predicting insertion failures such as tip foldover (this problem is more pronounced in perimodiolar EAs), EA bucking, and EA translocation. Further experiments need to be performed to demonstrate its applicability to overcome such failures that not only degrade CI postoperative performance but could also lead to internal infections.

References

[1] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, “Medical robotics and computer-integrated surgery,” *Springer handbook of robotics*, pp. 1657–1684, 2016.

[2] M. Casper, R. Mitra, R. Khare, B. Jaramaz, B. Hamlin, B. McGinley, D. Mayman, J. Headrick, K. Urish, M. Gittins *et al.*, “Accuracy assessment of a novel image-free handheld robot for total knee arthroplasty in a cadaveric study,” *Computer Assisted Surgery*, vol. 23, no. 1, pp. 14–20, 2018.

[3] T. Ind, A. Laios, M. Hacking, and M. Nobbenhuis, “A comparison of operative outcomes between standard and robotic laparoscopic surgery for endometrial cancer: A systematic review and meta-analysis,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 4, p. e1851, 2017.

[4] F.-G. Zeng, S. Rebscher, W. Harrison, X. Sun, and H. Feng, “Cochlear implants: system design, integration, and evaluation,” *IEEE reviews in biomedical engineering*, vol. 1, pp. 115–142, 2008.

[5] D. Vickers, L. De Raeve, and J. Graham, “International survey of cochlear implant candidacy,” *Cochlear Implants International*, vol. 17, no. sup1, pp. 36–41, 2016.

[6] A. A. Eshraghi and T. R. Van De Water, “Cochlear implantation trauma and noise-induced hearing loss: Apoptosis and therapeutic strategies,” *The Anatomical Record Part A: Discoveries in Molecular, Cellular, and Evolutionary Biology: An Official Publication of the American Association of Anatomists*, vol. 288, no. 4, pp. 473–481, 2006.

[7] E. Hoskison, S. Mitchell, and C. Coulson, “Systematic review: radiological and histological evidence of cochlear implant insertion trauma in adult patients,” *Cochlear implants international*, vol. 18, no. 4, pp. 192–197, 2017.

[8] T. T. Abd El Aziz, L. El Fiky, M. H. Shalaby, and A. Essam, “Radiological evaluation of inner ear trauma after cochlear implant surgery by cone beam ct (cbct),” *European Archives of Oto-Rhino-Laryngology*, vol. 276, no. 10, pp. 2697–2703, 2019.

[9] S. Appachi, S. Schwartz, S. Ishman, and S. Anne, “Utility of intraoperative imaging in cochlear implantation: A systematic review,” *The Laryngoscope*, vol. 128, no. 8, pp. 1914–1921, 2018.

[10] T. S. Rau, A. Hussong, M. Leinung, T. Lenarz, and O. Majdani, “Automated insertion of preformed cochlear implant electrodes: evaluation of curling behaviour and insertion forces on an artificial cochlear model,” *International journal of computer assisted radiology and surgery*, vol. 5, no. 2, pp. 173–181, 2010.

[11] M. Miroir, Y. Nguyen, G. Kazmitcheff, E. Ferrary, O. Sterkers, and A. B. Grayeli, “Friction force measurement during cochlear implant insertion: application to a force-controlled insertion tool design,” *Otology & neurotology*, vol. 33, no. 6, pp. 1092–1100, 2012.

[12] C.-T. Tan, M. Svirsky, A. Anwar, S. Kumar, B. Caessens, P. Carter, C. Treaba, and J. T. Roland Jr, “Real-time measurement of electrode impedance during intracochlear electrode insertion,” *The Laryngoscope*, vol. 123, no. 4, pp. 1028–1032, 2013.

[13] C. K. Giardina, E. S. Krause, K. Koka, and D. C. Fitzpatrick, “Impedance measures during in vitro cochlear implantation predict array positioning,” *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 2, pp. 327–335, 2018.

[14] J. Pile, A. D. Sweeney, S. Kumar, N. Simaan, and G. B. Wanna, “Detection of modiolar proximity through bipolar impedance measurements,” *The Laryngoscope*, vol. 127, no. 6, pp. 1413–1419, 2017.

[15] J. R. Clark, L. Leon, F. M. Warren, and J. J. Abbott, “Magnetic guidance of cochlear implants: Proof-of-concept and initial feasibility study,” *Journal of Medical Devices*, vol. 6, no. 3, p. 035002, 2012.

[16] K. D. Wise, P. T. Bhatti, J. Wang, and C. R. Friedrich, “High-density cochlear implants with position sensing and control,” *Hearing research*, vol. 242, no. 1-2, pp. 22–30, 2008.

[17] A. N. Vadivelu, Z. Liu, D. S. Gunawardena, B. Chen, H.-Y. Tam, S. O’Leary, and D. Oetomo, “Integrated force sensor in a cochlear implant for hearing preservation surgery,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 3819–3822s.

[18] A. A. Eshraghi, R. Nazarian, F. F. Telischi, S. M. Rajguru, E. Truy, and C. Gupta, “The cochlear implant: historical aspects and future prospects,” *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology*, vol. 295, no. 11, pp. 1967–1980, 2012.

[19] A. J. Hudspeth, “How the ear’s works work,” *Nature*, vol. 341, no. 6241, p. 397, 1989.

[20] R. J. Krogmann and Y. Al Khalili, “Cochlear implants,” in *StatPearls [Internet]*. StatPearls Publishing, 2021.

[21] P. S. Roland and C. G. Wright, “Surgical aspects of cochlear implantation: mechanisms of insertional trauma,” in *Cochlear and Brainstem Implants*. Karger Publishers, 2006, vol. 64, pp. 11–30.

[22] S. J. Rebscher, A. Hetherington, B. Bonham, P. Wardrop, D. Whinney, and P. A. Leake, "Considerations for the design of future cochlear implant electrode arrays: Electrode array stiffness, size and depth of insertion," *Journal of rehabilitation research and development*, vol. 45, no. 5, p. 731, 2008.

[23] A. Dhanasingh and C. Jolly, "An overview of cochlear implant electrode array designs," *Hearing research*, vol. 356, pp. 93–103, 2017.

[24] P. Gibson and P. Boyd, "Optimal electrode design: straight versus perimodiolar," *European annals of otorhinolaryngology, head and neck diseases*, vol. 133, pp. S63–S65, 2016.

[25] S. M. Cords, G. Reuter, P. R. Issing, A. Sommer, J. Kuzma, and T. Lenarz, "A silastic positioner for a modiolus-hugging position of intracochlear electrodes: electrophysiologic effects," *Otology & Neurotology*, vol. 21, no. 2, pp. 212–217, 2000.

