The Electrodeposition of Tin Coatings from Deep Eutectic Solvents and their Subsequent Whisker Growth

By

Christopher James Stuttle
MPhys (University of Southampton), MSc (Loughborough University)

A Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy of Loughborough University

September 2014

Supervisors Dr. Geoff D. Wilcox
Prof. Roger J. Mortimer

Department of Materials,

Loughborough University

©Christopher James Stuttle - 2015

Dedication

This thesis is dedicated to those I love: Christ my God, Lord, and saviour who has given me all I cherish; my parents, Andrew and Wendy, who have shown me living faith in times of blessing and trial; and my brothers and sisters in Christ.

This work is but the study of God's own handiwork:

"For ever since the world was created, people have seen the earth and sky. Through everything God made, they can clearly see his invisible qualities—his eternal power and divine nature."

The Holy Bible, Romans 1:20

Acknowledgments

I would like to thank: Dr. Geoff D. Wilcox, Prof. Roger J. Mortimer and Dr. Mark A. Ashworth for their supervision, encouragement, support, wisdom, and correction over the last few years; MacDermid Plc. For the provision of Tinmac electrolyte additives used throughout this project; and the Loughborough University Graduate School Centenary Studentship for funding me through this endeavour.

Abstract

Tin electrodeposits produced from aqueous electrolytes are frequently used within the electronics industry due to their high solderability and corrosion protection. One limitation to using these deposits is their spontaneous formation of long conductive filament whiskers. These whiskers grow post-electrodeposition and increase the risk of unwanted electrical shorts within electronic devices.

In this thesis, tin electrodeposits produced from a proprietary bright acid Tinmac electrolyte, currently used in industry, were studied. Electrodeposits were produced using a range of current densities with and without agitation and were characterised with respect to crystallographic orientation, topography and surface finish. Moreover, the intermetallic compound (IMC) growth produced at the copper substrate-tin coating interface was assessed over a period of time as its growth is considered to be a significant driving force behind whisker formation.

In addition, a technique for the electrochemical anodic oxidation of tin electrodeposits on copper substrates was developed. This technique was used throughout this project for the study of IMC growth from tin electrodeposits as it was able to effectively remove the tin whilst leaving the IMCs and substrate unaffected.

Ionic liquids exhibit promising electrochemical characteristics for electrodeposition but are still not widely utilised in industry. Their ability to deposit tin coatings has been studied in the present investigation. Trials concentrated on process optimisation to produce uniform electrodeposits by varying current density, $SnCl_2.2H_2O$ concentration, and electrolyte composition. These deposits were then characterised and compared to tin coatings of similar thickness produced from Tinmac with respect to topography, surface finish, crystallographic orientation, IMC growth, and whisker propensity. Electrodeposits produced from the ionic liquid electrolyte exhibited a different crystallographic texture, topography, and IMC growth compared to those produced from Tinmac. Moreover, the deposit produced from the ionic liquid featured increased whisker growth compared to those produced from Tinmac, but in a wider context, far less growth than conventional tin electrodeposits in the literature.

In addition, by exploiting other electrochemical characteristics of ionic liquids, such as their large potential window, future work may be able to produce novel tin or tin alloy electrodeposits which may further reduce the whisker propensity of deposits produced in this investigation.

Key words

Tin electrodeposition, ionic liquids, deep eutectic solvents, tin whiskers, intermetallic compound.

Frequently used abbreviations

POV	Point Of View
POI	Point Of Interest
LOS	Line Of Sight
Ref point	Reference Point
WD	Working Distance
IMC	Intermetallic Compound
DES	Deep Eutectic Solvent
SEM	Scanning Electron Microscopy
EBSD	Electron Back-Scatter Diffraction
EDX	Energy-Dispersive X-Ray Spectroscopy
XRD	X-Ray Diffraction
HBD	Hydrogen Bond Donor
RoHS	Restriction Of Hazardous Substances
WEEE	Waste Electrical And Electronic Equipment
CV	Cyclic Voltammetry
DC	Direct Current
AC	Alternating Current
SCE	Saturated Calomel Electrode
Ag-AgCl	Silver – Silver Chloride Electrode

Content

Dedication iii

Acknowledgments iv

Abstract v

Key words v

Frequently used abbreviations vi

Content vii

Chapter 1 Introduction 1

1.1 Legislative impact of lead removal 2

1.2 Non-aqueous electrolytes 3

1.3 Scope of this research 3

Chapter 2 Literature Review 5

2.1 Tin electrodeposition 5

2.1.1 Usages of tin electrodeposits 6

2.1.2 Electrodeposition equipment and terminology 8

2.1.3 Theoretical principles 9

2.1.3.1 Reduction and oxidation reactions 9

2.1.3.2 Cathodic reduction reactions 9

2.1.3.2.1 Anodic oxidation reactions 10

2.1.3.1 Electrolytic polarisation 10

2.1.3.2 Faraday's law 12

2.1.3.3 Electrodeposit nucleation and growth 13

2.1.4 Deposit manipulation by electrolyte formulation and process parameter variation 13

2.1.5 Electrolyte formulation variation 13

2.1.5.1 Aqueous electrolytes 14

2.1.5.2 Non-aqueous electrolytes 17

2.1.5.3 Aqueous– non-aqueous electrolyte comparison 19

2.1.6 Tin electrodeposition from ionic liquid electrolytes 21

2.1.7 Tin electrodeposition from DES electrolytes 22

2.2 Tin whisker growth 24

2.2.1 What are tin whiskers? 26

2.2.2 Filamental whiskers 27

2.2.3 A brief history of whiskers 28

2.2.4 Tin whisker caused failures 29

2.2.5 Whisker related industry standards 30

2.2.6 Whisker growth theory 31
2.2.6.1 Compressive stresses 31
2.2.6.1.1 Whisker growth thermodynamics 31
2.2.6.1.2 Cracked oxide theory 32
2.2.6.1.3 Intermetallic compound growth 33
2.2.7 Whisker propensity measurements 35
2.2.7.1 Distribution of whisker lengths 36
2.2.7.2 Time interval before whisker propensity measurements 36
2.2.8 Whisker measurement methods 38
2.2.9 Whisker risk management 38
2.2.10 Whisker mitigation methods 39
2.2.10.1 Substrate surface roughness increase 40
2.2.10.2 Use of an underlay coating 40
2.2.10.3 Appropriate tin electrodeposit selection 42
2.2.10.4 Post deposition annealing 47
2.2.10.5 Circuit and component design 47
2.2.10.6 Application of a conformal coating 47
2.3 Discussion of literature 48
2.4 Conclusion of literature 49
2.5 Aims and objectives 50

Chapter 3 Experimental Procedure 51
3.1 Copper substrate preparation 51
3.2 Electrodeposition from an aqueous electrolyte 53
3.2.1 Effect of deposition current density on deposit structure 53
3.3 Electrodeposition from non-aqueous deep eutectic solvents 53
3.3.1 Non-aqueous electrolyte DC voltammetry for tin electrodeposition 54
3.3.2 Optimisation of electrodeposition parameters 55
3.4 Comparison of electrodeposits from aqueous and non-aqueous electrolytes 58
3.5 Electrochemical anodic oxidation of tin 58
3.5.1 Procedure for technique development 58
3.5.2 Electrochemical anodic oxidation procedure of tin for IMC study 61
3.6 Surface and subsurface characterisation techniques 63
3.6.1 Scanning electron microscopy 63
3.6.2 X-ray photoelectron spectroscopy depth profiling 64
3.6.3 Dual beam microscopy 64
3.6.4 Whisker measurement techniques 64
3.7 Experimental summary 64

Chapter 4 Results and Discussion **66**
4.1 Electrodeposition from an aqueous electrolyte 66
4.2 Outcome of technique development for the electrochemical anodic oxidation of tin 82
4.3 Intermetallic compound growth for deposit produced from an aqueous electrolyte at 20 mA/cm^2 89
4.4 Electrodeposition from non- aqueous deep eutectic solvents 97
4.4.1 Non-aqueous electrolyte DC voltammetry and cathode current efficiency measurements for tin electrodeposition 97
4.4.1.1 Cathodic reactions 97
4.4.1.2 Anodic reactions 100
4.4.1.3 Cathode current efficiency 102
4.4.2 Non-aqueous electrolytes: optimisation of deposition parameters 107
4.4.2.1 Thin electrodeposit quality evaluation 107
4.4.2.2 Preparation of electrodeposits with increased thickness 113
4.5 Comparison of electrodeposits from aqueous and non-aqueous electrolytes 117
4.5.1 Electrodeposit topography comparison 119
4.5.2 Electrodeposit crystallographic comparison 120
4.5.3 Electrodeposit contamination comparison 122
4.5.4 Electrodeposit IMC growth comparison 123
4.5.5 Electrodeposit whisker growth comparison 125
4.6 Creation, development and testing of 3Dview for whisker length measurement 132
4.6.1 Translation of a three dimensional coordinate system 134
4.6.2 Rotation of a three dimensional coordinate system 134
4.6.3 Calculation of the coincident point for two POVs 137
4.6.3.1 Calculus minimisation of length S 137
4.6.3.2 S is defined to be perpendicular to both LOSs 138
4.6.4 Testing of 3Dview 138
4.6.4.1 Virtual agreement with 'Google SketchUp™' 138
4.6.4.2 Copper scratches 139
4.6.4.3 Reproducing lengths with real whiskers 140
4.6.4.4 Closing remarks on 3Dview testing 142

Chapter 5 Synoptic Discussion **143**

Chapter 6 Conclusions **147**
6.1 Electrodeposition from an aqueous electrolyte 147
6.2 IMC growth for electrodeposit produced from an aqueous electrolyte 147
6.3 Anodic electrochemical tin stripping 148
6.4 Non-aqueous electrolyte DC voltammetry for tin electrodeposition 148

6.5 Comparison of electrodeposits from aqueous and non-aqueous electrolytes 148
6.6 Whisker measurement programme 149

Chapter 7 Future work 150

Chapter 8 References 152

Chapter 9 Publications and presentations during the Ph.D. 171
9.1 Papers 173
9.2 Posters 174
9.3 Oral presentation slides 176
9.4 Competition winning micrographs 177

Chapter 10 Appendix 179
10.1 Non-aqueous electrolyte DC voltammetry for tin electrodeposition 179
10.2 Aqueous electrodeposit grain size distributions 180
10.3 Thermodynamics – whisker growth by stress relaxation 182
10.3.1 Whisker growth from the relaxation of its original grain 183
10.3.2 Whisker growth from deposit relaxation 183
10.4 3DView programme code 186
10.4.1 Form appearance 186
10.4.2 Programme code 188

Chapter 1 Introduction

The addition of a coating to another material optimises its surface, either for protective or decorative reasons [1]. A tin coating is one such example, by applying tin to the surface of an item, the item gains the surface characteristics of tin. Tin is non-toxic, low cost, solderable, conductive, ductile, and tarnish and corrosion resistant; making it a popular coating for many applications [2].

The most frequent use of tin as a coating material, in terms of tonnage, is found in the food industry in the form of tinplate used for tin cans; mild steel coated with a layer of tin [3]. A lesser consumer of tin coatings is found within the electronics industry. By coating solder contacts and component lead frames with tin the high solderability and protective nature of tin is exploited; increasing wettability during soldering, and reducing corrosion and tarnishing [4]. Although other materials exhibit comparable levels of solderability, it is tin's combined attributes that make it a preferred coating.

A tin coating for the electronics industry can be deposited with or without using an electrical current. Historically, tin coatings have been applied using "hot dipping", i.e. the immersion of a component leadframe into molten tin, using tin's high wetting ability to create a coating [5], [6]. Less frequently tin coatings are produced using techniques referred to as electroless deposition and immersion-tin. These coatings occur due to surface chemical reactions, and are frequently very thin. Whereas, the present-day primary means of component coating is electrolytic, by the electrochemical reduction of dissolved tin onto a surface by an applied cathodic electrical potential.

Although pure tin, as a coating, has many beneficial properties, it is not without two significant limitations; namely, *tin pest* and *tin whiskers* [7]–[9]. Tin pest is the allotropic phase transformation from β-phase to α-phase causing an atomic structure alteration. This results in the tin becoming brittle and expanding by up to 26%, leading to a loss of physical integrity, as seen in Figure 1.1. Although this transition initiates between 13-18 ^{o}C, for this phase change to be sustained temperatures of below -30°C must be experienced [10], [11]. Therefore, most circuits utilising pure tin coatings can be managed effectively with appropriate forethought, or by use of tin alloying with additions of bismuth, antimony or lead which are reported to reduce tin pest [7], [9], [12].

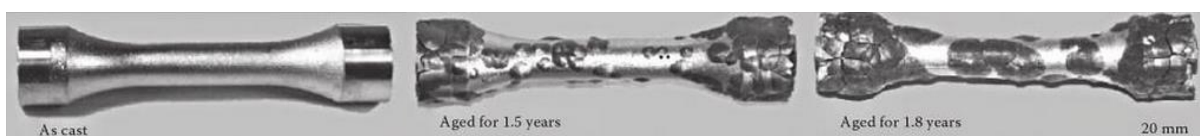


Figure 1.1: Photograph of as cast tin component displaying tin pest after prolonged low temperatures [12]

Tin whiskers, a significant feature of this thesis, are protruding growths formed from tin coatings post-deposition, with an example seen in Figure 1.2. Tin whiskers, most commonly reported as growing from electrodeposited tin, have also been seen to be formed from hot-dipped tin coatings [13]. Although some geometries of whiskers are harmless due to their short lengths, they have been reported to grow into long filaments with lengths exceeding 3 mm [14]. It is when whiskers grow beyond the pitch of electrical contacts that, by nature of being a conductor, circuit failures are caused by the formation of unwanted electrical connections. It is for this reason that the tin electrodeposition and electrical industries, who create and utilise tin coatings, are striving to lessen whisker growth propensities, and reduce the risk of electrical shorts caused by their growth, for the sake of increased product reliability.

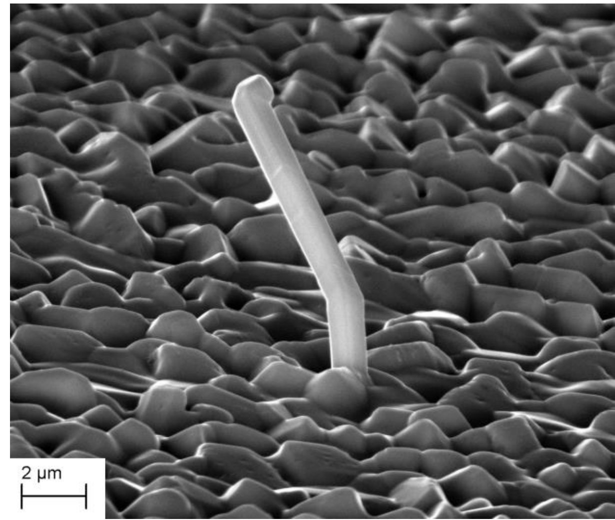


Figure 1.2: SEM micrograph of whisker growing from a tin electrodeposit

Historically, these industries utilised electrodeposited tin coatings and tin solders containing lead to mitigate against whiskers [8]. This means of mitigation was considered successful and was applied extensively throughout the electronics industry until recent legislation restricted the use of lead.

1.1 Legislative impact of lead removal

The use of lead within consumer products is increasingly difficult due to its detrimental health and environmental effects. Historically, legislation concerning lead content has applied to drinking water contamination, but over more recent years legislation has extended its reach towards electrical products, in the form of two directives: Waste from Electrical and Electronic Equipment (WEEE) [15]; and Restriction of Use of certain Hazardous Substances in Electrical Equipment (RoHS) [16].

As a result, electrical equipment which utilised lead containing solders or coatings were forced to find alternatives to comply with these regulations. Subsequently, the mitigation of whiskers offered by lead for tin solders or component coatings has been removed, causing an increase in whisker growth and their related failures [17].

1.2 Non-aqueous electrolytes

The number of electrodeposited tin alloys is limited by the electrochemical potential window of commonly utilised water based electrolytes. This potential window is larger for some non-aqueous electrolytes, allowing the possibility of the electrodeposition of novel pure tin and novel tin alloyed deposits. It is these electrodeposits of pure tin and novel tin alloys, enabled by recently discovered ionic liquids, which may form deposits which possess significantly lower whisker propensities compared to current alternatives produced from aqueous electrolytes.