[26] C. Richard, J. N. Fayad, J. Doherty, and F. H. Linthicum Jr, "Round window versus cochleostomy technique in cochlear implantation: Histological findings," *Otology & neurotology: official publication of the American Otological Society, American Neurotology Society [and] European Academy of Otology and Neurotology*, vol. 33, no. 7, p. 1181, 2012.

[27] D. Schurzig, R. J. Webster III, M. S. Dietrich, and R. F. Labadie, "Force of cochlear implant electrode insertion performed by a robotic insertion tool: comparison of traditional versus advance off-stylet techniques," *Otology & neurotology: official publication of the American Otological Society, American Neurotology Society [and] European Academy of Otology and Neurotology*, vol. 31, no. 8, p. 1207, 2010.

[28] R. J. Briggs, M. Tykocinski, R. Lazsig, A. Aschendorff, T. Lenarz, T. Stöver, B. Fraysse, M. Marx, J. T. Roland, and P. S. Roland, "Development and evaluation of the modiolar research array–multi-centre collaborative study in human temporal bones," *Cochlear implants international*, vol. 12, no. 3, pp. 129–139, 2011.

[29] P. A. Leake and S. J. Rebscher, "Anatomical considerations and long-term effects of electrical stimulation," in *Cochlear implants: Auditory prostheses and electric hearing*. Springer, 2004, pp. 101–148.

[30] A. A. Eshraghi, N. W. Yang, and T. J. Balkany, "Comparative study of cochlear damage with three perimodiolar electrode designs," *The Laryngoscope*, vol. 113, no. 3, pp. 415–419, 2003.

[31] O. Majdani, T. S. Rau, S. Baron, H. Eilers, C. Baier, B. Heimann, T. Ortmaier, S. Bartling, T. Lenarz, and M. Leinung, "A robot-guided minimally invasive approach for cochlear implant surgery: preliminary results of a temporal bone study," *International journal of computer assisted radiology and surgery*, vol. 4, no. 5, pp. 475–486, 2009.

[32] L. B. Kratchman, G. S. Blachon, T. J. Withrow, R. Balachandran, R. F. Labadie, and R. J. Webster, "Design of a bone-attached parallel robot for percutaneous cochlear implantation," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 10, pp. 2904–2910, 2011.

[33] B. Bell, C. Stieger, N. Gerber, A. Arnold, C. Nauer, V. Hamacher, M. Kompis, L. Nolte, M. Caversaccio, and S. Weber, "A self-developed and constructed robot for minimally invasive cochlear implantation," *Acta oto-laryngologica*, vol. 132, no. 4, pp. 355–360, 2012.

[34] P. Brett, X. Du, M. Zoka-Assadi, C. Coulson, A. Reid, and D. Proops, "Feasibility study of a hand guided robotic drill for cochleostomy," *BioMed research international*, vol. 2014, 2014.

[35] T. Klenzner, C. C. Ngan, F. B. Knapp, H. Knoop, J. Kromeier, A. Aschendorff, E. Papastathopoulos, J. Raczkowsky, H. Wörn, and J. Schipper, "New strategies for high precision surgery of the temporal bone using a robotic approach for cochlear implantation," *European archives of oto-rhino-laryngology*, vol. 266, no. 7, p. 955, 2009.

[36] J. T. Roland Jr, "A model for cochlear implant electrode insertion and force evaluation: results with a new electrode design and insertion technique," *The Laryngoscope*, vol. 115, no. 8, pp. 1325–1339, 2005.

[37] J. Zhang, K. Xu, N. Simaan, and S. Manolidis, "A pilot study of robot-assisted cochlear implant surgery using steerable electrode arrays," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2006, pp. 33–40.

[38] J. Zhang, J. T. Roland, S. Manolidis, and N. Simaan, "Optimal path planning for robotic insertion of steerable electrode arrays in cochlear implant surgery," *Journal of medical devices*, vol. 3, no. 1, p. 011001, 2009.

[39] J. Zhang, W. Wei, J. Ding, J. T. Roland Jr, S. Manolidis, and N. Simaan, "Inroads toward robot-assisted cochlear implant surgery using steerable electrode arrays," *Otology & neurotology*, vol. 31, no. 8, pp. 1199–1206, 2010.

[40] A. Hussong, T. S. Rau, T. Ortmaier, B. Heimann, T. Lenarz, and O. Majdani, "An automated insertion tool for cochlear implants: another step towards atraumatic cochlear implant surgery," *International journal of computer assisted radiology and surgery*, vol. 5, no. 2, pp. 163–171, 2010.

[41] D. Schurzig, R. Labadie, A. Hussong, T. Rau, and R. Webster III, "A second generation automated cochlear implant insertion tool with integrated force sensing," *IEEE Transactions on Mechatronics (In Review)*, 2010.

[42] J. Pile and N. Simaan, "Modeling, design, and evaluation of a parallel robot for cochlear implant surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1746–1755, 2014.

[43] T. L. Bruns and R. J. Webster III, "An image guidance system for positioning robotic cochlear implant insertion tools," in *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 10135. SPIE, 2017, pp. 199–204.

[44] O. Adunka, J. Kiefer, M. H. Unkelbach, T. Lehnert, and W. Gstoettner, "Development and evaluation of an improved cochlear implant electrode design for electric acoustic stimulation," *The Laryngoscope*, vol. 114, no. 7, pp. 1237–1241, 2004.

[45] Y. Nguyen, M. Miroir, G. Kazmitcheff, J. Sutter, M. Bensidhoum, E. Ferrary, O. Sterkers, and A. B. Grayeli, "Cochlear implant insertion forces in microdissected human cochlea to evaluate a prototype array," *Audiology and Neurotology*, vol. 17, no. 5, pp. 290–298, 2012.

[46] C. A. Todd, F. Naghdy, and M. J. Svehla, "Force application during cochlear implant insertion: an analysis for improvement of surgeon technique," *IEEE transactions on biomedical engineering*, vol. 54, no. 7, pp. 1247–1255, 2007.

[47] G. Kontorinis, T. Lenarz, T. Stöver, and G. Paasche, "Impact of the insertion speed of cochlear implant electrodes on the insertion forces," *Otology & neurotology*, vol. 32, no. 4, pp. 565–570, 2011.