1.3 Scope of this research

The research for this thesis has focused upon the electrodeposition of tin from three ionic liquid electrolytes from within a subgroup referred to as Deep Eutectic Solvents (DES): Reline, Ethaline and Propeline. By optimising deposition parameters and electrolyte tin concentration for deposit uniformity, pure tin coatings on copper substrates were created from the DESs and compared to tin deposits produced from a proprietary aqueous electrolyte. These deposits were characterised with respect to: uniformity, topography, crystallographic structure, deposit thickness, intermetallic compound (IMC) growth, and whisker propensity.

To facilitate this research a review of the current literature for the electrodeposition of tin, and the growth of tin whiskers was undertaken. Literature pertaining to the process of electrodeposition and specifically the electrodeposition of tin is extensive. Therefore, from this a summary has been compiled presenting historical background, theoretical principles, and examples of electrodeposit manipulation by process parameter and electrolyte formulation variation. Within these areas attention has been drawn to novel non-aqueous ionic liquid electrolytes. Moreover, within this specific niche, emphasis has been placed upon the deposits formed from these electrolytes rather than their electrochemistry.

Additionally, a review upon the current literature associated with tin whisker growth has been undertaken. This review comments upon historical background, thermodynamic theory, whisker growth measurements, and means by which risk due to whisker growth may be reduced. Within this area there has been recent emphasis upon theoretical understanding to mitigate their growth, whereas in this review emphasis is placed upon applicable means by which whisker risk can be managed.

The experimental research focuses upon the optimisation and characterisation of electrodeposits formed from non-aqueous electrolytes. Due to the novelty of these electrolytes both process parameter and electrolyte formulation were varied, attempting to create an electrodeposit

similar in uniformity, homogeneity, and thickness to a deposit produced from a proprietary electrolyte. It was soon discovered that deposits produced from the non-aqueous electrolytes experienced significant reductions in uniformity with respect to thickness.

The results and discussion in this thesis shows that the electrolyte formulations utilised within this study, with appropriate tin additions, are able to present similar deposition rates and cathode current efficiencies compared to an aqueous alternative but it was soon seen that the uniformity of the majority of deposits was of far lower quality. Nevertheless, one deposit was produced uniformly and was characterised comparatively to deposits of equivalent thickness from the aqueous electrolyte. The topography, crystallographic orientation, IMC distribution for both deposits were shown to be different. Moreover, the whisker propensity for the deposits produced from the DES were seen to be larger than those from the aqueous electrolyte.

Furthermore, due to an atypical IMC growth distribution and whisker propensity compared to alternative studies expressed during the earlier comparison of electrodeposits, deposits at increased thicknesses were produced from the aqueous control electrolyte and examined in greater detail. They showed that, as increased thicknesses deposits developed, a distinct preferred crystallographic orientation and an atypical IMC distribution were evident.

Chapter 2 Literature Review

2.1 Tin electrodeposition

The theoretical process of electrodeposition, i.e. the reduction of metallic cations from aqueous solution onto an electrode, was first proposed by the alchemist Zozimus around 201 to 400 AD [18]. Although passive electrodeposition, requiring no electrical driving potential, was developed in 1801 by utilising two electrode metals with different electronegativities, the process of driven electrodeposition required the discovery of the voltaic pile [18]. Although the voltaic pile was able to provide some electrical power, significant impetus was provided later throughout the 1800s by the creation of the dynamo by Michael Faraday; providing sustained, controlled and sizeable electrical power [19]. This power source, allowed for the electrodeposition of more metals such as nickel, copper, brass and tin.

It was in the 1950s that the benefits of alloying tin with lead for hot-dipped coatings to mitigate whisker growth were highlighted [20]. Then in 1959, lead was seen to be advantageous for whisker mitigation in electroplated tin [8]. The addition of lead, at 3-10% by weight, to tin electroplate was a simple and effective method of preventing whiskers and as such remained the primary tin coating used in electronics for over 50 years.

Lead poisoning has been known about for over 2500 years, causing symptoms from mild fatigue and exhaustion to peripheral neuropathy and encephalopathy [21], yet modern lead containing materials are not always disposed of appropriately. As a result lead reaches landfills and over many years seeps into the local environment, and eventually into the local water-table [22], [23].

Two directives have been signed (27th January 2003) and enforced (1st July 2006) by the European Union to combat lead pollution; referred to as: WEEE; and RoHS. These directives attempt to minimise lead containing products from reaching the environment by managing production, within the RoHS directive, and disposal, within the WEEE directive.

Lead has been eliminated from many items, such as paint and fuel additives as their chemicals detrimentally affect health. The RoHS and WEE directives reach further, banning materials which affect the environment and may eventually indirectly affect health. Although the risk of lead poisoning from electrical devices by direct contact must be thought negligible, the environmental impact caused by lead entering landfills, and subsequently the water table, has caused concern.

The resulting legislation for the European Union member states, consequently followed by other nations, ban lead from being used in most materials, with minor exceptions to military and aerospace applications. Although these industries are allowed to utilise lead within their products, its removal from the majority of industries forces manufacturers to consider the production of components with surface finishes containing or void of lead. As a result of the reduced number of components which will be used in lead containing products, there will most likely be an increased cost of lead containing items as their production methods become increasingly bespoke. Moreover, companies permitted to use tin-lead coatings, at expense, are re-finishing pure tin coated components to ensure a tin-lead finish [24].

The effect that lead had in electronics was unparalleled by any other addition. It was able to mitigate tin pest and reduce whisker growth for tin rich materials to such an extent that little concern was given to the risk that tin whiskers could create. As a result, when the RoHS directive was enforced there was no immediate equivalent replacement.

2.1.1 *Usages of tin electrodeposits*

Industry standards concerning the electrodeposition of tin by the American Society for Testing and Materials (ASTM) introduces many requirements of tin coatings for various industrial applications [25]. Within these standards minimum thickness, dependent upon application, are given shown in Table 2.1. Moreover, tin whiskers are mentioned within this document as a warning to specific tin electrodeposit usage.

Requirements concerning tin electrodeposit quality from ASTM state that:

- The substrate must be cleaned before application.
- A coating will only be applied post-substrate heating or bending.
- A coating’s appearance shall be uniform throughout, adherent to the substrate and free from blisters, pits, peeled areas, cracks, nodules, discoloured areas, unplated areas, and free from grease and oil.
- Coatings will be absent from defects or damage visible at 4x magnification
- Tin coatings used for soldering below 2.5 µm in thickness shall utilise a 1.3 µm thick nickel under-plate.

- Brass substrates, or other zinc containing alloys above 5%, shall use a 2.5 µm copper or 1.3 µm nickel underlay, with thicker underlays used for longer-term applications to limit zinc diffusion.

Table 2.1: Common tin electrodeposited coating thickness requirements and applications [25]

Minimum thickness	Typical application
1.5 µm	Similar to 2.5 µm thick coatings, but for products with shorter shelf-life applications.
2.5 µm	Coatings which are shielded from atmospheric effect such as electrical connector housing. Used to provide corrosion and tarnish resistance where thicker coatings are undesired. Often not to be used for soldering but must offer low resistance electrical contact surfaces.
5 µm	Coatings to be used in harsher conditions than 2.5 µm thick coatings. Used for solder contacts, surface preparation for painting, antigalling purposes and stopoff in nitriding.
8 µm	Coatings to be used in harsher conditions than 5 µm thick coatings, and primarily for in-door use. Used in solder contacts in electrical devices even after long storage times
15 µm	Coatings to be used in severe service conditions, including exposure to damp and mild corrosion environments. Used for gas meters, and automotive accessories and in some electronic devices where wanted.
30-125 µm	Coatings used in very severe service conditions, including elevated temperatures. Coatings are subject to mechanical abrasion or corrosive environments. Coatings used for water containers, oil drilling and seacoast applications

Tin electrodeposits are used in many applications, some of which are given within an industry review undertaken in 1970 for tin and tin-alloy electrodeposits, shown in Table 2.2. Of interest within this table is the large number of pure tin applications within the electronics industry prior to the implementation of the RoHS directive. Were these parts plagued by tin whiskers and the electrical shorts they caused?

Table 2.2: Tin and tin-alloy applications [26]

Tin	Tin-lead	Tin-copper	Tin nickel
Copper wire	Overlay on bearings	Nitriding stopoff	Decorative coating
Tags	Radio-TV chassis	Undercoat for tin	Surgical instruments
Terminal pins	Radio-TV fittings	Undercoat for Ni-Cr plating trophies	Drawing instruments
Eyelets	Capacitor cans		Watch pars
Fasteners	Small electronic parts		Musical instruments
Washers	Copper wire		Balance weights
Screws	Printed circuit boards		Optical apparatus
Tubing			Printed circuit boards
Water coolers			Electrical switches
Aluminium bus bars			Electrical connectors
Switch parts			Plumbing hardware
Circuit breaker parts			
Chassis and parts			
Overlay on bearings			
Copper and brass sheets			
Aid in drawing steel			

2.1.2 *Electrodeposition equipment and terminology*

Electrodeposition is described by many interrelating principles, terms and processes. As such, there are many different experimental arrangements and procedures that can be used to undertake and analyse this process. This section will describe the equipment, and some of the fundamental terminology utilised when describing electrodeposition. Firstly, the simplest experimental arrangement for electrodeposition is achieved by passing an electrical current, produced by a power supply or potentiostat, between two electrodes submersed in conductive electrolyte, leading to electrochemical reduction at the cathode, forming the desired deposit. A simple development of this arrangement is the addition of a reference electrode, as shown in Figure 2.1.

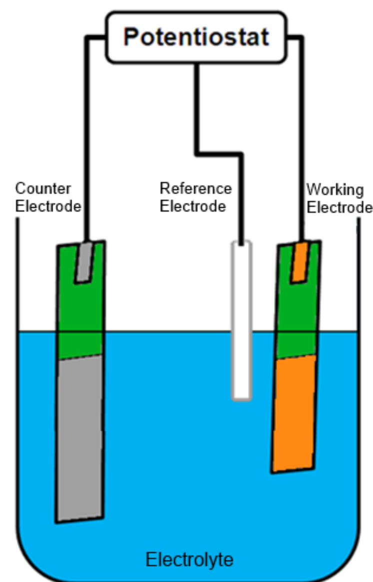


Figure 2.1: Schematic of a three electrode cell commonly used for electrodeposition and electrochemical analysis

By the addition of a reference electrode, it is possible to record the potential difference for the working and counter electrodes. This is then able to discriminate the cell voltage, the potential drop between the working and counter electrodes, into two separate potentials; the working electrode potential and the counter electrode potential.

2.1.3 *Theoretical principles*

There are many theoretical principles that attempt to describe the complex field of electrodeposition. Therefore, for the sake of brevity, only the specific, significant and fundamental theoretical principles have been highlighted.

2.1.3.1 *Reduction and oxidation reactions*

The process of electrodeposition is fundamentally based upon a reduction and oxidation reaction at cathodic and anodic electrodes respectively. The generalised redox reaction, coupled reduction and oxidation, are represented in Equations 2.1 and 2.2.

Reduction: $$\mathrm{M^{y+z}_{(aq)} + ze^- \rightarrow M^{y}_{(s)}}$$ Equation 2.1

Oxidation: $$\mathrm{M^{y}_{(s)} - ze^- \rightarrow M^{y+z}_{(aq)}}$$ Equation 2.2

Where: z is an integer scalar denoting the number of electrons and valence state, M denotes an element in valence state y or y+z, and subscript 's' and 'aq' defines the element in solid or aqueous form respectively.

As seen in Equations 2.1 and 2.2, the species of the material being reduced or oxidised has been generalised, whereas the only predefined term within these reactions is the electron. Therefore, the essential distinction between these reactions is based upon their electric polarisation; cathodic or anodic. For this reason the reactions have been separated into the following cathodic and anodic subsections.

2.1.3.2 *Cathodic reduction reactions*

Any cathodic electrode potential which is utilised may itself be an over-potential for several specific reduction reactions, resulting in the possibility of multiple species being reduced at the cathode. The cathodic reduction of tin taking place on the working electrode during electrodeposition can be described by Equation 2.3.

$$\mathrm{Sn^{2+}_{(aq)} + 2e^- \rightarrow Sn^{0}_{(s)}}$$ Equation 2.3

The potential required to initiate Equation 2.3 is primarily dependent upon the valence change of tin requiring potentials below -0.136 V [27], but is also dependent upon other minor factors which

further decrease the deposition over potential. Therefore, this initiation potential is best defined empirically. Moreover, by utilising increasing over-potentials the rate of deposition is seen to increase and the character of the deposit is often seen to vary [28]–[30]. Additionally, at increasing tin reduction over-potentials it is possible to encroach upon other reduction reactions in the search for increased deposition rate or various deposit characteristics.

Given sufficient tin reduction over-potentials electrolyte reduction is achieved, leading to multiple reduction reactions occurring simultaneously. In the case of aqueous electrolytes, electrolyte reduction is described in Equation 2.4.

$$2H_2O + 2e^- \rightarrow H_2 + 2OH^- \quad \text{Equation 2.4}$$

The specific reduction of water highlights important features pertaining to the general reduction of electrolytes. Firstly, the reduction creates product concentration increases in close proximity to the forming electrodeposit which can result in deposit characteristic manipulation [31], [32]. Secondly, the consumption of current by the reduction of the electrolyte causes the total current passing through the cell to be distributed between electrolyte reduction, and tin reduction. This then leads to a reduction in cathode current efficiency in connection with the desired metal reduction. Although reduced cathode current efficiency is undesirable, it may be considered of lesser importance than other features, and be sacrificed to achieve a desired result, e.g. for the sake of deposit quality, deposition rate, or whisker propensity, cathode current efficiency could be compromised.

2.1.3.2.1 *Anodic oxidation reactions*

Similarly to cathodic reactions, it is possible to create additional species by electrolyte oxidation at the anode. These products are free to pass through the electrolyte and may result in affecting the electrodeposit much in the same way as reduction pollutants may. Additionally, during electrodeposition dissolved tin is consumed from the electrolyte resulting in a reduction in concentration. One simple resolution to these issues simultaneously, is to utilise an anode made of tin; reducing electrolyte oxidation and ensuring tin replenishment as long as passivation does not occur.

2.1.3.1 *Electrolytic polarisation*

The driving force for reactions taking place at the electrode surface is caused by a gradient in potential; commonly referred to as an electric field. The potential gradients experienced from the surface of an electrode are highly localised, propagating around 1 nm. As a result of this short

propagation, providing a potential change of 1 V over this distance, electric fields within this region can exceed 10^7 V/cm [33]!

The application of a potential to an electrode causes its surface to take up characteristic charges within the electrolyte, affecting localised electrolyte composition, and creating a double layer [34]. The double layer increases the distance over which electrode potential is experienced within the electrolyte, thus reducing the electric field. It is this reduction in electric field which is commonly referred to as polarisation. Furthermore, as work is done and charges undergo reduction or oxidation reactions the double layer becomes dynamic in nature as ions are consumed from the electrolyte and additional ions diffuse from its bulk, sustaining the double layer.

Electrolyte polarisation is non-uniform across an electrode with its edge regions experiencing greater rates of diffusion than central ones. As a result, during galvanostatic deposition, although there is a constant average current density across the whole of the electrode, central regions experience decreased localised current densities by increased polarisation. Therefore, this causes an increase in deposition rate at edge regions to sustain a constant average current density [33].

One resolution to the variation of current density across a working electrode is to utilise a small area electrode (commonly 7 mm^2), reducing the area of the purely central regions. Although this results in a more constant current density across the surface of an electrode, this situation becomes less realistic compared to industrial practices. Therefore, laboratory electrodeposition should be assessed, not only with small electrode surface areas, but also with larger areas which contain edge, corner and central regions akin to industrial examples.

By reducing polarisation the electric fields experienced at the surface of the electrode increase. Reducing polarisation can be achieved by increasing diffusion causing the size of the double layer at equilibrium to become smaller. Therefore, diffusion of a reactive species within an electrolyte can be effected by its diffusion rate related to viscosity, its concentrations, and extent of electrolyte agitation. By agitating the solution it is possible to increase the levels of ionic diffusion and lessen the levels of polarisation. Although this method may appear uncomplicated, achieving uniform agitation across the surface of an electrode can be difficult in practice.

Additionally, if it is possible for a protrusion to form that pierces the diffusion layer, then due to virgin electrolyte and fully radial concentration gradients, localised current densities at this point increase significantly [35]. This type of growth is commonly referred to as dendritic, and as a result the uniformity of the electrodeposit is significantly decreased, as seen in Figure 2.2.

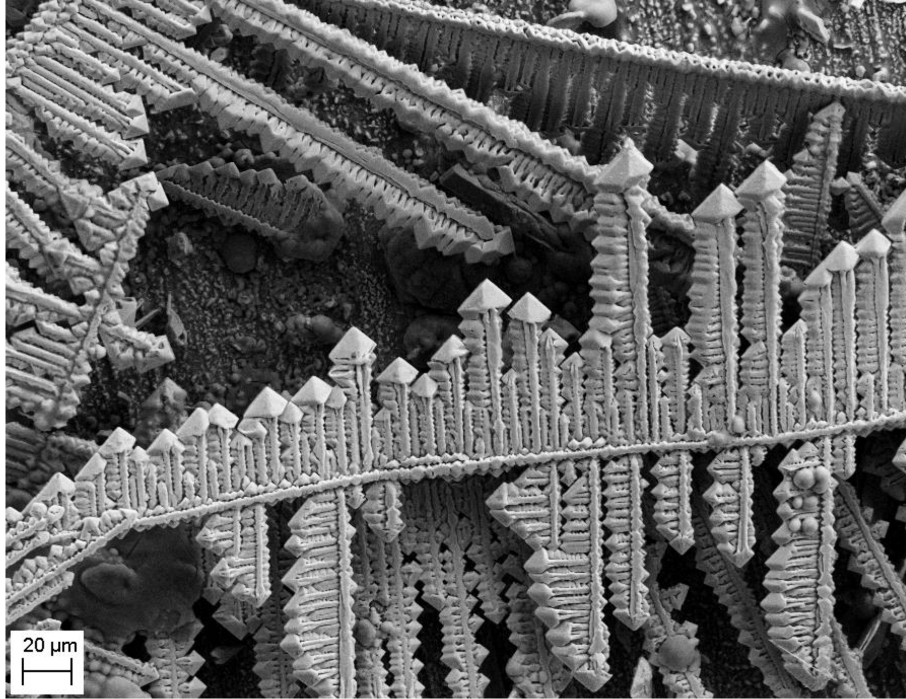


Figure 2.2: SEM micrograph of tin dendrites formed by the electrodeposition of tin from an ionic liquid electrolyte.

2.1.3.2 *Faraday's law*

Faraday's law of electrolysis is able to calculate the mass gained, 'm', during electrodeposition, as shown in Equation 2.5.

$$m = \frac{QM}{Fz} \qquad \text{Equation 2.5}$$

Where: 'Q' is charge passed; 'M' is molar mass; 'z' is the valence change; and 'F' is Faraday's constant.

Due to simultaneous reduction reactions occurring at the cathode, cathode current efficiency regarding tin reduction will often be less than 100%. For the electrodeposition of tin from a +2 valence state to its metallic form, and measuring its mass change during deposition, it is possible to utilise Equation 2.6 to calculate cathode current efficiency.

$$\varphi = \frac{2(m_2 - m_1)}{ItM_{Sn}} \qquad \text{Equation 2.6}$$

Where: 'φ' is the cathode current efficiency for tin electrodeposition; I is the current during galvanostatic deposition; and m_1 and m_2 are electrode masses prior and post deposition respectively.

Moreover, Equation 2.6 only calculates average efficiency over the whole period of electrodeposition for the net surface area of the cathode, i.e. it does not take into account time or location specific variations in efficiency.

2.1.3.3 *Electrodeposit nucleation and growth*

The study of how an electrodeposit forms on a substrate is, again, a complex system and often is empirically studied by investigating various electrolytes and process parameters [36]–[39]. Nevertheless, the growth of a deposit can often be separated into two main sections which are significantly influenced by the substrate, epitaxial growth; and those which are not, bulk growth.

As a deposit thickens each deposition layer is affected by the previous, or, in the case of the initial layer, affected by the substrate. Epitaxial regions of an electrodeposit are those which are crystallographically affected by the underlying substrate. The appearance of an epitaxial layer may be due to two reasons: the variation in deposition over-potential dependent upon substrate crystallographic orientation between grains [40]; or the crystallographic mismatch between substrate and deposit [41]. The initiation of such regions occurs within close proximity to the substrate, shown in other materials deposited by other means to be between 2 and 20 µm, but as each layer of tin is added these effects can be experienced further within the deposit. In addition, as the deposit thickens the electrodeposit is less constrained by substrate affects and experiences bulk growth [42].

2.1.4 *Deposit manipulation by electrolyte formulation and process parameter variation*

The variation of agitation method [43] and degree [44]; current density [8], [45], [46]; and operating temperature [47], [48] manipulate many electrodeposit characteristics for specific electrolyte formulations. Additionally, electrolyte formulation variations alter the affect which other variables have upon deposit characteristics, e.g. post-deposition whisker growth due to current density effects: some electrolyte formulations suggest decreased whisker growth due to increased deposition current density [45], [46], whereas others suggest different relationships [49], [50]. Therefore, deposit variations due to deposition process manipulation should include the caveat; the documented connections between process parameters and deposit characteristics have been seen for specific electrolyte formulations, which, may or may not be extrapolated to others.

2.1.5 *Electrolyte formulation variation*

The dominating process variable upon a deposit's characteristics is the electrolyte formulation. Although chemical concentration variations may result in minor electrodeposit characteristic changes, the addition of a chemical to the electrolyte may alter the deposit considerably [45], [48], [51]–[55]. A clear example of this is shown in the electrodeposition of tin from an acid sulphate electrolyte in Figure 2.3, with each chemical addition the deposit clearly improves until finally

reaching the formulations of a proprietary electrolyte with a bright deposit (Additions A – E cannot be specified due to intellectual property reasons).

Conversely, the concentration increase or supplement of an additional chemical to an already existing bath may result in no deposit manipulation [56]. This highlights the importance and complexity of additive formulation, although the resulting deposit characteristics may be desired, discovering the specific chemical to cause them may be time consuming. Nevertheless, by trial and error this development has occurred resulting in many electrolytes and additive formulations being available to the market providing many different deposits. As a result the specific additive formulations are often closely guarded trade secrets, resulting in reports generally referring to their brand-name, or generic titles, rather than specifically mentioning chemical formulae [45], [57], [58].

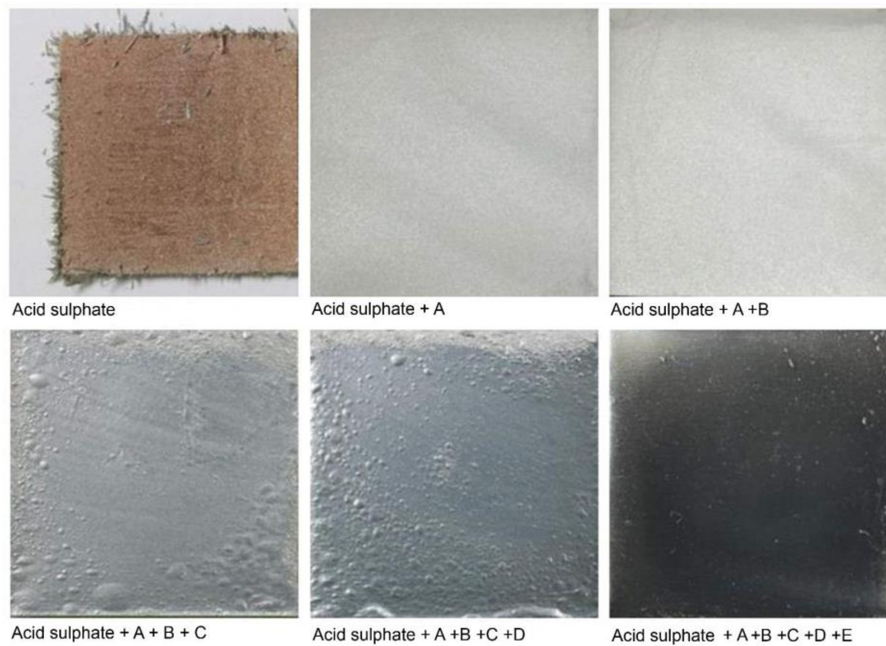


Figure 2.3: Photographs of tin electrodeposits created from an acid sulphate bath followed by various combined additive; A, B, C, D and E [59]

2.1.5.1 *Aqueous electrolytes*

Aqueous electrolytes, those utilising water as their solvent, are by far the most documented and developed, and as such their study is too large a subject to be presented in full. Therefore only a few electrolytes of interest have been selected to represent tin electrodeposition.

Water, has a low viscosity, high rates of mass transfer, high solubility for metal salts, is non-flammable, chemically and atmospherically stable, low in cost, and readily available; making it a highly attractive electrolytic medium. This solvent is then mixed with various solutes to create the electrolytes; some of the most frequently used in industry for tin deposition are shown in Table 2.3.

Table 2.3: Commonplace industrial aqueous electrolytes for tin electrodeposition. [60], [61]

Acidic	Alkaline
Fluoroboric	Stannate
Chloride	
Sulphate	
Methanesulphonate	
Citric	

Of the electrolytes shown in Table 2.3 methanesulphonic acid is by far the most frequently utilised due to its low corrosivity and high solubility to metal salts and other additives [62], [63]. Although over recent years, there has been an increasing trend toward the usage of the sulphate bath due to its reduced environmental and financial impact [64]–[66].

As aqueous electrolytes utilise water as the electrolyte solvent, deposition potentials without electrolyte degradation are limited to below 1.23 V [67]. Therefore, due to this limiting potential window, over-potentials which do not cause electrolyte degradation are confined, resulting in limited maximum current densities and, more importantly, a limited number of metals which can undergo reduction. Examples of some metals which can be reduced within the potential window of water, based upon standard reduction potentials, is shown in Figure 2.4.

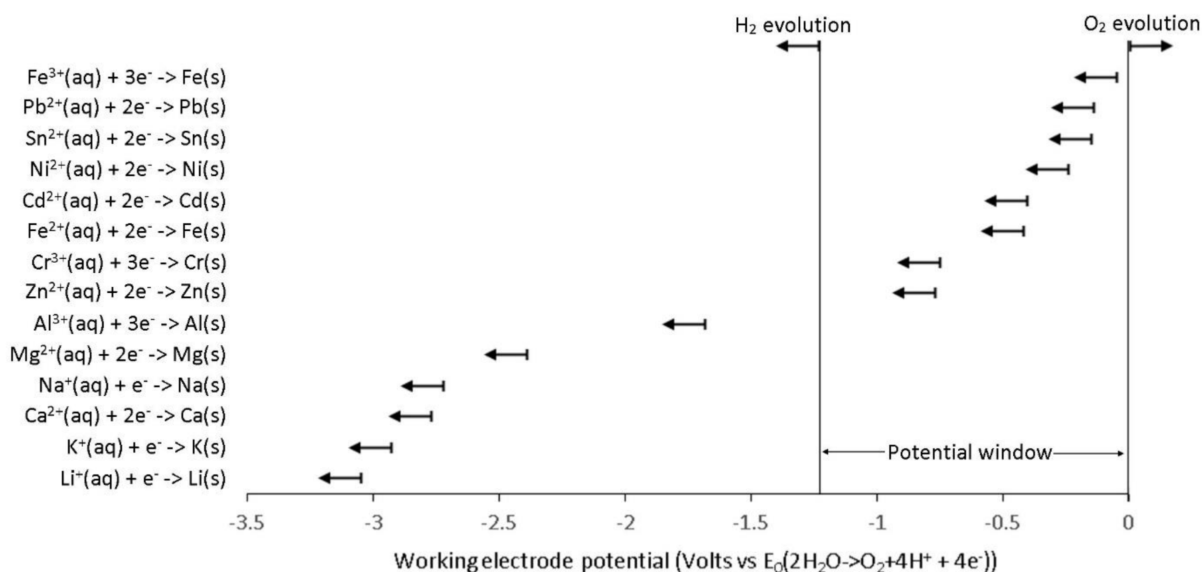


Figure 2.4: Standard reduction potentials relative to oxygen evolution for some metals within an aqueous solution to their metallic form [68]

In addition, although the electrolytes in Table 2.3 are able to deposit tin, the majority of them require additives to operate effectively, otherwise the resulting electrodeposits are of far lower quality; exhibiting low coherence, dendritic, or spongy forms [69]–[71]. Nevertheless, the proprietary electrolytes, including additives, deposit tin and its alloys well. One public domain

proprietary formulation is a sulphate solution containing tin and additions of sulphamic acid (NH_2SO_3H) [72]. This bath is of specific interest as it is able to produce a bright finish deposit which requires no additives. Although the lack of additives and public domain intellectual rights is attractive, the high whisker propensity of the deposit is far less so.

In addition to deposit uniformity varying with electrolyte formulation, preferred crystallographic orientation is also manipulated [45], [73]–[76]. One such example of this is shown in Figure 2.5.

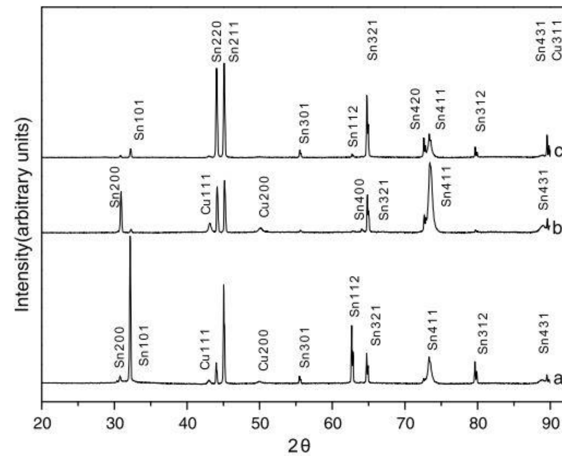


Figure 2.5: XRD profile of tin electrodeposit: (a) sulphate-based bath, (b) alkaline stannate bath, and (c) stannous chloride-based bath. [76].

Although surface grain size is seen to vary for electrodeposited tin from various aqueous electrolytes [57], [77]–[80] its substructure is commonly columnar unless a specific additive formulation has been used [45], [81]–[83]. The most well-known additive to encourage a transition from a traditional columnar tin electrodeposit structure to equiaxed is in the form of lead alloying [83]–[86]. An example of this transition is shown in Figure 2.6.

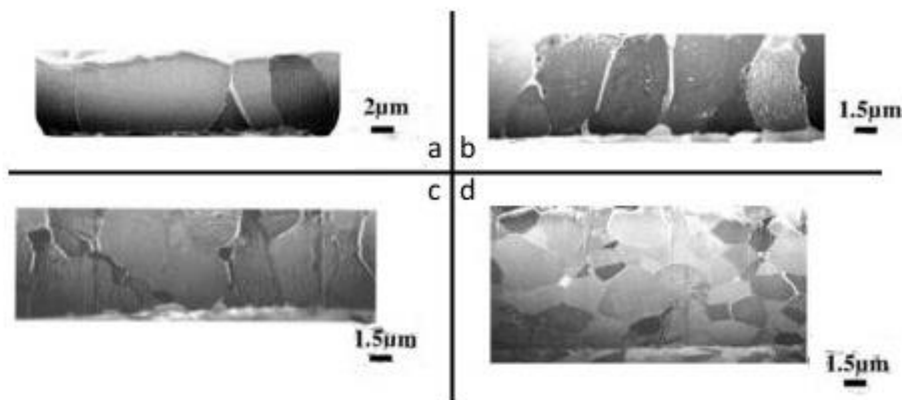


Figure 2.6: SEM micrograph of tin electrodeposit showing a transition in subsurface grain structure with increasing lead concentration; (a) Sn 100%, (b) 95%Sn–5%Pb, (c) 90%Sn–10%Pb, and (d) 60%Sn–40%Pb [83].

2.1.5.2 *Non-aqueous electrolytes*

Non-aqueous electrolytes are those which contain no water, and ideally consist of small molecules with dipolar charges resulting in a highly fluid and conductive liquid, but in reality molecules which exhibit these features also express strong metal ion bonds making reduction difficult and of little gain compared to current processes. Deviation from these optimal molecular characteristics often result in liquids which are either less conductive or feature larger viscosities, making them less attractive for electrodeposition. Nevertheless, over recent years a variety of liquids have been produced which are attractive to the electrodeposition industry largely due to their increased electrochemical potential windows compared to that of water.

Non-polar organic solvents are one such early non-aqueous electrolyte, but under examination were limited in their practicality due to their low conductivity [87], [88]. Nevertheless, these solvents expressed large potential windows and thus displayed promising possibilities for the electrodeposition of more 'exotic metals' than those deposited by aqueous electrolytes.

Other than molecular solvents, it is possible to use ionic fluids, also known as molten salts, as electrolytes. These electrolytes, due to their strong ionic bonds, require high temperatures of up to 1000 ^{o}C to produce a molten electrolyte and are said to have the majority of the benefits of the aqueous electrolytes and very few of the limitations [89], [90]. Although these electrolytes are limited in usage by their operation temperatures, they are used in the electrowinning of lithium, sodium, titanium and aluminium from their native ores [91].