[48] T. S. Rau, S. Hügl, T. Lenarz, and O. Majdani, "On the benefit of ultra-slow insertion speed: reduced insertion forces in cochlear implantation surgery," *Laryngo-Rhino-Otologie*, vol. 97, no. S 02, p. 10544, 2018.

[49] J. Zhang, S. Bhattacharyya, and N. Simaan, "Model and parameter identification of friction during robotic insertion of cochlear-implant electrode arrays," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, 2009, pp. 3859–3864.

[50] O. Majdani, D. Schurzig, A. Hussong, T. Rau, J. Wittkopf, T. Lenarz, and R. F. Labadie, "Force measurement of insertion of cochlear implant electrode arrays in vitro: comparison of surgeon to automated insertion tool," *Acta oto-laryngologica*, vol. 130, no. 1, pp. 31–36, 2010.

[51] D. De Seta, R. Torres, F. Y. Russo, E. Ferrary, G. Kazmitcheff, D. Heymann, J. Amiaud, O. Sterkers, D. Bernardeschi, and Y. Nguyen, "Damage to inner ear structure during cochlear implantation: Correlation between insertion force and radio-histological findings in temporal bone specimens," *Hearing research*, vol. 344, pp. 90–97, 2017.

[52] P. Mittmann, A. Ernst, and I. Todt, "Electrode design and insertional depth-dependent intra-cochlear pressure changes: a model experiment," *The Journal of Laryngology & Otology*, vol. 132, no. 3, pp. 224–229, 2018.

[53] S. A. Wade, J. B. Fallon, A. K. Wise, R. K. Shepherd, N. L. James, and P. R. Stoddart, "Measurement of forces at the tip of a cochlear implant during insertion," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1177–1186, 2014.

[54] J. Pile, G. B. Wanna, and N. Simaan, "Robot-assisted perception augmentation for online detection of insertion failure during cochlear implant surgery," *Robotica*, vol. 35, no. 7, pp. 1598–1615, 2017.

[55] Z. Zhu, Q. Tang, F.-G. Zeng, T. Guan, and D. Ye, "Cochlear-implant spatial selectivity with monopolar, bipolar and tripolar stimulation," *Hearing research*, vol. 283, no. 1-2, pp. 45–58, 2012.

[56] M. J. Osberger and L. Fisher, "Sas-cis preference study in postlingually deafened adults implanted with the clarion® cochlear implant," *Annals of Otology, Rhinology & Laryngology*, vol. 108, no. 4_suppl, pp. 74–79, 1999.

[57] R. Y. Litovsky, M. J. Goupell, A. Kan, and D. M. Landsberger, “Use of research interfaces for psychophysical studies with cochlear-implant users,” *Trends in hearing*, vol. 21, p. 2331216517736464, 2017.

[58] A. Volta and J. Banks, “I. on the electricity excited by the mere contact of conducting substances of different kinds,” *The Philosophical Magazine*, vol. 7, no. 28, pp. 289–311, 1800.

[59] G. S. Ohm, *Bestimmung des Gesetzes, nach welchem Metalle die Contaktelektricität leiten: Nebst einem Entwurfe zur einer Theorie des Voltaischen Apparates und des Schweiggerschen Multiplicators*. Journal für Chemie und Physik, 1826.

[60] M. Faraday, “Experimental researches in electricity. seventh series,” *Philosophical Transactions of the Royal Society of London*, vol. 124, pp. 77–122, 1834.

[61] C. F. Varley, “Vii. polarization of metallic surfaces in aqueous solutions. on a new method of obtaining electricity from mechanical force, and certain relations between electrostatic induction and the decomposition of water,” *Philosophical Transactions of the Royal Society of London*, vol. 161, pp. 129–136, 1871.

[62] H. V. Helmholtz, “Studien über electrische grenzschichten,” *Annalen der Physik*, vol. 243, no. 7, pp. 337–382, 1879.

[63] E. Warburg, “Ueber das verhalten sogenannter unpolarisirbarer elektroden gegen wechselstrom,” *Annalen der Physik*, vol. 303, no. 3, pp. 493–499, 1899.

[64] H. Fricke, “Xxxiii. the theory of electrolytic polarization,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 14, no. 90, pp. 310–318, 1932.

[65] J. E. B. Randles, “Kinetics of rapid electrode reactions,” *Discussions of the faraday society*, vol. 1, pp. 11–19, 1947.

[66] M. Sluyters-Rehbach and J. Sluyters, *Sine Wave Methods in the Study of Electrode Processes*, 1970. [Online]. Available: https://books.google.co.uk/books?id=_fsMtwAACAAJ

[67] L. Geddes and L. Baker, *Principles of Applied Biomedical Instrumentation*, ser. A Wiley-Interscience publication. Wiley, 1989. [Online]. Available: https://books.google.co.uk/books?id=lDVrAAAAMAAJ

[68] T. Carr, A. Holt, and P. Katz, “Nonlinear aspects of the bioelectrode-electrolyte interface,” *Biomedical Electrode Technology: Theory and Practice*, p. 143, 1974.

[69] L. Geddes and R. Roeder, “Measurement of the direct-current (faradic) resistance of the electrode-electrolyte interface for commonly used electrode materials,” *Annals of biomedical engineering*, vol. 29, no. 2, pp. 181–186, 2001.

[70] M. Suesserman and F. Spelman, “Lumped-parameter model for in vivo cochlear stimulation,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 3, pp. 237–245, 1993.

[71] F. J. Vanpoucke, A. J. Zarowski, and S. A. Peeters, "Identification of the impedance model of an implanted cochlear prosthesis from intracochlear potential measurements," *IEEE Transactions on biomedical engineering*, vol. 51, no. 12, pp. 2174–2183, 2004.

[72] M. Tykocinski, L. T. Cohen, and R. S. Cowan, "Measurement and analysis of access resistance and polarization impedance in cochlear implant recipients," *Otology & Neurotology*, vol. 26, no. 5, pp. 948–956, 2005.

[73] R. T. Karunasiri, "Systems and methods for measuring electrode impedance during a normal operation of a cochlear implant system," Jun. 27 2017, uS Patent 9,687,651.

[74] J. Anso, T. W. Balmer, Y. Jegge, H. Kalvoy, B. J. Bell, C. Dür, E. M. Calvo, T. M. Williamson, N. Gerber, D. Ferrario *et al.*, "Electrical impedance to assess facial nerve proximity during robotic cochlear implantation," *IEEE transactions on biomedical engineering*, vol. 66, no. 1, pp. 237–245, 2018.

[75] T. M. Mitchell, "Artificial neural networks," *Machine learning*, vol. 45, pp. 81–127, 1997.