A significant improvement of the high temperature ionic fluids was their evolution to room temperature molten salts. Due to large anions, such as chloroaluminate, it was possible to reduce the lattice binding energy and therefore their operation temperature [92]. By reducing the operation temperature to below 100°C it was possible to use these ionic fluids for electrodeposition, and are commonly referred to as room temperature molten salts or ionic liquids.

Chronologically, development of Ionic liquids can be classified into three generations: first, second and third [93], [94]. The first generation commenced with the discovery of ethylammonium nitrate in 1914 followed by the creation of electrolytes commonly containing anions such as chloroaluminate and cations such as alkylpyridinium and dialkylimidazolium derivatives [95]. Subsequently these liquids were soon classed as non-biodegradable and highly toxic, thus limiting their usage [96]. Moreover, they were seen to be sensitive to water and oxygen contamination and were limited in their application due to demanding working conditions [97].

The second generation of ionic liquids has been described as possessing discrete anions. This generation commenced with the development of the reactive anions, replacing the water and

oxygen sensitive anions of the first generation with halides or anions such as PF_6^-, BF_4^- $Me_2CO_2^-$, $N(SO_2CF_3)_2^-$ [98]. Due to removing the significant limitation of the first generation ionic liquids, the second generation were approached with more enthusiasm and were applied to many fields of study. Although this was a significant step in the right direction, the liquids were still non-biodegradable and expensive due to high production and disposal cost.

The third generation of ionic liquids utilised anions such as sugars, $[(CF_3SO_2)_2N^-]$, organic acids, alkylphosphates, alkylsulphates, and amino containing cations such as choline. This resulted in the creation of a biodegradable, lower cost, readily available, low toxicity, room temperature ionic liquids which were air and water stable. It is this third generation of ionic liquids which contains the sub group referred to as DESs.

Additionally, it is possible to classify ionic liquids into two subgroups, confusingly also referred to as *generations*, those utilising eutectics and those utilising discrete anions. Eutectic ionic liquids consist of two components that when mixed together at a specific ratio experience a depression of freezing point known as the eutectic point. These eutectic solvents can be additionally categorised into type I, II, III and IV. Each type of eutectic solvent has been defined by its constituents as shown in Table 2.4.

Table 2.4; Eutectic solvent classification [91], [99], [100]

Generations	General formula	Description
Type I	MCl_x, M = Zn, Sn, Fe, Al, Ga	Metal salt + organic salt
Type II	$MCl_{x}.{}_yH_2O$, M = Cr, Co, Cu, Ni, Fe	Metal salt hydrate + organic salt
Type III	R_5Z with Z = –$CONH_2$, –COOH, –OH	Organic salt + hydrogen bond donor
Type IV	MClx, Z = –$CONH_2$, –COOH, –OH	Metal salt (hydrate) + hydrogen bond donor

DESs, are found in type III eutectics and are generally characterised by being low cost, renewable, biodegradable, and of low toxicity [100], [101]. These DESs achieve charge delocalisation by the formation of a hydrogen bond between a halide anion and an amide, carboxylic acid or alcohol moiety; allowing for a liquid at room temperature [91], [100]. Furthermore, there are many possible combinations of two, or three, molecules which are able to create a DES, thus leading to a large range of possible electrolytes for examination [101].

A frequently used organic salt halide anion for DESs is choline chloride. Choline chloride is mixed with a hydrogen bond donor (HBD) to create a cation anion pair of, Choline + Cl-HBD. Although 2-ethylene glycol is already liquid at room temperature, utilising 2-ethylene glycol as the HBD creates a liquid which is still classed as a DES even though the mixture is liquid at a large range of molecular ratios. Therefore, it is not technically a eutectic solvent as it does not require this eutectic

point to function. This subjectivity is used to draw focus to the applicability of DESs rather than their technical composition. Nevertheless, this focus emphasises that these DESs are highly promising as ionic liquids to be used in industry rather than solely within a laboratory [100]–[103].

Ionic liquid electrolytes, due to increased electrochemical potential windows, as seen in Table 2.5, are able to electrodeposit additional transition and main group metals. In addition, the electrodeposition of water sensitive metals, such as magnesium [104] and aluminium [105], [106], are possible due to their low water content. Moreover, if, by utilising aqueous electrolytes, electrodeposition suffers from hydrogen embrittlement or low cathode current efficiency, the use of ionic liquids may reduce these effects. Furthermore, ionic liquids may also offer a low toxicity replacement for some of the toxic aqueous electrolytes and additives.

Table 2.5: Deep eutectic solvents and their corresponding electrochemical potential windows compared to water [107]

Deep eutectic solvent	Potential windows (V)
Choline chloride + oxalic acid	1.43
N,N-diethylethanol ammonium chloride + malonic acid	1.6
Choline chloride + malonic acid	1.62
Choline chloride + triethanolamine	1.76
Choline chloride + zinc nitrate hexahydrate	1.86
Choline chloride + 2,2,2-trifluoroacetamide	1.94
N,N-diethylethanol ammonium chloride + $Zn(NO_3).6H_2O$	2.44
Water	1.23

Although ionic liquids may offer many benefits, due to low mass transport properties attributed to high viscosities (see Table 2.7), the electrodeposition rate is often limited by diffusion [108]–[110]. Additionally, although the cost of ionic liquids is reducing, they are still relatively expensive compared to water (e.g. Glyceline, Ethaline, or Oxaline at 200 £/Kg [111] vs. deionised water 2.90 £/Kg [112]) [103]. Furthermore, at the present time, electrodeposition from ionic liquids is still in its infancy and although it is possible to electrodeposit many of the novel and traditional metals from an ionic electrolyte, it is far less commonplace to do so in a bright, or uniform manner. This is why recent work has been undertaken to develop ionic liquids and ionic liquid additives [54], [99], [113].

2.1.5.3 *Aqueous– non-aqueous electrolyte comparison*

Due to varying potential windows different ranges of metals can be electrodeposited from aqueous, ionic liquid and DES electrolytes. These ranges have been summarised into Figure 2.7.

Although Figure 2.7 describes the range of possible deposition metals, it does not comment upon the quality of the electrodeposit, cost of deposition, electrolyte toxicity, or environmental concerns due to use and disposal.

Figure 2.7: Periodic table summarising the metals reported to have been electrodeposited from aqueous, ionic liquid, and deep eutectic solvents electrolytes [91], [114], [115]

There are many benefits and limitations of using current aqueous electrolytes, or more novel ionic liquid or DESs:

- **Aqueous electrolytes** are limited in the number of deposition metals due to their narrow electrochemical potential windows. However, they are widely utilised and highly developed.

- **Ionic liquid electrolytes**, as a whole, provide the largest range of deposition metals, but are expensive, and require much further development to overcome their limitations.

- **Deep eutectic solvent electrolytes** offer less deposition metals than ionic liquids, and theoretically more than aqueous electrolytes, are cheaper than ionic liquids, but more expensive than aqueous electrolytes, and require further development to overcome their limitations.

To encourage industrial viability, DESs must outperform current aqueous alternatives. As aqueous electrolyte formulations are highly developed, there is little possibility that novel electrolytes will outperform them solely evaluated upon the quality of the electrodeposit produced.

On the other hand, the formation of a novel electrodeposited metal or alloy; or electrodeposit with unique characteristics from a DES would significantly encourage their use.

2.1.6 *Tin electrodeposition from ionic liquid electrolytes*

Tin electrodeposition has been previously studied using ionic liquid electrolytes [116]–[126]. The electrolyte, tin salt, tin salt concentration, and deposition rate from some of these investigations are shown in Table 2.6. Moreover, although the majority of this work focuses upon the electrochemistry of these liquids, there are a few examples of deposit surface morphology shown in Figures 2.8 to 2.16.

Table 2.6: Electrolyte constituents and tin deposition variables for various ionic liquids

	Electrolyte	Tin Salt	Tin Salt Concentration	Deposition rate/potential	
A	Aluminium Chloride-1-methyl-3-ethylimidazolium Chloride	Tin(II) chloride	0.020 to 0.045 M	N/A	[116]
B	Zinc Chloride-1-ethyl-3-methylimidazolium Chloride	Tin(II) chloride	0.005 to 0.030 M	0.07 V (vs. Zn wire)	[117]
C	1-n-butyl-1-methylpyrrolidinium bis(trifluoromethylsulphonyl)imid	Tin(II) (anodically dissolved)	N/A	0.05 to 0.7 mA/cm^2	[119]
D	1-ethyl-3- methylimidazolium dicyanamide	Tin(II) chloride	0.050 M	−0.65 V, −0.75 V, and −0.84V (vs. Fc)	[120]
E	1-ethyl-3-methylimidazolium dicyanamide	Tin(II) chloride	0.050 M	−0.74 V and −0.84 V (vs. Fc) 0.6 and 1.2 mA/cm^2	[121]

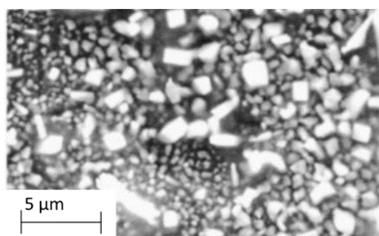


Figure 2.8: SEM micrograph of tin electrodeposit on nickel [117]. Table 2.6.B

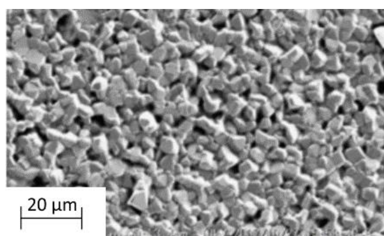


Figure 2.9: SEM micrograph of tin electrodeposit on copper for 20 h at 0.05 mA/cm^2 [119]. Table 2.6.C

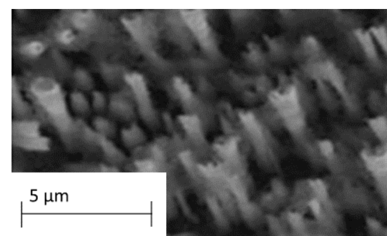


Figure 2.10: SEM micrograph of tin electrodeposit on copper foil at -0.65 V [120]. Table 2.6.D

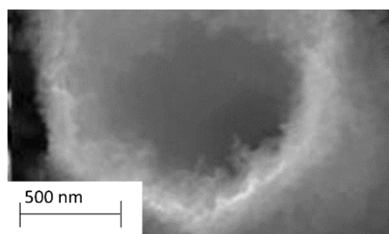


Figure 2.11: SEM micrograph of tin electrodeposit on copper foil at -0.65 V [120]. Table 2.6.D

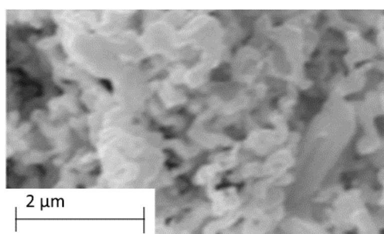


Figure 2.12: SEM micrograph of tin electrodeposit on copper foil at -0.75 V [120]. Table 2.6.D

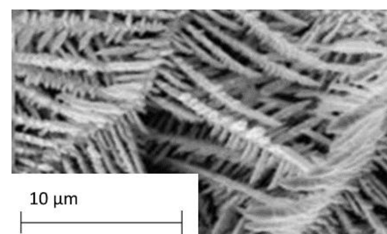


Figure 2.13: SEM micrograph of tin electrodeposit on copper foil at -0.84 V [120]. Table 2.6.D

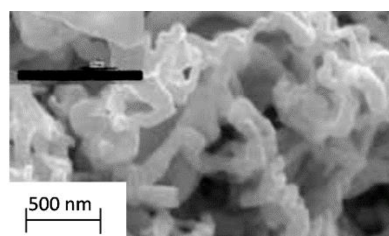


Figure 2.14: SEM micrograph of tin electrodeposit on copper for 1.2 C/cm^2 at -0.74 V [121]. Table 2.6.E

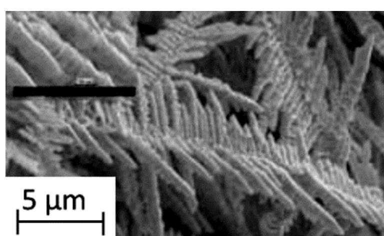


Figure 2.15: SEM micrograph of tin electrodeposit on copper for 1.2 C/cm^2 at -0.84 V [121]. Table 2.6.E

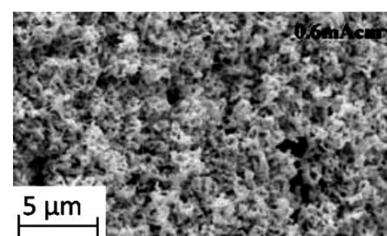


Figure 2.16: SEM micrograph of tin electrodeposit on copper for 2000 s at 0.6 mA/cm^2 [121]. Table 2.6.E

It is clear from Table 2.6 that tin reduction is possible, but the formation of a uniform deposit is difficult to achieve. Process C, from Table 2.6 and shown in Figure 2.9, could be the most promising in terms of uniformity, although of 2-5 µm in thickness, and is referred to as being “A smooth and adhesive electrodeposit with slight brightness” [119].

2.1.7 *Tin electrodeposition from DES electrolytes*

Tin electrodeposition using DES electrolytes has been undertaken due to their inherent benefits, but again these studies have focused upon the electrolyte electrochemistry rather than deposits formed [54], [79], [127]–[130]. The specific DESs used in these studies evolve around the choline chloride molecule with various hydrogen bond donors. These DESs are commonly referred to as: Reline, Ethaline, Propeline and Maline, as defined by Table 2.7.

Table 2.7: Compositional constituents of choline chloride based DES

Name	Reagents	Molar ratio	Viscosity at 24˚C (cP) [54], [131]	Viscosity at 75 ⁰C (cP) [129]
Ethaline	Choline chloride : Ethylene glycol	1 ChCl:2 EG	36	16
Propeline	Choline chloride : Propylene glycol	1 ChCl:2 PG	89	20
Reline	Choline chloride : Urea	1 ChCl:2 U	632-750	167
Maline	Choline chloride : Malonic acid	1 ChCl:2 M	1124	

These more common DESs reduce their cost of production since the reagents are used within other large quantity applications, i.e. Choline chloride is a constituent of chicken feed; urea is a component of fertiliser, propylene glycol is used for the manufacture of polyester resins and within the food industry; ethylene glycol is used to create polyester fibres; and malonic acid is used within the pharmaceutical, cosmetic, industrial and food industries. Due to these components being used in large quantities, there are significant cost savings due to their mass production and known methods of disposal. Reports which have focused in part upon the electrodeposition of pure tin from these solutions are seen in Table 2.8 with corresponding deposit morphology as shown in Figures 2.17 to 2.22.

Table 2.8: Electrolyte and tin deposition variables for various DESs.