[76] S.-s. Liu and Y.-t. Tian, "Facial expression recognition method based on gabor wavelet features and fractional power polynomial kernel pca," in *International Symposium on Neural Networks*. Springer, 2010, pp. 144–151.

[77] V. Mitra, H. Franco, R. M. Stern, J. Van Hout, L. Ferrer, M. Graciarena, W. Wang, D. Vergyri, A. Alwan, and J. H. Hansen, "Robust features in deep-learning-based speech recognition," in *New Era for Robust Speech Recognition*. Springer, 2017, pp. 187–217.

[78] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, pp. 1–36, 2019.

[79] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[80] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[81] V. N. Vapnik, "The nature of statistical learning theory," *New York: Springer2V erlag*, 1995.

[82] V. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[83] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[84] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[85] P. Schiilkop, C. Burgest, and V. Vapnik, "Extracting support data for a given task," in *Proceedings of the 1st international conference on knowledge discovery & data mining*, 1995, pp. 252–257.

[86] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[87] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of potential function method in pattern recognition," *Automation and Remote Control*, vol. 25, no. 6, pp. 917–936, 1964.

[88] V. Franc and V. Hlavác, "Multi-class support vector machine," in *2002 International Conference on Pattern Recognition*, vol. 2. IEEE, 2002, pp. 236–239.

[89] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[90] Z. Geler, V. Kurbalija, M. Radovanović, and M. Ivanović, "Comparison of different weighting schemes for the k nn classifier on time-series data," *Knowledge and Information Systems*, vol. 48, no. 2, pp. 331–378, 2016.

[91] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers-a tutorial," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–25, 2021.

[92] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[93] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[94] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[95] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[96] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[97] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.

[98] N. Hafeez, X. Du, N. Boulgouris, P. Begg, R. Irving, C. Coulsan, and G. Tourrels, "Towards unblinding the surgeons: Complex electrical impedance for electrode array insertion guidance in cochlear implantation," in *Proceedings of the International Symposium on Auditory and Audiological Research*, vol. 7, 2019, pp. 29–36.

[99] R. C. Weast, M. J. Astle, W. H. Beyer *et al.*, *CRC handbook of chemistry and physics*. CRC press, Boca raton FL, 1989, vol. 1990.

[100] A. Wolf and U. S. P. H. Service, *Aqueous Solutions and Body Fluids: Their Concentrative Properties and Conversion Tables*. Hoeber Medical Division, Harper & Row, 1966. [Online]. Available: https://books.google.co.uk/books?id=j_5qAAAAMAAJ

[101] J. Law and R. Rennie, *A dictionary of physics*. OUP Oxford, 2015.

[102] A. Ruiz-Vargas, A. Ivorra, and J. W. Arkwright, "Design, construction and validation of an electrical impedance probe with contact force and temperature sensors suitable for in-vivo measurements," *Scientific reports*, vol. 8, no. 1, pp. 1–11, 2018.

[103] N. Instrument, "Resistance measurement with multifunction daq-national instruments," https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000004ABbSAM&l=en-GB, 2020, accessed: 2021-04-19.

[104] W. Wimmer, F. Venail, T. Williamson, M. Akkari, N. Gerber, S. Weber, M. Caversaccio, A. Uziel, and B. Bell, "Semiautomatic cochleostomy target and insertion trajectory planning for minimally invasive cochlear implantation," *BioMed research international*, vol. 2014, 2014.

[105] Z. Wang, J. Li, Y. Wu, R. Zhu, B. Wang, and K. Zhao, "Optimal path generation in scala tympani and path planning for robotic cochlear implant of perimodiolar electrode," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 234, no. 6, pp. 578–589, 2020.

[106] M. W. Skinner, T. A. Holden, B. R. Whiting, A. H. Voie, B. Brunsden, J. G. Neely, E. A. Saxon, T. E. Hullar, and C. C. Finley, "In vivo estimates of the position of advanced bionics electrode arrays in the human cochlea," *Annals of Otology, Rhinology & Laryngology*, vol. 116, no. 4_suppl, pp. 2–24, 2007.

[107] A. Aschendorff, J. Kromeier, T. Klenzner, and R. Laszig, "Quality control after insertion of the nucleus contour and contour advance electrode in adults," *Ear and hearing*, vol. 28, no. 2, pp. 75S–79S, 2007.

[108] N. Hafeez, X. Du, N. Boulgouris, P. Begg, R. Irving, C. Coulson, and G. Tourrel, "Electrical impedance guides electrode array in cochlear implantation using machine learning and robotic feeder," *Hearing research*, vol. 412, p. 108371, 2021.

[109] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015, vol. 72.

[110] M. Waskom, O. Botvinnik, J. Ostblom, M. Gelbart, S. Lukauskas, P. Hobson, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole *et al.*, "mwaskom/seaborn: v0.10.1 (april 2020)," *zenodo*, 2020.

[111] C. V. G. Zelaya, "Towards explaining the effects of data preprocessing on machine learning," in *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 2019, pp. 2086–2090.

[112] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[113] M. Chris, "Overview on extracted features with TSFRESH," https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html, 2016, [Online; accessed 23-August-2021].

[114] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzande, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104210, 2021.

[115] V. Topsakal, M. Matulic, M. Z. Assadi, G. Mertens, V. Van Rompaey, and P. Van de Heyning, "Comparison of the surgical techniques and robotic techniques for cochlear implantation in terms of the trajectories toward the inner ear," *The journal of international advanced otology*, vol. 16, no. 1, p. 3, 2020.

[116] W. G. Morrel, K. E. Riojas, R. J. Webster, J. H. Noble, and R. F. Labadie, "Custom mastoid-fitting templates to improve cochlear implant electrode insertion trajectory," *International journal of computer assisted radiology and surgery*, vol. 15, pp. 1713–1718, 2020.

[117] R. Torres, G. Kazmitcheff, D. De Seta, E. Ferrary, O. Sterkers, and Y. Nguyen, "Improvement of the insertion axis for cochlear implantation with a robot-based system," *European Archives of Oto-Rhino-Laryngology*, vol. 274, no. 2, pp. 715–721, 2017.

[118] Torres *et al.*, "An Optimized Robot-Based Technique for Cochlear Implantation to Reduce Array Insertion Trauma," *Otolaryngology - Head and Neck Surgery (United States)*, vol. 159, no. 5, pp. 900–907, 11 2018.

[119] R. Torres, G. Kazmitcheff, D. Bernardeschi, D. De Seta, J. L. Bensimon, E. Ferrary, O. Sterkers, and Y. Nguyen, "Variability of the mental representation of the cochlear anatomy during cochlear implantation," *European Archives of Oto-Rhino-Laryngology*, vol. 273, no. 8, pp. 2009–2018, 2016.

[120] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[121] G. Baggio, D. S. Bassett, and F. Pasqualetti, "Data-driven control of complex networks," *Nature communications*, vol. 12, no. 1, pp. 1–13, 2021.

[122] H. A. Poonawala and U. Topcu, "Robustness of classifier-in-the-loop control systems: A hybrid-systems approach," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2738–2743.

[123] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, and G. Di Caro, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[124] Z.-H. Zhou, "Ensemble learning," in *Machine Learning*. Springer, 2021, pp. 181–210.

[125] R. Kloeckner, D. P. dos Santos, J. Schneider, L. Kara, C. Dueber, and M. B. Pitton, "Radiation exposure in ct-guided interventions," *European journal of radiology*, vol. 82, no. 12, pp. 2253–2257, 2013.

[126] M. M. Khan, R. F. Labadie, and J. H. Noble, "Preoperative prediction of angular insertion depth of lateral wall cochlear implant electrode arrays," *Journal of Medical Imaging*, vol. 7, no. 3, p. 031504, 2020.

[127] G. Mertens, V. Van Rompaey, P. Van de Heyning, E. Gorris, and V. Topsakal, "Prediction of the cochlear implant electrode insertion depth: clinical applicability of two analytical cochlear models," *Scientific reports*, vol. 10, no. 1, pp. 1–10, 2020.

[128] C. Rathgeb, M. Demattè, A. Yacoub, L. Anschuetz, F. Wagner, G. Mantokoudis, M. Caversaccio, and W. Wimmer, "Clinical applicability of a preoperative angular insertion depth prediction method for cochlear implantation," *Otology & neurotology*, vol. 40, no. 8, pp. 1011–1017, 2019.

[129] K. S. van der Marel, J. J. Briaire, R. Wolterbeek, B. M. Verbist, and J. H. Frijns, "Development of insertion models predicting cochlear implant electrode position," *Ear and hearing*, vol. 37, no. 4, pp. 473–482, 2016.

[130] S. Chen, F. Wang, Y. Lin, Q. Shi, and Y. Wang, "Ultrasound-guided needle insertion robotic system for percutaneous puncture," *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, no. 3, pp. 475–484, 2021.

[131] J. Chevrie, N. Shahriari, M. Babel, A. Krupa, and S. Misra, "Flexible needle steering in moving biological tissue with motion compensation using ultrasound and force feedback," *IEEE Robotics and automation letters*, vol. 3, no. 3, pp. 2338–2345, 2018.

[132] T. D. Yang, K. Park, H.-J. Kim, N.-R. Im, B. Kim, T. Kim, S. Seo, J.-S. Lee, B.-M. Kim, Y. Choi *et al.*, "In vivo photothermal treatment with real-time monitoring by optical fiber-needle array," *Biomedical optics express*, vol. 8, no. 7, pp. 3482–3492, 2017.

[133] K. Koka, W. J. Riggs, R. Dwyer, J. T. Holder, J. H. Noble, B. M. Dawant, A. Ortmann, C. Valenzuela, J. K. Mattingly, M. Harris *et al.*, "Intra-cochlear electrocochleography during cochear implant electrode insertion is predictive of final scalar location," *Otology & neurotology: official publication of the American Otological Society, American Neurotology Society [and] European Academy of Otology and Neurotology*, vol. 39, no. 8, p. e654, 2018.

[134] M. S. Harris, W. J. Riggs, C. K. Giardina, B. P. O'Connell, J. T. Holder, R. T. Dwyer, K. Koka, R. F. Labadie, D. C. Fitzpatrick, and O. F. Adunka, "Patterns seen during electrode insertion using intracochlear electrocochleography obtained directly through a cochlear implant," *Otology & neurotology: official publication of the American Otological Society, American Neurotology Society [and] European Academy of Otology and Neurotology*, vol. 38, no. 10, p. 1415, 2017.

[135] C. K. Giardina, T. E. Khan, S. H. Pulver, O. F. Adunka, C. A. Buchman, K. D. Brown, H. C. Pillsbury, and D. C. Fitzpatrick, "Response changes during insertion of a cochlear implant using extracochlear electrocochleography," *Ear and hearing*, vol. 39, no. 6, p. 1146, 2018.

[136] P. Aebischer, S. Meyer, M. Caversaccio, and W. Wimmer, “Intraoperative impedance-based estimation of cochlear implant electrode array insertion depth,” *IEEE transactions on biomedical engineering*, vol. 68, no. 2, pp. 545–555, 2020.

[137] C. A. Buchman, M. T. Dillon, E. R. King, M. C. Adunka, O. F. Adunka, and H. C. Pillsbury, “Influence of cochlear implant insertion depth on performance: a prospective randomized trial,” *Otology & Neurotology*, vol. 35, no. 10, pp. 1773–1779, 2014.

[138] B. P. O’Connell, J. B. Hunter, D. S. Haynes, J. T. Holder, M. M. Dedmon, J. H. Noble, B. M. Dawant, and G. B. Wanna, “Insertion depth impacts speech perception and hearing preservation for lateral wall electrodes,” *The Laryngoscope*, vol. 127, no. 10, pp. 2352–2357, 2017.

[139] G. d. S. Q. Martins, R. V. Brito Neto, R. K. Tsuji, E. M. M. S. Gebrim, and R. F. Bento, “Evaluation of intracochlear trauma caused by insertion of cochlear implant electrode arrays through different quadrants of the round window,” *BioMed research international*, vol. 2015, 2015.

[140] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang, and Z. Zhao, “An overview of data fusion techniques for internet of things enabled physical activity recognition and measure,” *Information Fusion*, vol. 55, pp. 269–280, 2020.