	Electrolyte	Tin Salt	Tin Salt Concentration	Deposition rate/potential	
A	Reline	Tin(II) chloride	0.05 M	10 mA/cm^2	[127]
B	Ethaline	Tin(II) chloride	0.10 M	0.5 V (vs Li^+/Li)	[130]
C	Ethaline	Tin(II) chloride	0.06 M	-1.7 V (vs CE(ChCl))	[79]
D	Ethaline	Tin(II) chloride			[54]
E	Maline and Ethaline	Tin(II) chloride	0.05-0.5 M	1 to 60 mA/cm^2	[128]
F	Ethaline, Propeline and Reline	Tin(II) chloride	0.05 M	-1.25 V (vs CE(ChCl))	[129]

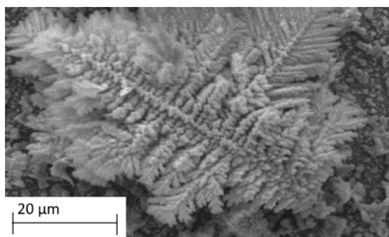


Figure 2.17: SEM micrograph of tin electrodeposit on copper for 30 min in Reline at 10 mA/cm^2 [127]. Table 2.8.A

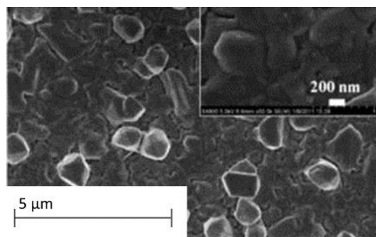


Figure 2.18: SEM micrograph of tin electrodeposit on copper foil for 15 min at 0.5 V in Ethaline [130]. Table 2.8.B

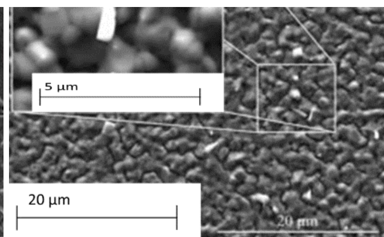


Figure 2.19: SEM micrograph of tin electrodeposit on a platinum electrode for 10 min in Ethaline at -1.7 V [79]. Table 2.8.C

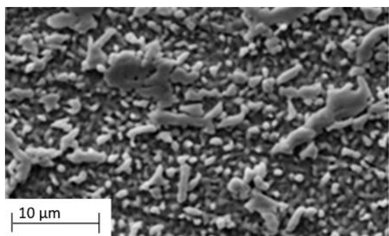


Figure 2.20: SEM micrograph of tin electrodeposit on copper for 20 min in Ethaline at 9 mA/cm2 [128] Table 2.8.E

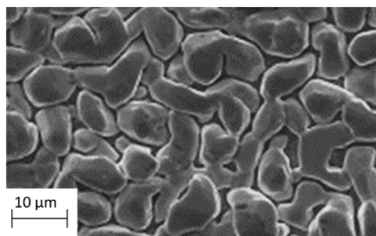


Figure 2.21: SEM micrograph of tin electrodeposit on copper for 15 min in Maline at 4 mA/cm2 [128] Table 2.8.E

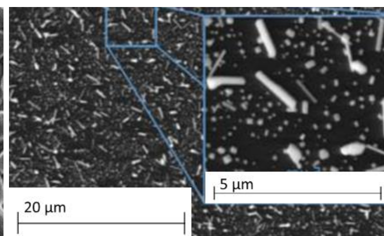


Figure 2.22: SEM micrograph of tin electrodeposit on a glassy carbon electrode for 60 second in Reline at -1.25 V [129]. Table 2.8.F

From Figures 2.17 to 2.22, it is clear, much like the ionic liquid electrolytes, the creation of a uniform or bright deposit is difficult, with many of the deposits featuring either fern- or needle-like dendritic growth. Moreover, since deposition has been undertaken galvanostatically, potentiostatically with various reference electrodes, and for a range of tin concentrations, deposition times, deposition rates, deposition potentials, amounts of charge passed, resulting cathode current efficiencies, and deposit thicknesses, it is difficult to define any trend in electrolyte or process parameters which produce uniform deposits.

There appears to be little work attempting to vary common deposition parameters to optimise deposit quality. As the purpose of electrodeposition is the production of a useful electrodeposit, deposition from DESs must attempt to meet the ASTM tin electrodeposition standards [25]. These deposits must be homogeneous, of uniform thickness, without defect when view at 4x magnification, and within a specific minimum thickness for select applications.

2.2 Tin whisker growth

Research in the growth of tin whiskers from electrodeposits has been carried out for over 60 years [13], and, although a sufficient mitigation strategy was found, it no longer passes modern European Union regulations [16]. This mitigation strategy was based on the deposition of tin-lead rather than pure tin, but since lead is to be a strictly controlled substance, whisker mitigation research has resumed, searching for a second mitigation method.

Whiskers on the surface of a tin electrodeposit, when used as a coating in modern electronics, are a source of reliability issue [132], [133]. Although hillocks, a form of tin whisker, are not a source of electrical shorting due to their restricted length, long filamental whiskers are. The concern with respect to whiskers producing electrical shorts is accentuated within the areas of high reliability, long lifetime electronics; areas of manufacture which cannot afford to leave tin whiskers unmitigated.

Although a whisker may be bridging two electrical contacts, the effect of this bridge may vary. Sometimes the layer of oxide on a whisker will inhibit shorting, but provided there is sufficient electrical potential, whiskers can become conductive [132], [134], [135]. This either creates an intermittent or long term connection where the whisker is left intact, or the whisker can short the circuit and then melt leaving no evidence of a fault mechanism. Moreover, in extreme cases of high voltage and sustained currents it is possible for a whisker to, not only melt, but vaporise and turn into highly conductive plasma, resulting in extensive damage [136]. Regardless of the extent of damage caused, be it an intermittent fault, single component damage or a system failure, whisker growth is of concern.

2.2.1 *What are tin whiskers?*

Whiskers are protrusions which form and grow on a metal coating after deposition. In common usage there are several named morphologies of whiskers, with examples highlighted in Figure 2.23.

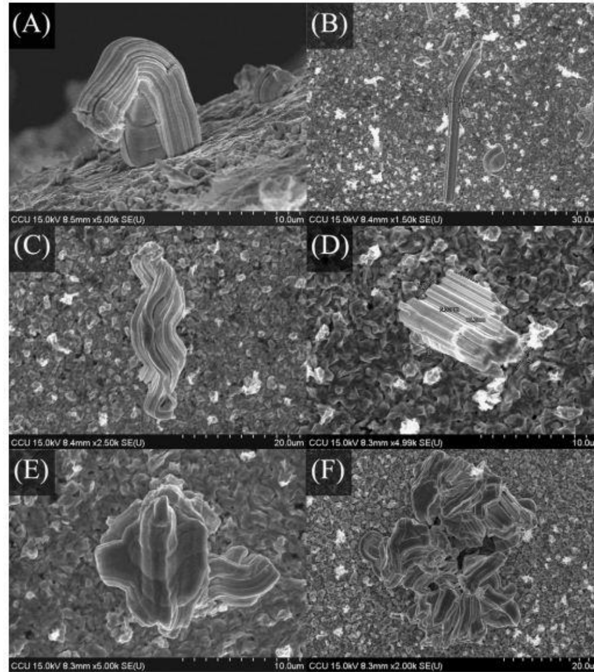


Figure 2.23: SEM micrographs showing various whisker shapes: (A) toothpaste-like; (B) filamental; (C) roll-like; (D) short filament-like; (E) nodule-like; (F) irregular-shaped whiskers [137]

Of these geometries this thesis separates them into broader definitions: filamental whiskers, which are of concern within the electronics industry; and hillocks which, due to their short lengths are not of interest. Therefore for this thesis the term whiskers refers to any feature which could cause an electrical short, filamental whiskers, and a hillock refers to other forms of post-deposition growth.

Other metals have also been known to form with examples being seen on aluminium [138], bismuth [139], cadmium [140], gold [141], [142], tungsten [143], and zinc [144], [145]. An extreme example of zinc whisker formation on a ferrous substrate is shown in Figure 2.24.

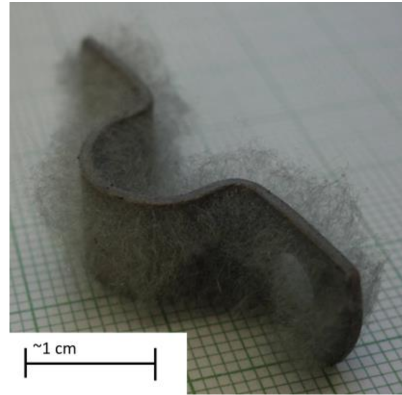


Figure 2.24: Zinc whisker growth on zinc coated iron (scale unspecified in original image) [146]

Terminology referring to whiskers does not always refer to metallic features which form post deposition, rather the terminology solely refers to feature geometry [147]–[149]. For example, the formation of sodium chloride in [150], and lithium fluoride in [151] are referred to as whiskers due to their geometry. It is for this reason that care must be taken with terminology. One such confusing report is the electrodeposition of tin from ethylene glycol which refers to dendritic structures as whiskers [152].

2.2.2 *Filamental whiskers*

Filamental whiskers are metal filaments which grow to lengths between 5 µm and 10 mm in length, 1 µm to 5 µm in diameter and often with a constant cross section [8], [152]. Additionally, they have been seen to be: completely straight or having many kinks and curves; protruding at the surface of the deposit at almost any angle; formed from a single or multiple grains; and able to emerge from any given area of the deposit including hillocks and other whisker bases as shown in Figure 2.25.

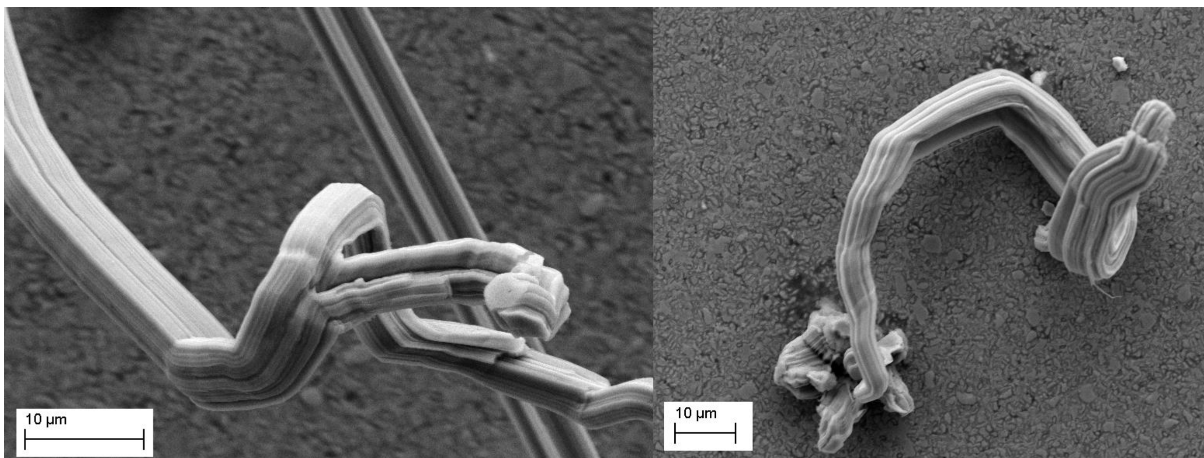


Figure 2.25: Tin whiskers grown from 2µm thick tin electrodeposits on brass substrates featuring filamental growth from nodules and other whisker bases.

2.2.3 *A brief history of whiskers*

Whiskers have been seen on multiple metal finishes and were first documented on cadmium coated plates in capacitors in 1946 [8]. It was then in 1951, the Bell Telephone Corporation reported cadmium whiskers on their filters for multi-channel telephone lines; thus starting the first long-term studies of whiskers on metal finishes [8].

Then, in 1959, lead was seen to be beneficial in whisker mitigation in electroplated tin by alloying during electrodeposition [8]. Lead at 3-10% by weight was commonly alloyed into the tin coating resulting in little change in the properties of the electrodeposit and only experiencing whisker growth under high compressive stresses. The addition of lead was a simple and effective method of preventing whiskers and as such remained the primary mitigation method for over 50 years [86].

Although tin-lead electrodeposits were commonplace, some components were still being coated with pure tin. It was noted in 1998 that some resistors and capacitors on the market were reported with very few whisker failures while still using pure tin as a finish for over 35 years [8], [26]. It was found that they had been created incorporating a nickel underlay [8], [153]; a heat treatment [154]; or had been alloyed with bismuth [8], [155].

This surprising result concerning the low levels of faults over 35 years for pure tin coatings can be attributed to several different reasons:

- Although these alternative methods to lead alloying are insufficiently effective on their own, they were sufficiently effective when combined together.
- Analysts were not looking for whiskers as a source of faults, and as such were less likely to attribute them to failures.
- Only a small amount of components were coated with pure tin, reducing the probability of failure by lessened surface area.
- PCB design featured large component-component gaps which limited whisker shorts

Previously, due to the extensive use of tin-lead as a coating the risk and reported levels of whisker failures was considered negligible. But, now that almost all of the tin coatings are lead free the risk factors increase to a concerning and, more importantly, unknown level. This therefore

causes anxiety for high reliability, long lifetime electronics manufacturers where risk calculation and minimisation is vital.

2.2.4 *Tin whisker caused failures*

There are many examples of whisker related electrical failures or 'events' within valuable electrical systems. Moreover, in addition to failures which are definitively attributed to whisker growth, there are predicted to be many more which go unreported. A few examples of tin whisker failures in electrical products are shown in Table 2.9. Furthermore, the list within the common public domain of reported whisker failures only features approximately 50 occurrences from 1986 to 2006 [156].

Table 2.9: Tin whisker reported failures [156]

Year	Product/Industry	ref.
1986	Pacemaker	[157]
1995	GALAXY IIIR satellite	[158]
1999	SOLIDARIDAD 1 satellite	[158]
2000	Raytheon missile	[159]
2002	Relay failure in military airplane	[160]
2002	Nuclear power plant power supply	[161]
2005	Nuclear power plant PCB	[162]
2005	Nuclear power plant PCB	[163]

Although the rate of reported failures appears low this may lead to a false sense of security as these values do not take into consideration: the number of whisker failures which are undiscovered; the number of whisker failures which are discovered yet unreported for the sake of company repercussions; and that these reported failures are prior to the implementation of RoHS in 2006. Therefore it would be expected, unless additional whisker mitigation methods are implemented, whisker related electrical device failures will increase significantly compared to this historic list of failures.

In more recent years failure study of the electrical shorts of electronic throttle control systems in Toyota cars was undertaken. These systems were found to malfunction, causing the car throttles to be fully opened without aid from the driver. Reports were initially undertaken by the American National Highway Traffic Safety Administration (NHTSA) followed by a report by NASA, requested by the NHTAS [164], [165]. These reports found whiskers growing between electrical contacts on throttle control systems, as seen in Figure 2.26

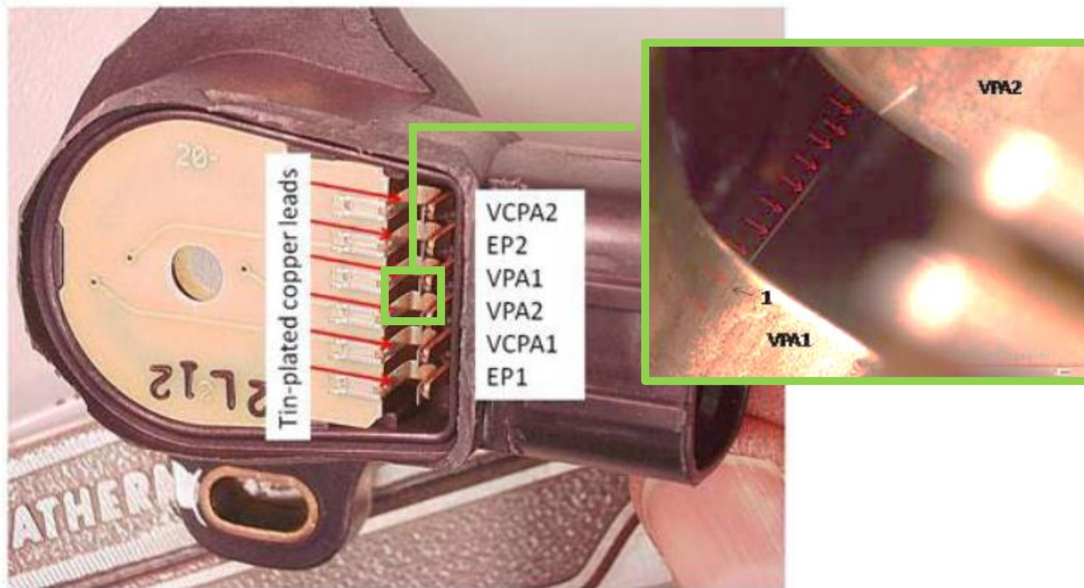


Figure 2.26: Disassembled Accelerator Pedal Assembly potentiometer featuring tin whisker from tin electroplated copper leads [165]

Within the NASA report tin whiskers are attributed as a possible cause for the malfunction by shorts between contacts VPA1 and VPA2 leading to an open throttle, but within the NHTAS report the risk due to shorting between these contacts is considered low; If the short caused by whiskers did occur the full releasing of the throttle would lead to a closed throttle regardless of the presence of an electrical short. Therefore, the cause of the malfunction of uncontrollable throttles was not attributed to tin whisker growth, but the transformation of a variable resistor to one with only fully closed or fully open positions were.

2.2.5 *Whisker related industry standards*

To unify industries manufacturing and utilising lead free tin coatings and solders afflicted by tin whiskers the Joint Electronic Device Engineering Council (JEDEC), international Electronics Manufacturing Initiative (iNEMI), and Interconnecting and Packaging electronic Circuits (IPC) undertook the publication of three documents: JESD201 entitled “Environmental Acceptance Requirements for Tin Whisker Susceptibility of Tin and Tin Alloy Surface Finishes” [166]; JP002 entitled “Current Tin Whiskers Theory and Mitigation Practices Guideline” [167]; and JESD22-A121 entitled “Test Method for Measuring Whisker Growth on Tin and Tin Alloy Surface Finishes” [168].

JP002 attempts to summarise the prevailing theories associated with the formation and growth of tin whiskers, providing practical mitigation methods which can be used by supplier and user. Whereas, JESD201 intended to provide applicable methodologies for testing and reporting tin whisker growth from tin and tin alloys surfaces, and is to be used alongside JESD22-A121.