Appendix A

MATLAB Code

Actuation System

1. Load Driver function

```
function Controller= DriverObject()
disp('*****************Driver Object is ...
    Loading*********************');
addpath ( 'C:\Users\Public\PI\PI_MATLAB_Driver_GCS2' );
if ( ¬exist ( 'Controller', 'var' ) || ¬isa ( Controller, ...
    'PI_GCS_Controller' ) )
    Controller = PI_GCS_Controller ();
end;
end
```

2. Controller Object function

```
function [Controller1,Controller2,Controller3, ...
    PIdevice]=ControllerObject(Controller)

disp('*********************Controller Objects are ...
    initialising****************');

boolPIdeviceConnected = false; if ( exist ( 'PIdevice', 'var' ...
    ) ), if ( PIdevice.IsConnected ), boolPIdeviceConnected = ...
    true; end; end;
```

```
if ( ¬(boolPIdeviceConnected ) )


    controllerSerialNumber = '0135500161';
    PIdevice=Controller.OpenUSBDaisyChain(controllerSerialNumber);
    controller1_HardwareAddress=1;
    controller2_HardwareAddress=2;
    controller3_HardwareAddress=3;
    Controller1=...
    PIdevice.ConnectDaisyChainDevice(controller1_HardwareAddress);
    Controller2=...
    PIdevice.ConnectDaisyChainDevice(controller2_HardwareAddress);
    Controller3=...
    PIdevice.ConnectDaisyChainDevice(controller3_HardwareAddress);

end

% Query controller identification string
connectedControllerName1 = Controller1.qIDN();
connectedControllerName2 = Controller2.qIDN();
connectedControllerName3 = Controller3.qIDN();

% initialize PIdevice object for use in MATLAB
Controller1 = Controller1.InitializeController ();
Controller2 = Controller2.InitializeController ();
Controller3 = Controller3.InitializeController ();

%% Show connected stages
disp('******************Connected ...
   Stages*************************');
% query names of all controller axes
availableAxes1 = Controller1.qSAI_ALL
availableAxes2 = Controller2.qSAI_ALL
availableAxes3 = Controller3.qSAI_ALL
% get Name of the stage connected to axis 1
stage1=Controller1.qCST ( '1' )
stage2=Controller2.qCST ( '1' )
stage3=Controller3.qCST ( '1' )
% Show for all axes: which stage is connected to which ...
   axis-Controller1
for idx1 = 1 : length ( availableAxes1 )
    % qCST gets the name of the
    stageName1 = Controller1.qCST ( availableAxes1 { idx1 } );
    disp ( [ 'Axis ', availableAxes1{idx1}, ': ', stageName1 ...
        ] );
end

% Show for all axes: which stage is connected to which ...
    axis-Controller2
for idx2 = 1 : length ( availableAxes2 )
    % qCST gets the name of the
    stageName2 = Controller2.qCST ( availableAxes2 { idx2 } );
    disp ( [ 'Axis ', availableAxes2{idx2}, ': ', stageName2 ...
        ] );
end

% Show for all axes: which stage is connected to which ...
    axis-Controller3
for idx3 = 1 : length ( availableAxes3 )
    % qCST gets the name of the
    stageName3 = Controller3.qCST ( availableAxes3 { idx3 } );
    disp ( [ 'Axis ', availableAxes3{idx3}, ': ', stageName3 ...
        ] );
end
end
```

3. Stages Referencing function

```

function [axis1, axis2, ...
    axis3,positionReached1,positionReached2,positionReached3]...
=Referencing(Controller1, Controller2, Controller3)
disp('*********************Referencing of ...
    Stages*************************');
axis1 = '1';
axis2= '1';
axis3= '1';
% switch servo on for axis
switchOn    = 1;
Controller1.SVO ( axis1, switchOn );
Controller2.SVO ( axis2, switchOn );
Controller3.SVO ( axis3, switchOn );
% reference axis
Controller1.FRF ( axis1 );  % find reference
```

```
disp ( 'Stage-1 is referencing')
% wait for referencing to finish
while(0 ≠ Controller1.qFRF ( axis1 ) == 0 )
    pause(0.1);
    fprintf('.');
end
fprintf('\n');
Controller2.FRF( axis2);
disp ( 'Stage-2 is referencing')
while(0 ≠ Controller2.qFRF ( axis2 ) == 0 )
    pause(0.1);
    fprintf('.');
end
fprintf('\n');
Controller3.FRF( axis2);
disp ( 'Stage-3 is referencing')
while(0 ≠ Controller3.qFRF ( axis3 ) == 0 )
    pause(0.1);
    fprintf('.');
end
fprintf('\n');
positionReached1 = Controller1.qPOS(axis1);
positionReached2 = Controller2.qPOS(axis2);
positionReached3 = Controller3.qPOS(axis3);
end
```

4. Move X function

```
function [positionReached1]=moveX(Controller1,axis1,xaxis,vel)
%% Move Stage
disp('**********************Stage ...
    Moving***********************');
%SET Velocity
Controller1.VEL(axis1, vel);
positionReached1 = Controller1.qPOS(axis1);
commandedPosition1=positionReached1+xaxis;
Controller1.MOV ( axis1, commandedPosition1 );
disp ( 'Stage-1 is moving | x-axis')

while(0 ≠ Controller1.IsMoving ( axis1 ) )
      pause ( 0.1 );
      fprintf('.');
end
positionReached1 = Controller1.qPOS(axis1);
end
% similar functions for y-axis and rotational axis by ...
    changing the controller and axis objects
```

5. DAQ Initialization

```
function  [s s1 s2 s3]=DAQ_Init(duration)
daqreset
% measurement settings
fsout = 250000;
fsin = 125000;
%  time vector
xlen = round(duration*fsout);
t = (0:xlen-1)/fsout;
s=daq.createSession('ni');
s1 = daq.createSession('ni');
s2 = daq.createSession('ni');
s3 = daq.createSession('ni');
outCh=addAnalogOutputChannel(s,'Dev2',0, 'voltage'); % DAQ6011
s.Rate=fsout;
inCh=addAnalogInputChannel(s,'Dev2',0:1, 'voltage'); % DAQ6211
addDigitalChannel(s1,'Dev2','port1/line0:2','OutputOnly');
addDigitalChannel(s2,'Dev1','port0/line0:2','OutputOnly');
outCh=addAnalogOutputChannel(s3,'Dev2',1, 'voltage'); % DAQ6011
inCh(1).TerminalConfig = 'Differential';
inCh(1).Range = [-10 10];
vee=-5;
outputSingleScan(s3,vee);
end
```

6. Insertion alongwith complex impedance measurement and live plotting

```

function [positionReached2,Mag_Z1,Phase_Z1,RZ,XZ]=...
moveY_MeasureImp(Controller2,axis2,dist,vel,sample,EP,s,s1,s2,s3...
,Zgraph_3,Pgraph_2)

%% Move Stage
```

```
disp('**********************Stage ...
   Moving**********************');
Controller2.VEL(axis2,vel);
positionReached2 = Controller2.qPOS(axis2);

commandedPosition2=positionReached2+dist;
fs_out = 250000;
dur = 0.5;
cal_out = 1;
cal_ch1 = 1;
cal_ch2 = 3.3;
x_length = round(dur*fs_out);
t = (0:x_length-1)/fs_out;
channel_skew=4e-6;
% Multiplexing
pair1 = [0 0 0];pair2 = [0 0 1];pair3 = [0 1 0];
pair4 = [0 1 1];pair5 = [1 0 0];pair6 = [1 0 1];
pair7 = [1 1 0];pair8 = [1 1 1];

Samples1=1:1:sample;
%% Y axis Controller MOve
Controller2.MOV ( axis2, commandedPosition2 );
positionReached2 = Controller2.qPOS(axis2);
%% Impedance measurement
for samples=1:sample
for cc=1:EP
    if (Controller2.IsMoving ( axis2 )==1 )
            disp ( 'Stage-2 is moving | y-axis')
    end
    if cc==1
            outputSingleScan(s1,pair1); %6211
            outputSingleScan(s2,pair1); %6009
    elseif (cc==2)
            outputSingleScan(s1,pair2);
            outputSingleScan(s2,pair2);
    elseif (cc==3)
            outputSingleScan(s1,pair3);
            outputSingleScan(s2,pair3);
    elseif (cc==4)
            outputSingleScan(s1,pair4);
            outputSingleScan(s2,pair4);
    elseif (cc==5)
            outputSingleScan(s1,pair5);
            outputSingleScan(s2,pair5);
```

```
elseif (cc==6)
        outputSingleScan(s1,pair6);
        outputSingleScan(s2,pair6);
elseif (cc==7)
        outputSingleScan(s1,pair7);
        outputSingleScan(s2,pair7);
    else
        outputSingleScan(s1,pair8);
        outputSingleScan(s2,pair8);
end

f = 1000;
out_data = 1*sin(2*pi*f*t)'*cal_out;
queueOutputData(s,out_data);
wait(s, 1.1*dur+0.5);
[input_data,¬] = startForeground(s);
input_data = detrend(inpit_data);
input_ch2=input_data(:, 1)*cal_ch2;
input_ch1 = input_data(:,2)*cal_ch1;
[Mag_Z, Phase_Z] = ...
   frequency_response(input_ch1, input_ch2);
phi = 2*pi*f*chskew;
Phase_Z = Phase_Z + phi;
Phase_Z=unwrap(Phase_Z);
Phase_Z=rad2deg(Phase_Z);
% save the data into frequency response vectors
if (cc==1)
Mag_Zvec(cc,samples) = Mag_Z;
Phase_Zvec (cc,samples)= Phase_Z;
R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
elseif (cc==2)
Mag_Zvec(cc,samples) = Mag_Z;
Phase_Zvec (cc,samples)= Phase_Z;
R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
elseif (cc==3)
Mag_Zvec(cc,samples) = Mag_Z;
Phase_Zvec (cc,samples)= Phase_Z;
R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
elseif (cc==4)
Mag_Zvec(cc,samples) = Mag_Z;
Phase_Zvec(cc,samples)= Phase_Z;
```

```
                R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
                X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
                elseif (cc==5)
                Mag_Zvec(cc,samples) = Mag_Z;
                Phase_Zvec (cc,samples)= Phase_Z;
                R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
                X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
                elseif (cc==6)
                Mag_Zvec(cc,samples) = Mag_Z;
                Phase_Zvec (cc,samples)= Phase_Z;
                R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
                X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
                elseif (cc==7)
                Mag_Zvec(cc,samples) = Mag_Z;
                Phase_Zvec (cc,samples)= Phase_Z;
                R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
                X(cc,samples)=abs(Mag_Z*sind(Phase_Z));
                else
                Mag_Zvec(cc,samples) = Mag_Z;
                Phase_Zvec (cc,samples)= Phase_Z;
                R(cc,samples)= abs(Mag_Z*cosd(Phase_Z));
                X(cc,samples)=abs(Mag_Z*sind(Phase_Z));

            end
            % Live graphs in GUI
            if (cc==1)
                semilogy (Pgraph_2,Samples1(1:samples),...
                abs(Phase_Zvec(1,1:samples)),'-r')
                drawnow limitrate;
                semilogy (Zgraph_3,Samples1(1:samples),...
                Mag_Zvec(1,1:samples),'-r')
                drawnow limitrate;
            elseif (cc==2)
                semilogy (Pgraph_2,Samples1(1:samples),...
                abs(Phase_Zvec(1,1:samples)),'-r',...
                Samples1(1:samples),abs(Phase_Zvec(2,1:samples))...
                ,'-b');
                drawnow limitrate;
                semilogy (Zgraph_3,Samples1(1:samples),...
                Mag_Zvec(1,1:samples),'-r',Samples1(1:samples),...
                Mag_Zvec(2,1:samples),'-b');
                drawnow limitrate;
            else
                semilogy(Pgraph_2,Samples1(1:samples),...
        abs(Phase_Zvec(1,1:samples)),'-r',...
        Samples1(1:samples),abs(Phase_Zvec(2,1:samples)),...
        '-b',Samples1(1:samples),abs(Phase_Zvec(3,1:samples))...
        ,'-g');
        drawnow limitrate;
        semilogy(Zgraph_3,Samples1(1:samples),...
        Mag_Zvec(1,1:samples),'-r',Samples1(1:samples),...
        Mag_Zvec(2,1:samples),'-b',Samples1(1:samples),...
        Mag_Zvec(3,1:samples),'-g');
        drawnow limitrate;
      end
      end;
      Mag_Z1=Mag_Zvec;
      Phase_Z1=Phase_Zvec;
      RZ=R;
      XZ=X;
end
```

7. Machine learning algorithms

```
% k-nearest neighbour
knn_SC =fitcknn(x,y,'NumNeighbors',5);
save('knn_SC','knn_SC')

% Random Forest
RF_SC =TreeBagger(5,x,y);
save('RF_SC','RF_SC')

% Shallow Neural Network
trainFcn = 'trainscg';
hiddenLayerSize = [100,10];
net1 = patternnet(hiddenLayerSize, trainFcn);
% Train the net1work
[net1,tr] = train(net1,x,y);
save('net1','net1');

% Ensemble classifier Function
function [y]=ensemble_classifier(x)
load knn_SC.mat
load net1.mat
load RF_SC.mat
yknn=predict(knn_SC,x);
yqdisc=predict(RF_SC,x);
```

```
ynn = net1(x');
ynn = vec2ind(ynn);
predictions=[yqdisc yknn ynn];
[count,values]=hist(predictions,unique(predictions));
[Vmax,argmax]=max(count);
y=values(argmax);
end
```