These standards increase the awareness of tin whiskers within the electronics industry, and lessens fears related to the effects of the enforcement of RoHS directive. Nevertheless, many of the mandates declared within these documents can be debated, e.g. the effectiveness of thermal cycling as a whisker acceleration method, but primarily the dominant inaccuracy with respect to this thesis is the referral to many coatings within too broad a classification, e.g. bright or matte tin coatings. Tin whisker propensity varies within each classification dependent upon electrolyte formulation and deposition parameters, dismissing some tin coatings and elevating others, regardless of whisker mitigating ability.

2.2.6 *Whisker growth theory*

Work has been undertaken attempting to explain and model whisker growth in terms of its mechanism, growth rate, and various electrodeposit characteristics [77], [81], [169]–[175]. Although some of these theories have been disproven, there is still a desire to understand how and why whiskers grow. Moreover, some of the current theoretical models are difficult to validate and provide limited benefit of practical whisker mitigation. Therefore, although this area produces impressive results, their worth measured by applicability requires consideration. For this reason, this section focuses solely upon basic whisker theory founded on thermodynamics.

2.2.6.1 *Compressive stresses*

Early on in whisker research, compressive stresses were known to be a necessary factor in their formation [77], [174], [176]. Compressive stresses in surface finishes have been reported to be generated by several methods: electroplating processes [50], [177], IMC [50], [178], thermal cycling [179], substrate-electrodeposit thermal mismatch [180], surface corrosion [181], mechanical bending [155], and external pressure [182]. Of these, the most consequential are stresses from [183]: IMC formation; mechanical stresses from manufacture; and the thermal mismatch between substrate and coating. This therefore suggests an answer to the question; why whiskers grow? To relieve compressive stresses. Furthermore, this concept has been empirically proven by reduced deposit stresses evident around whisker bases caused by stress relaxation [184], [185].

2.2.6.1.1 *Whisker growth thermodynamics*

Calculations concerning whisker growth thermodynamics are shown (Chapter 10, Appendix 10.3 Thermodynamics – whisker growth by stress relaxation, page 182). These calculations are based upon energy requirements of additional free surface area supplied by compressive stress relaxation within the deposit. Resulting from this, Equation 2.7 is created expressing that as compressive stresses increase, whisker radius and/or whisker length are able to increase. Therefore this would

suggest, given a constant value of energy due to relaxation, thinner whiskers result in having longer lengths.

$$\Delta E_R \geq \gamma 2\pi r_w L_W \qquad \text{Equation 2.7}$$

Where: ΔE_R is the energy change due to deposit relaxation; γ is the free surface energy for tin; and r_w is the radius of a whisker; and L_w is whisker length. Taken from Chapter 10, Appendix 10.3 Thermodynamics – whisker growth by stress relaxation, page 225

Additionally, calculations within the appendix attempt to quantify the area of deposit undergoing relaxation for a given whisker length and radius. As a result they show that whiskers must be formed by long range diffusion and relaxation over a significantly large area of deposit to form such relatively large volume structures [186]–[188].

Furthermore, as the diameter of a whisker increases the energy held within its free surface area per unit length and volume per unit length increases, but at different rates. As a result, larger radii whiskers require lower energy densities to form as their volume to surface area ratio is lower than for smaller radii whiskers. Since the energy density of mass within the deposit must be less than that within the whisker for mass to pass between the two the deposit requires a threshold strain energy density within the deposit as described by Equation 2.8

$$\rho_1 \leq \frac{-2\,\gamma}{r_w} \qquad \text{Equation 2.8}$$

Where: ρ is strain energy density

This highlights a significant feature of whisker thermodynamics, although there may be enough energy within a large volume of deposit to form a long filament whisker, the localised strain energy density at its base may be too low to initiate/continue its growth.

2.2.6.1.2 *Cracked oxide theory*

Both filamental whisker and hillock formations must conform to the above thermodynamics, but from Equation 2.8, small radii whiskers require lower strain energy densities within the deposit compared to larger ones. Therefore, this states that as compressive stresses increase within a deposit, whiskers with large radii should form prior to thinner ones, but this has not been reported.

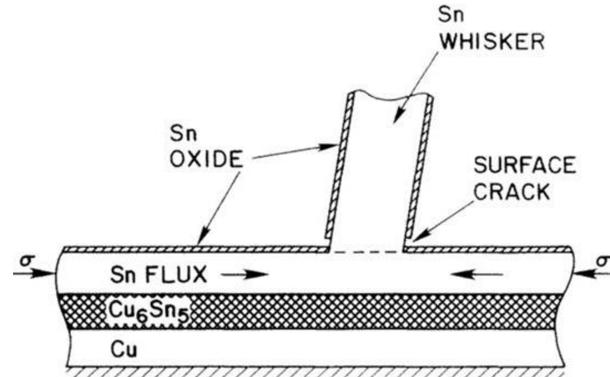


Figure 2.27: Schematic of cracked oxide layers facilitating whisker growth due to compressive stresses [173]

The theory known as 'cracked oxide' was first coined in 1994 [173], it states that an oxide formed on the surface of tin can crack, producing a low energy path way for whisker growth, as seen in Figure 2.27. Furthermore, not only does the cracking of the oxide encourage whisker growth, but it is also provides explanation for why thin filamental whiskers grow.

Expansion of a compressed free volume normally experiences relaxation at all points, but if there were to be a significant energetic cost in surface oxide deformation and fracture, the minimisation of energy could result in oxide failure along a single line of smallest length and expansion solely at this point. Therefore, although larger radii whiskers are energetically favourable per unit volume of relaxation, they feature larger circumferences. It is these large circumferences which require larger energies to crack oxide, thus restricting their growth and encouraging long thin whisker formation.

2.2.6.1.3 *Intermetallic compound growth*

At the interface, diffusion of copper from the substrate into the tin coating takes place resulting in the formation of Cu_6Sn_5, as seen in Equation 2.9, at room temperature (η'–phase IMC) and, with sufficient copper concentration and temperatures above approximately 60˚C, the formation of Cu_3Sn [180], [189], [190].

$$6\text{Cu} + 5\text{Sn} \leftrightarrow \text{Cu}_6\text{Sn}_5 \qquad \text{Equation 2.9}$$

Growth of η'-phase IMCs with a wedge shaped morphology at the tin grain boundaries has been reported to result in localised expansion; thereby generating compressive stresses within the tin electrodeposit leading to whisker formation [50], [80], [191]. Further evidence of the link between IMC growth, compressive stress increase and whisker formation is evident from the time dependencies of each feature involved, i.e. increasing amounts of IMC leads to increased

compressive stress and whisker growth [189], [190], [192], [193]. Therefore, showing a correlation between IMC growth and whisker formation Cu_6Sn_5. Furthermore, IMC growth has been shown to be dependent upon deposit thickness with increasing deposit thickness leading to increasing IMC growth [174].

The volume change produced due to IMC formation is debated. It is believed that due to copper's greater diffusivity, compared to that of tin, diffusion is primarily undertaken by copper atoms into the tin coating [190]. If this were not the case, and tin and copper were diffusing equally, there would be a resulting 5%.vol contraction, leading to a tensile stress within the deposit. Consequently, this contraction was disproven by measured compressive stresses of around 8-10 MPa produced by IMC growth [50]. Therefore this results in the diffusion of copper into tin as described by Equation 2.10 and additionally by the more energetically favourable Equation 2.11.

$$6\mathrm{Cu} + 11\mathrm{Sn} \rightarrow \mathrm{Cu_6Sn_5} + 6\mathrm{Sn} \quad \text{Equation 2.10}$$

$$12\mathrm{Cu} + 11\mathrm{Sn} \rightarrow 2\mathrm{Cu_6Sn_5} + 1\mathrm{Sn} \quad \text{Equation 2.11}$$

These equations result in considerably large expansion factors of 1.21 or 1.43 per unit volume of IMC formation [50], [192]. Evidence to confirm copper diffusion from the substrate has been reported by the formation of Kirkendall zones within the copper substrate in close proximity to the coating as shown in Figure 2.28 [194].

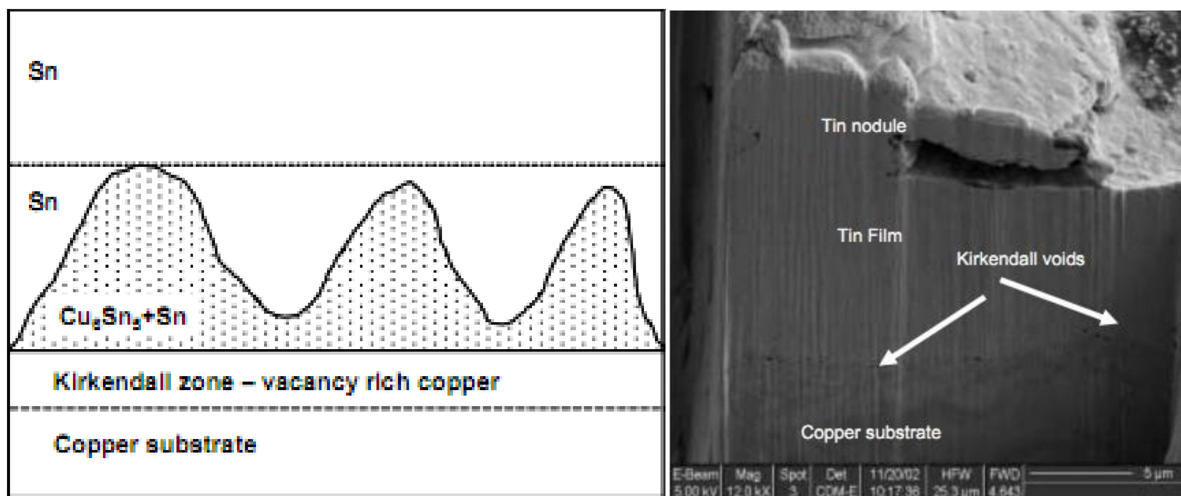


Figure 2.28: Schematic and SEM micrograph of tin electroplated copper substrate expressing Kirkendall zone formation [194]

Therefore the distribution and morphology of IMCs and their role in whisker growth are important features, requiring simultaneous assessment for individual deposits. Commonly, the study of IMC growth is achieved with the use of cross-sectioning, either mechanical, which is time intensive, or by using focused ion beam milling (FIB), which provides a limited area for analysis. Both

of these techniques are unable to show explicitly the distribution of IMCs at the interface in more than two–dimensions [80], [195]. A method that overcomes these limitations is the chemical removal of tin, which enables the top-down three–dimensional analysis of IMCs over more extensive areas [191].

Current chemical tin stripping techniques often rely upon single use solutions containing oxidising salts to achieve coating removal [6]. Highly acidic nitric acid solutions are frequently used for this process, thus requiring neutralisation and precipitation followed by landfill after use [196]. Moreover, the valuable metal and chemical content of these landfilled liquids is lost during disposal. Therefore, due to the financial and environmental cost, and increasingly stringent legislation concerning chemical landfill, alternative methods are desired [196], [197].

2.2.7 *Whisker propensity measurements*

To evaluate the extent to which a coating produces whiskers, empirical measurements are required. Whisker growth is often evaluated by: the density of whiskers [153], [168], [198], [199]; the length of the longest whisker [76], [200], [201]; or distribution of whisker lengths [132], [139], [153], [155], [199] with whisker lengths being measured in two or three dimensions [153]. Moreover, on rare occasions, whisker growth over a time period is condensed into a single value [202]. Additionally, whisker growth is measured relative to the time period from electrodeposition.

Due to whisker growth being evaluated based on counting, its statistical error is formed upon the total number of counted whiskers. Therefore, for reliable results many whiskers must be counted as error is equal to the square root of the number of whiskers counted. This then results in a time intensive counting process, requiring many whiskers to be measured or counted to ensure reliability.

As whiskers cause electrical shorts because of their lengths, measurement of whisker density is of little use on its own as it does not comment upon this feature. On the other hand, longest whisker or average whisker length is of little use when calculating risk without a measured density value. Therefore, the most powerful means of whisker growth evaluation is in the measuring of a large number of whiskers in three dimensions over a sufficiently large area. To achieve such results requires significant effort, but by doing so the distribution simultaneously contains information such as longest whisker and whisker density. The caveat to the desire of obtaining large statistical quantifiable evidence is, if all that is required to be evaluated is a qualitative result, to say that deposit ‘A’ produces far less whiskers than deposit ‘B’, then a macroscopic image of each deposit will often suffice; significantly reducing data acquisition.

Additionally, literature describes various measurement techniques, and as such the comparison between results becomes difficult. Moreover, whisker length is often measured using various definitions of length, and whisker counting requires a definition for what constitutes a whisker or hillock. Therefore the comparison of deposits within a paper can have their whisker growth measured effectively utilising systematic measurements, but, comparison externally becomes difficult as measurement techniques vary between researchers.

2.2.7.1 *Distribution of whisker lengths*

Presenting the distribution of whisker lengths for a given deposit is by far the most informative means of representing whisker propensity. An example of the distribution of whisker lengths is shown in Figure 2.29.

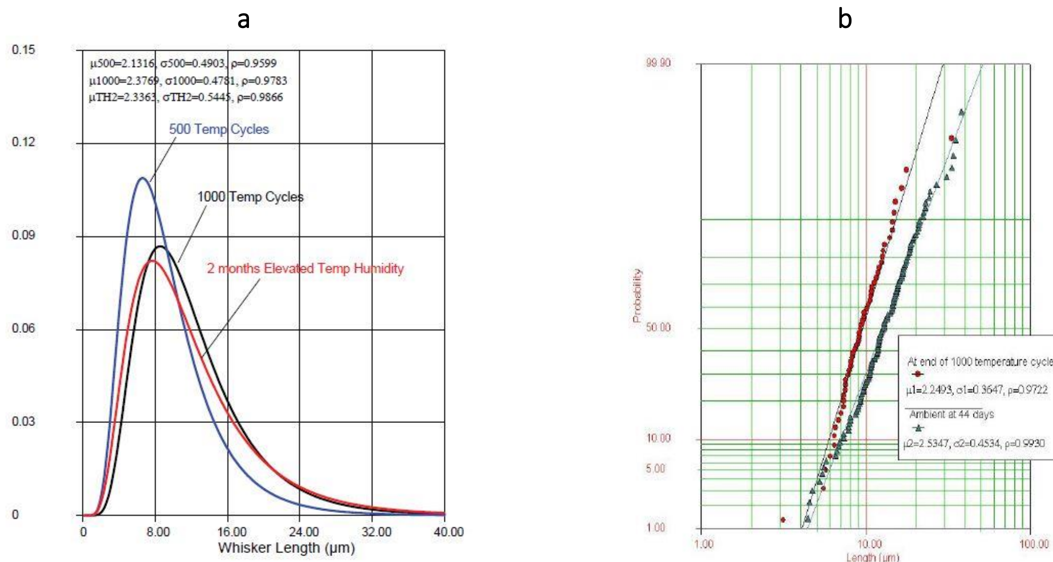


Figure 2.29; Length distribution of tin whiskers after vaious heat treatments. A: graph showing frequency vs whisker length; b: graph showing log-normal cumulative distribution of frequency vs whisker length [153], [203]

Figure 2.29, and other whisker length data, is said to have a lognormal distribution [139], [153], [203], [204]. Therefore, as lognormal distributions are based upon a multiplicative effect of normal positive variables, whisker length can be considered to be also [205]. This implies that the distribution of whisker length is created by multiple random variables interacting simultaneously.

2.2.7.2 *Time interval before whisker propensity measurements*

Individual whisker length, whisker length distributions, and whisker density have been shown to be time dependent [153], [174], [182], [191]. This highlights the importance of recording time periods between whisker measurement and deposition.

Individual whiskers have been seen to experience an incubation period and a range of constant growth rates as shown in Figure 2.30. Moreover, it is clear that incubation and whisker growth rates vary between whiskers.

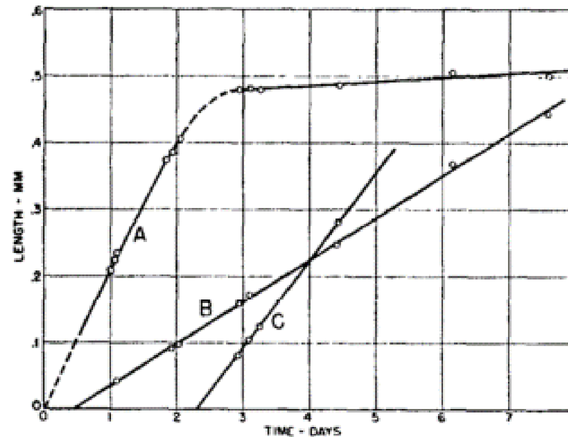


Figure 2.30: Individual whisker length with respect to time for three whiskers [182]

Therefore, as whisker growth does not decrease, a single measurement of deposit propensity, providing sufficient storage periods, can suffice for comparative propensity studies [183], [195], [206]. On the other hand, as whisker growth rates and incubation periods can vary, this concept fails if long enough storage periods are not observed, i.e. varying incubation periods could lead to significant growth after the final recorded propensity. Therefore, to deduce a deposit's whisker propensity over a long time period requires a long interval between deposition and measurement, or the use of an accelerating method.

Accelerative methods to reduce the storage time between deposition and examination is often undertaken, but these methods themselves manipulate other deposit characteristics leading to altered whisker growth [58], [203], [207]. Often elevated temperature and humidity are utilised, but these have been shown to effect whisker growth themselves and therefore cannot be considered effective accelerators [181], [199], [207]. Another method of accelerating whisker growth is by induced compressive stresses by mechanical bending [20], [182], [188], [208]. Although this process does not manipulate whisker growth as temperature or humidity does, it imparts stresses which may never be naturally experienced by the deposit, i.e. a given deposit may whisker under extreme compression but in regular storage, this would not occur.

This leaves whisker research with an issue, with no definitive acceleration method, how are deposits to be evaluated with respect to whisker propensity? The only answer to this problem is long storage periods which are highly unattractive to the field of research. Nevertheless, it is only deposit observation for the appearance of whiskers that requires long periods of storage, whereas in the

case of developing whisker mitigation methods independent of deposit characteristics, such as conformal coatings, short time periods can be utilised by the use of rapidly whiskering deposits.

2.2.8 *Whisker measurement methods*

Whisker propensity measurement is often a time consuming activity utilising high magnification optical or scanning electron microscopy (SEM) [76], [209]. The processes of whisker counting or whisker length measurement have different requirements. Whisker counting can be undertaken with low magnifications allowing the use of optical microscopy, whereas, whisker length measurements require two images at differing viewing angles to be achieved and often demands magnifications provided only by SEM [153].

The process of measuring whisker length is mostly undertaken using a single viewing angle. For this reason, due to the whisker growth direction, the length of a whisker when measured produces only positive values with negative errors between 0, and 100%. The resulting whisker length is therefore measured to be between its actual length and zero; a significant difference. It is for this reason that three dimensional analysis is required to obtain the actual length of the whisker.

2.2.9 *Whisker risk management*

Whisker growth is seen by many as another element of risk to be considered and controlled when evaluating probability of product failure [210]–[214]. This requires controls be put in place that minimise risk due to whiskers, but, as with any additional control measures there is a financial cost. For this reason, some areas of the electronics industry are not concerned with whisker failure control; whereas others are concerned to the point that pure tin coatings and solders have been banned at great expense. For ease of understanding, guidelines in GEIA-STD-0005-2 created the product classifications seen in Table 2.10 [215].

The role of risk assessment is to avoid faults by minimising failure probability. To be able to undertake risk calculations for a product the risk factor due to whisker failure must be known. Therefore mathematical modelling has been undertaken attempting to predict risk factors involved with using pure tin coated components and solders [132], [212], [213], [216]. These predictions are limited in their applicability as they predict growth factors for a single electrodeposited coating, whereas in reality different coatings whisker at varying rates, lengths and densities. Therefore, calculating a risk factor for whisker growth is beneficial, but portraying risk to be uniform for all components and coatings results in either insufficient or excessive cost due to the lack of or inclusion of whisker mitigation methods.

Table 2.10: Classification definitions of whisker control levels [215]

Level 1	Tin whisker risk is uncontrolled,
Level 2A	Control of tin whisker risk is exclusively by design rules such as increased pick size and critical component separation.
Level 2B	Control of tin whisker risk is predominantly by design rules, and in exceptional cases by tin avoidance
Level 2C	Control of tin whisker risk is predominantly by tin avoidance, and in exceptional cases by design rules
Level 3	Control of tin whisker risk is exclusively by tin avoidance
Too much control	excessive control levels are in use compared to the required risk levels incurring unrequired cost
Too little control	insufficient control levels are in use compared to the required risk levels incurring increased portability of failure

2.2.10 *Whisker mitigation methods*

Mitigation of whisker growth and their associated electrical failures can be undertaken by several means, each with their own strengths and limitations. These methods can be collected into two groups as shown in Table 2.11; whisker growth mitigation and whisker management. The growth mitigation methods are often interconnected, affecting each other and attempt to reduce the growth of whiskers, whereas the methods used for whisker management are unconnected and attempt to reduce failures after the whiskers have grown.

Table 2.11: Whisker mitigation methods

Whisker growth mitigation	Whisker management
Increasing substrate surface roughness	Circuit and component design
Use of a coating underlay	Conformal coating
Appropriate tin deposit selection	
Post-deposition annealing	

Furthermore, although the majority of mitigation methods offer a reduction in whisker risk, none of them reduce it to zero individually. Nevertheless, the combined use of all of these mitigation methods could reduce the whisker failure risk to an acceptable value, but to what value is still unknown.

2.2.10.1 *Substrate surface roughness increase*

IMC growth as a function of substrate roughness has previously been studied in connection with whisker growth on electrodeposited tin [82], [217]. These studies revealed that as surface roughness increased the IMC distribution became more uniform. This change in IMC morphology and distribution was accompanied by a reduction in whisker density, average whisker length and the length of the longest whisker.

2.2.10.2 *Use of an underlay coating*

The use of an underlay between the substrate material and tin coating is used to reduce inter-diffusion and IMC growth. It has been reported that for over twenty years an electrical component supplier was able to fully mitigate whiskers with the use of a nickel underlay beneath their tin electrodeposits [27]. More than likely this claim coincided with additional mitigation methods, but nevertheless is still impressive. Nickel is often a common underlay at thicknesses in the order of 10µm between the copper substrates and tin electrodeposits, although examples of silver have also been reported, and are both thought to reduced whisker propensities [153], [180], [218]–[220].

The use of an underlay is attempting to reduce whisker growth by inhibiting compressive stresses produced by IMC growth. In practice, grain boundaries within fine grained nickel underlays reduce its effectiveness by the presence of higher diffusion pathways. To further accentuate this diffusion barrier it has been shown that an additional underlay of 100-nm-thick $Zr_{46}Ti_{26}Ni_{28}$ has been beneficial in reducing whisker growth, as shown in Figure 2.31 [219].

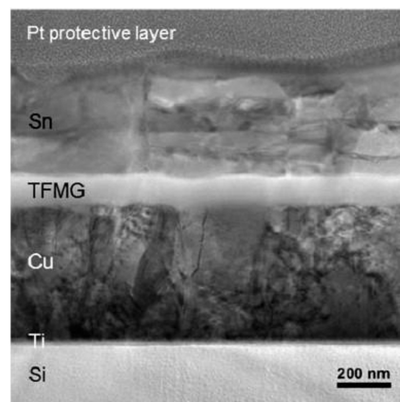


Figure 2.31 TEM cross-section of tin coating on copper with a nickel and $Zr_{46}Ti_{26}Ni_{28}$ underlay [219]

Conversely, whisker growth has been shown to be unaffected by nickel underlays [153]. Therefore these contrasting statements may be explained by work in [153] being carried out for two

years at elevated temperatures. Although nickel underlays reduce diffusion, they do not completely inhibit it, especially at elevated temperatures for long periods of time. Therefore, copper will, given sufficient time, reach the tin coating creating tin-copper IMCs.

Furthermore, the nickel underlay does not just inhibit the formation of IMCs, but additionally inhibits surface oxidation of the more reactive substrate constituents. An example of the resulting surface oxidation of a tin electrodeposited onto brass without the use of a nickel underlay is shown in Figure 2.32.

It was thought that the increased surface zinc oxide formation seen on these tin coatings was responsible for the increased whisker formations seen for the tin on brass substrates. Additionally, these investigations demonstrated that increased thicknesses of surface tin oxide reduced the formation of whiskers by inhibiting the zinc oxide formation. Therefore, highlighting the importance of substrate-coating interface reactions and coating-atmosphere reactions due to diffusions from the former interface.

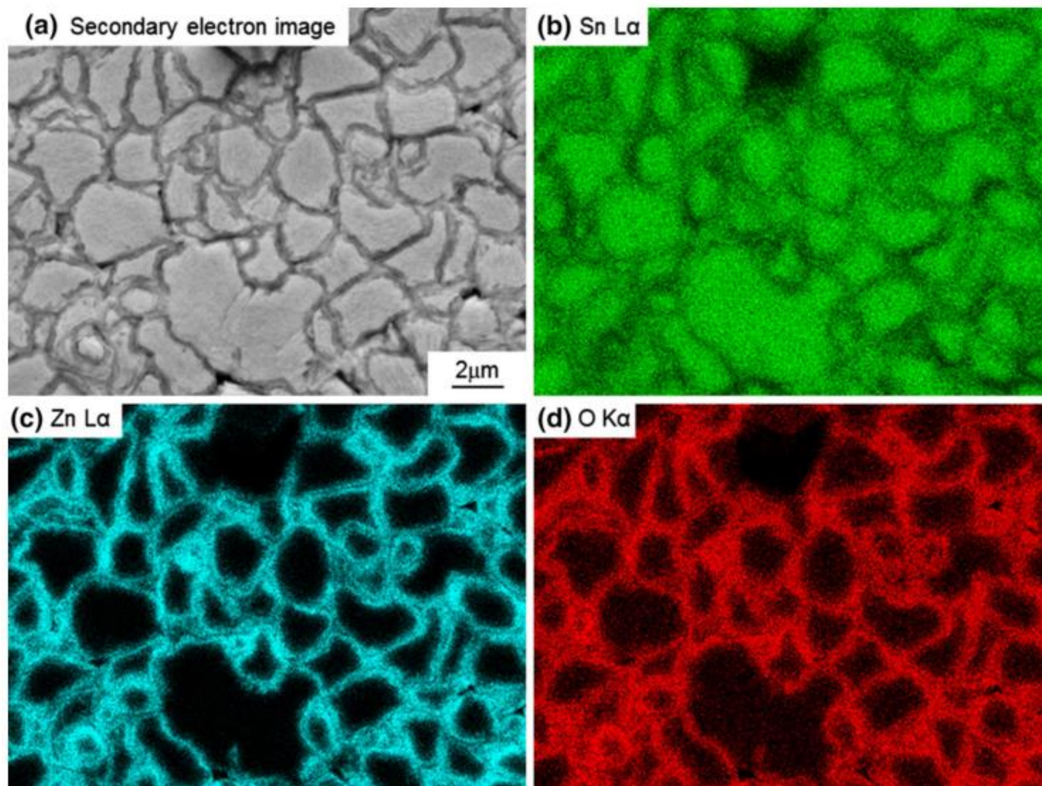


Figure 2.32: SEM analysis of a 5 μm tin deposit on brass electroplated at 5 mA/cm^2, 6 months after deposition: (a) secondary electron image, (b) Sn Lα x-ray map, (c) Zn Lα x-ray map and (d) O Kα x-ray map [207]

Although a nickel underlay operates as an effective diffusion barrier its continued use is of increasing concern. Nickel exposure in direct contact with skin can often cause dermatitis as it can

cause allergic reactions and is restricted by legislation when used for items such as piercings, jewellery or orthodontic work [221]. Furthermore, chronic nickel exposure is classed as a carcinogenic, and has been connected to cardiovascular disease, lung cancer, developmental deficits in childhood, neurological deficits, and high blood pressure [222]. Moreover, nickel exposure is also thought to cause liver and kidney damage, confirmed when studies were carried out on nickel electroplaters in Egypt expressing reduced liver function [223], [224]. Moreover nickel has been recognised as a toxin severely damaging reproductive health and can result in miscarriage, infertility, and birth defects [225], [226]. In addition, according to the International Agency for the Research of Cancer (IARC) nickel containing compounds used in electrodeposition are regarded as carcinogens [227]. Therefore, it is due to these medical concerns that nickel electrodeposition, and its resulting use as a diffusion barrier, is under threat.

2.2.10.3 *Appropriate tin electrodeposit selection*

Appropriate tin electrodeposits can greatly reduce the growth of whiskers through selection of the correct alloying element, proprietary electrolyte and deposition variables. An example of whisker growth affect by electrolyte formation is shown in Figure 2.33.

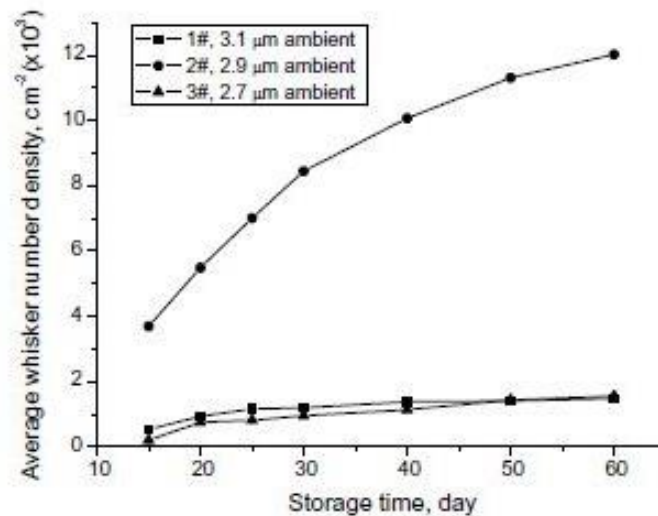


Figure 2.33; Whisker number density (>10 µm) as a function of storage for electrodeposits produced from three electrolytes. 1#: sulphate-based bath, 2#: alkaline stannate bath, and 3#: stannous chloride-based bath [76]

There are a range of whisker propensities produced by different electrolyte formulations which deposit pure tin. Although a large range of deposits have had their whisker propensities characterised, the lack of attributing this to specific electrolytes causes the comparison of the available formulations to be difficult. As a result, the lack of a ‘market survey’ for the whisker

propensity for each electrolyte formulation is not available it becomes difficult for electroplaters to select the best deposits.

Appropriate deposit thickness selection has additionally been shown to be significant. Results confirming this relationship between whisker density and deposit thickness are shown in Figure 2.34 [50], [153], [228]. Therefore thicker deposits often experience smaller levels of whisker growth allowing the selection of a large deposit thickness to perhaps be an effective mitigation method.

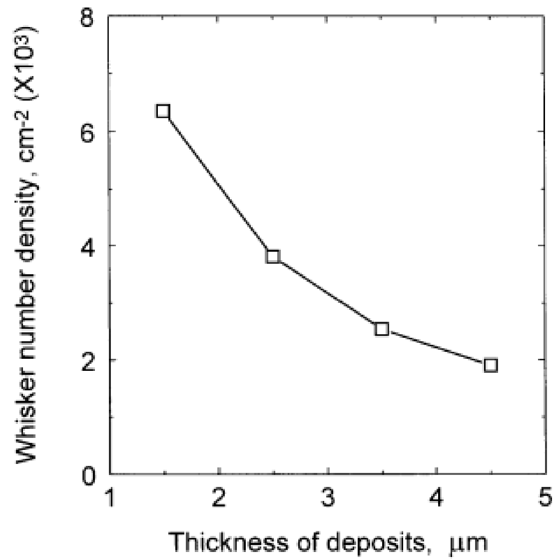


Figure 2.34 Number density of 20µm or larger whiskers grown from a tin deposit after 30 days as a function of deposit thickness [50]

Additionally, current density during deposition has been shown to affect whisker growth [50]. This is thought to be due to increased compressive stresses at varying current densities leading to elevated whisker growth [45]. But, it should be noted that these relationships are irregular, unpredictable, and unique to the specific electrodeposit and therefore cannot be applied generally for all deposits.

Moreover, as commented on during the review on electrodeposition, deposit grain size and orientation have been seen to be affected by deposition parameters and electrolyte formulation. Additionally, grain size [78], and grain orientation are factors affecting whisker growth [50], [57], [137], [175], [229]. But by manipulating tin electrodeposition to vary grain size or deposit orientation additional deposit characteristics often change, undermining the assumption that a sole variable directly affects whisker growth.

By the addition of another metal salt to the electrolyte an alloyed tin deposit is created featuring different deposit characteristics as well as a different elemental content. The effect which the

elemental contents and the deposit characteristics have upon whisker growth cannot be separated as they are interrelated. Nevertheless, some alloyed tin deposits are better whisker mitigators than others [202], [206], [229]–[233].