8. Online prediction and correction - GUI code

```
app.MLPrediction_impact.Value=('');
app.MLPrediction_trajectory.Value=('');
[app.positionReached2,app.ModZ,app.PhZ,app.RZ,app.XZ]=...
MoveY_MeasureImp(app.Controller2,app.axis2,app.dist,app.vel,...
app.sample,app.EP,app.s,app.s1,app.s2,app.s3,app.Zgraph_3,...
app.Pgraph_2);
app.ModZ_PhZ=[app.ModZ(1:1,:) app.PhZ(1:1 ,:) app.RZ(1:1 ,:) ...
    app.XZ(1:1 ,:) ]
assignin('base','ModZ_PhZ1', app.ModZ_PhZ);
%Path Prediction
yfit_SC= ensemble_classifier_SC(app.ModZ_PhZ);
if (yfit_SC==1)
    app.MLPrediction_impact.Value=('before curved impact');
else
    app.MLPrediction_impact.Value=('after curved impact');
end
%direction prediction
yfit=ensemble_classifier(app.ModZ_PhZ);
if (yfit==1)
    app.MLPrediction_trajectory.Value=('Left');
elseif (yfit==3)
    app.MLPrediction_trajectory.Value=('Right');
else
    app.MLPrediction_trajectory.Value=('Middle');
end

% Actuator Correction according to prediction
if (yfit==1)
    app.positionReached3=rotation(app.Controller3,app.axis3,0.5,1)
    app.Rpos.Value=num2str(app.positionReached3);
elseif (yfit==3)
    app.positionReached3=rotation(app.Controller3,app.axis3,-0.5,1)
    app.Rpos.Value=num2str(app.positionReached3);
```

```
else
app.positionReached2 = app.Controller2.qPOS(app.axis2);
app.Ypos.Value=num2str(app.positionReached2);
end
app.positionReached1 = app.Controller1.qPOS(app.axis1);
app.positionReached2 = app.Controller2.qPOS(app.axis2);
app.positionReached3 = app.Controller3.qPOS(app.axis3);
app.Xpos.Value=num2str(app.positionReached1);
app.Ypos.Value=num2str(app.positionReached2);
app.Rpos.Value=num2str(app.positionReached3);
```

or

Python code

1. Classification

```
def Classification_(xdata,labels):
    cv = KFold(n_splits=5, random_state=1, shuffle=True)
    #Classifiers
    clf_KNN = KNeighborsTimeSeriesClassifier(n_neighbors=5)
    clf_SNN = MLPClassifier(hidden_layer_sizes=(100,10)
    ,max_iter=1000,activation = ’relu’,solver=’adam’,random_state=3)
    clf_SVM = svm.SVC(kernel="rbf")
    clf_RF = RandomForestClassifier(n_estimators=5)  #Random Forest
    #scaling
    scaler = StandardScaler()
    scaled_data = scaler.fit_transform(xdata)
    Z=scaled_data

    #model fitting
    scores_SVM= cross_val_score(clf_SVM, Z, labels, scoring=’accuracy’,
    cv=cv, n_jobs=-1)
    scores_SNN = cross_val_score(clf_SNN, Z, labels, scoring=’accuracy’,
    cv=cv, n_jobs=-1)
    scores_KNN = cross_val_score(clf_KNN, Z, labels, scoring=’accuracy’,
    cv=cv, n_jobs=-1)
    scores_RF = cross_val_score(clf_RF, Z, labels, scoring='accuracy',
    cv=cv, n_jobs=-1)

    scores= [mean(scores_SVM),std(scores_SVM),mean(scores_SNN),std(scores_SNN),
            mean(scores_KNN), std(scores_KNN),
            mean(scores_RF), std(scores_RF),
            mean(scores_QD), std(scores_QD)]
    return scores
```

2. Feature Extraction and Selection

```
#Feature extraction using TSFRESH
from tsfresh.feature_extraction import extract_features,
MinimalFCParameters, EfficientFCParameters
def feat_extract(df):
    eff_feat=extract_features(df,column_id="Insertion",
    column_sort="timeStamps", default_fc_parameters=EfficientFCParameters())
    return eff_feat


#Feature selection
def select_feat(feat_set,labels,k):
    from sklearn.feature_selection import f_classif
    min_max_scaler = preprocessing.MinMaxScaler()
    minimal_features1 = min_max_scaler.fit_transform(feat_set.values)
    Efficient_features_df=pd.DataFrame(data=minimal_features1,
    index=np.arange(1,138), columns=feat_set.columns)
    selector = SelectKBest(f_classif, k)
    selector.fit(Efficient_features_df, labels)
    cols = selector.get_support(indices=True)
    features_df_new = Efficient_features_df.iloc[:,cols]
    return features_df_new
```

3. Convolution and LSTM Hybrid Regression Model

```
    from keras.models import Sequential
```

```
from keras.layers import Dense, Conv1D, Flatten,
BatchNormalization,MaxPooling1D
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from keras.layers.recurrent import LSTM
from keras.callbacks import EarlyStopping
from keras.optimizers import SGD
from keras.optimizers import Adam

data=ds1[:,:4]
depth=ds1[:,-1]
x = data.reshape(data.shape[0], data.shape[1], 1)
y=depth
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
    test_size=0.15, shuffle = False, random_state = 8)
xtrain, xval, ytrain, yval = train_test_split(xtrain, ytrain,
    test_size=0.176, random_state= 8) # 0.176 x 0.85 = 0.15

model = Sequential()
model.add(Conv1D(12, 4, activation='relu',
padding='same', input_shape=(4,1)))
model.add(LSTM(12))
model.add(Dense(25, activation="relu"))
model.add(Dense(1))

opt = Adam(lr=0.001)
model.compile(loss="mse", optimizer=opt)
model.summary()
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=200)
history = model.fit(xtrain, ytrain, epochs=1000, batch_size=30,
                    validation_data=(xval, yval), verbose=2,
                    shuffle=False,callbacks=[es])

plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='validate')
plt.legend()
plt.show()
ypred = model.predict(xval)
print(model.evaluate(xtrain, ytrain))
print("MSE: %.2f" % mean_squared_error(yval, ypred))
x_ax = range(len(ypred))
plt.scatter(x_ax, yval, s=5, color="blue", label="original")
plt.scatter(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```