Data summarising the variations in whisker growth for various electrolyte formulations and different alloy compositions on various substrates are shown in Figure 2.35 and [234]. These results show that each alloy can be produced by a range of electrolyte formulations and that each of these formulations produced a different whisker propensity. Therefore it can be said that whisker growth is dependent upon interdependent variables of substrate material, electrolyte formulation, alloy content, and deposition parameters.

Moreover, data in Figure 2.35 measures whisker growth in complex units calculated from several whisker density measurements at multiple time periods reduced into a single number. Furthermore, substrates seen in Figure 2.35 referred to as alloy 42 and olin 194 comprise, Alloy 42 - Nickel 41%, carbon 0.05%, manganese 0.80%, phosphorus 0.025%, sulphur 0.025, silicon 0.003, chrome 0.25, aluminium 0.1%, iron remainder and; Olin 194 - copper 97.5%, iron 2.35%, phosphorus 0.03%, and zinc 0.12%.

Additionally it should be noted that although all of the current alloys that can be created from an aqueous bath have been evaluated for whisker growth, none of them feature whisker mitigation to the same extent as that of lead alloying.

It appears correct selection of the electrolyte formulation and deposition variables can result in reduced whisker propensities, but claiming that a resulting characteristic of these deposits is responsible for its whisker mitigation is dangerous. Warning must be given when commenting upon the interconnection between the electrodeposition process, deposit characteristics, and whisker growth.

Some electrodeposit characteristics are said to be interconnected, e.g. a universal relationship between surface finish and whisker propensity, suggesting that bright deposits produce elevated whisker propensities [235], whereas in reality some exhibit lower whisker propensities than some matte finishes [8], [236].

Moreover, realistically, during electrodeposition all of a deposit's characteristics are interconnected, causing manipulation of a single characteristic to lead to the variation of many others. In addition, although science is able to characterise several significant deposit features it is still limited to the analytical techniques available, leaving some to be unquantifiable. As a result, this produces a system, attempting to optimise single deposit characteristics to be unworkable as there

are too many interconnected deposit characteristics and unquantifiable characteristics. Therefore, thinking should aim to acknowledge inter-characteristic relationships but to not focus upon them, seeing each deposit as unique.

Nevertheless, although seeing each deposit as unique depicts an optimum system of relating deposition parameters (electrolyte formulation, current density, agitation, etc.) to whisker growth, the nature of intellectual property concerning electrolyte formulation hinders the open disclosure of possibly negative whisker growth propensities. This then leaves tin whisker mitigation, by appropriate optimised deposit selection, being able to examine many electrolyte formulations but not being permitted to disseminate results due to negative repercussions.

Work concerning whisker growth due to electrodeposition variables has focused upon the interconnections between various deposit characteristics and whisker growth. This is a very convoluted methodology attempting to understand all interconnected factors. Nevertheless, work aiming to evaluate individual deposits based upon their deposition parameters has produced a few clear examples which show some deposits to be good or bad whisker mitigators, often undertaken for non-commercially available formulations [76], [236]. The issue arises when deposits are discovered that produce minimal whisker growth accompanied by incomplete disclosure upon electrolyte formulation specifics [82], [202], [237].

Therefore, this leaves this form of whisker mitigation, appropriate deposit selection by electrolyte and deposition variable selection, to be considerably hindered. As a result all tin deposits are considered equally poor, when in reality there are is a range of whisker propensities for the tin deposits available. Equally this statement could also be applied to tin-lead alloys, suggesting that not all tin-lead deposits whisker equally as whiskers have been seen to form from some electrolyte formulations and not others [202], [238].

A few electroplating suppliers claim to possess pure tin coatings which form no whiskers over long time periods [239]–[241]. Some of which are supported by documented scientific comparative results made available by the supplying company, but as of yet there has been no independent review undertaken [242]. Moreover, these results compare only a few electrolyte formulations missing many possibly better depositing systems.

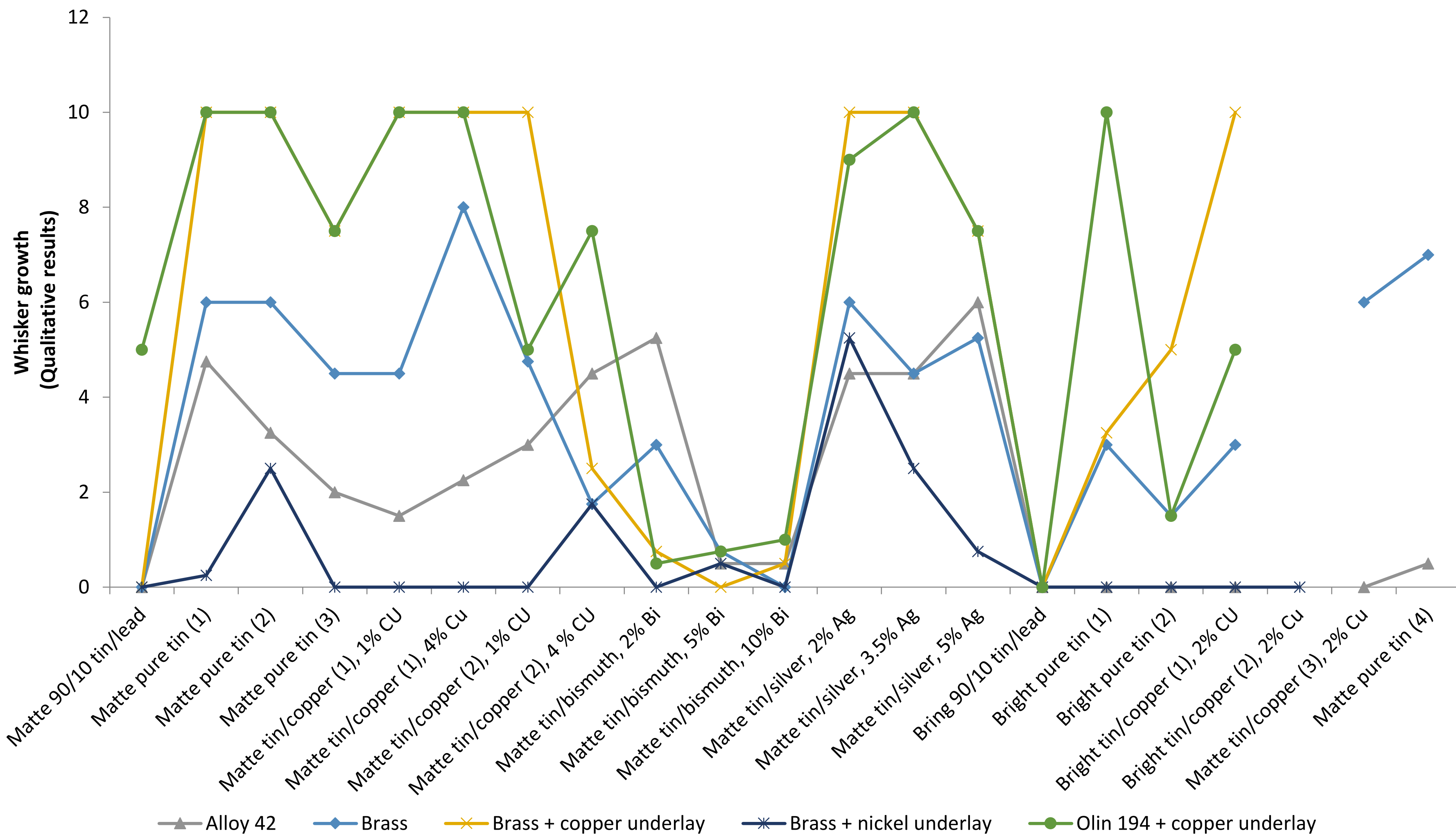


Figure 2.35: Whisker growth measured for various electrolyte formulations, alloy composition and substrate materials [202]. Breif description of whisker growth units is shown in text below.

2.2.10.4 *Post deposition annealing*

The annealing of a deposit at temperatures of approximately 150-200°C has been shown to reduce whisker growth [155], [199], [243]. This is thought to be due to compressive stress relaxation by elevated temperature. A technique similar in nature to annealing is referred to as reflowing, this uses temperatures above the melting point of tin, around 230°C, to cause solid tin coatings to melt effecting whisker mitigation [244]. Deposit annealing is not always beneficial as some bright tin coatings have been known to blister during this process, whereas this is not the case for matte deposits [27].

The temperatures used in annealing and reflow encourage stress relaxation by increased recrystallisation leading to a coarsened grain sizes, but in addition temperature is known to increase atomic diffusion rates, oxide formation and increase IMC growth [245]. Therefore, whisker propensity manipulations are difficult to attribute to a single factor. Additionally, although whisker propensity is commonly reduced, a few examples have been shown where annealing causes increase in, or does not affect, whisker growth [244].

These thermal treatments appear similar to that of the accelerating method for tin whisker growth. It is primarily because both methods occasionally show results affirming their whisker manipulation which encourages their usage, but it should be noted that conflicting results for thermal treatment have been reported, bringing into question their effectiveness.

2.2.10.5 *Circuit and component design*

Simply by increasing the gap size between components, increasing component pitch size, moving components with high risk away from system vital areas or simply by adding an insulating barrier between two connections, whiskers can be managed effectively [246]. However, this is not always possible; a large number of circuit boards are prefabricated thus denying bespoke whisker considered design. Moreover, whisker mitigating designs often reduce component density, which is contrary to the demands in electronics. Furthermore, whiskers have been known to grow to lengths beyond 10 mm [247], thus requiring pitch sizes exceeding this length; an impractical solution.

2.2.10.6 *Application of a conformal coating*

A polymer conformal coating aims to manage whiskers rather than preventing their growth. These insulating coatings aim to act as a barrier to a conductive whisker, allowing insulated connection inhibiting shorting [248]–[252]. This highlights two means by which conformal coatings may reduce unwanted electrical shorts, both of which are depicted in Figure 2.36.

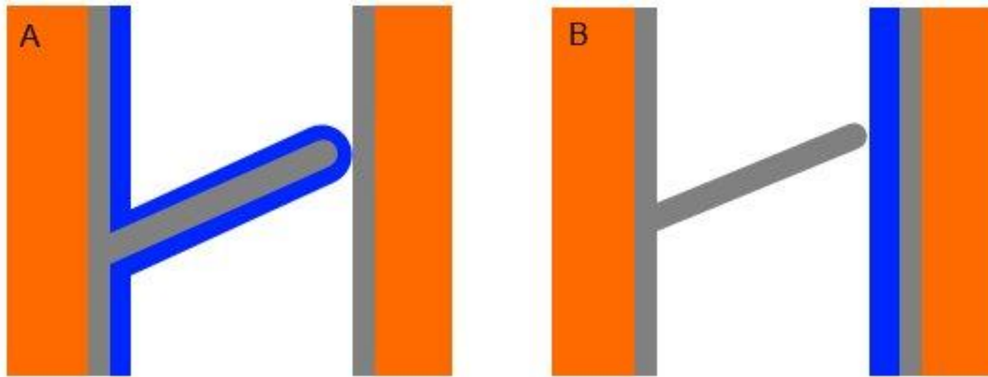


Figure 2.36: Schematic describing the mechanism by which a conformal coating inhibits whisker related failures. Featuring A: whisker growth underneath the conformal coating; B: conformal coating acting as a barrier 'outside' whisker growth

Figure 2.36 describes the two means by which a whisker is insulated while in connection with another electrical contact. Situation 'A' requires a conformal coating to be elastic enough to stretch around the whisker protruding from its own surface, whereas 'B' relies upon the whisker being unable to penetrate through the conformal coating on another surface. Situation A, as described by Figure 2.36, often results in the tenting of the conformal coating, or its complete failure leaving the whisker unaffected [252]. Whereas situation B can be far more reliable as it requires forces which are inhibited by the buckling of a long whisker [253].

Nevertheless, although conformal coatings are effective at managing whiskers they only do so at effective thicknesses. Whisker containment by conformal coatings has been shown to be improved by thicker coatings, up to 150 µm in thickness. Moreover, whisker containment has been reported to be dependent upon the material used, such as urethanes, acrylics, and silicones often used due their frequent use in alternative applications [249]. Furthermore, often when applying a conformal coating there is a thinning effect at the edges causing the conformal coating too thin to an extent that it is no longer effective [250], [253].

2.3 Discussion of literature

The means of tin whisker mitigation for tin was found historically in lead alloying, but now lead has been banned by EU legislation the electroplating and electronics industry is searching for an alternative. Due to the fact that the electrodeposition industry is not forthcoming concerning proprietary electrolyte formulations the open disclosure of the comparison of these coatings with respect to their whisker growth is rare. Therefore research into whisker mitigation is aiming to define which individual deposit characteristics are beneficial to identify which coatings should be used based upon these characteristics. This approach does not necessarily take into account the fact that each electrodeposit is unique, and engineering one characteristic often results in others

changing simultaneously. Therefore, each deposit requires empirical measurements to evaluate whisker growth.

This approach relies upon the open comparison of the deposits which, due to the secretive nature of the industry, does not take place. This results in whisker mitigation by deposit optimisation being difficult. Therefore whisker mitigation research continues by studying other mitigation methods such as nickel underlays, conformal coatings and circuit design. Therefore, to optimise the deposit an alternative direction of development is required, i.e. a step change, by the creation of a novel alloy or deposition process akin to the change experienced by the discovery of lead alloying.

The recent development of ionic liquids and DESs are one such medium that could offer additional alloys to electrodeposition and whisker mitigation research. By their increased potential windows, compared to that of water, they allow for the electrodeposition of a significantly increased number of metals. On the route to developing a novel tin alloy using these liquids, the electrodeposition of tin should be first studied to increase understanding of a simpler system prior to a more complex alloy formulation, as well as to see if the new electrolyte chemistry of these electrolytes produces an electrodeposit which exhibits differing levels of whisker mitigation.

2.4 Conclusion of literature

Electrodeposition utilising aqueous electrolytes is a highly developed and frequently utilised system but is limited in the number of metals it is able to deposit by a narrow potential window. Recent advances in non-aqueous electrolytes have produced liquids which are able to deposit tin and tin alloys but with limited deposit uniformity.

Whisker growth on tin electrodeposits cause increased risk of electrical shorting and as such are mitigated. Although whisker growth theory is incomplete, mitigation can be undertaken by several methods, including the selection of specific low whiskering coatings created from different electrolyte formulations.

The novel electrolyte formulations provided by the creation of DESs allows for the possibility of new tin electrodeposits produced from their use. It is because of their different electrolyte chemistries, compared to the traditional aqueous systems, and the variation of whisker growth due to different electrolyte formulations which require that deposits from these electrolytes be created and assessed upon whisker propensity.

2.5 Aims and objectives

This literature review shows each foundational subject, deep eutectic solvents and tin whisker mitigation, are well established and well-studied. The area which is less developed is the overlap between these two subjects. Therefore, efforts will be undertaken to achieve 3 objectives in this less developed area:

1. Create a uniform, homogeneous pure tin electrodeposit produced using a deep eutectic solvent.
2. Compare the electrodeposits produced from a deep eutectic solvent with those created from a proprietary aqueous electrolyte with respect to whisker propensity and other metallurgical characteristics.
3. To develop a means by which a tin electrodeposit can be removed by electrochemical means to allow characterisation of the Cu-Sn IMC and apply this technique to deposits produced from aqueous and non-aqueous electrolytes in objective 2.

Chapter 3 Experimental Procedure

The experimental procedure can be broken down into several stages; substrate preparation, aqueous electrodeposition using a proprietary pure tin electroplating bath, the electrodeposition of tin from DESs and tin electrodeposit anodic electrochemical oxidation. Many of these processes required either modification of existing techniques or the development of new methods. Therefore, in this chapter, the development process is presented, together with a description of the optimised techniques. The various surface analytical techniques to characterise the resultant electrodeposits are also outlined.

3.1 Copper substrate preparation

As-received copper substrates (99.9% purity, Advent Research Materials) were 100 µm thick featuring a rolled finish, as seen in Figure 3.1.A. To compare surface roughness effects at the coating-substrate interface, and to reduce polishing times, some of the substrates were electropolished. Samples were cut to 4 cm x 2 cm and were masked off using chemically resistant polyester tape (3M, 851 Greenback Printed Circuit Board Tape) leaving a single flat area of 4 cm^2 (2 x 2 cm) for experimentation. Electropolishing was undertaken potentiostatically in 50 vol.% phosphoric acid (S.G. 1.25, 98% purity, Acros Organics) using a potentiostat (Solartron ECI 1286) with counter and reference electrodes of 100 µm thick copper foil and a saturated calomel electrode (SCE), respectively. The polishing process was carried out at 1 V vs. SCE at room temperature for two time periods; 4 and 8 mins.

It is clear from Figure 3.1 (A and B) that a 4 min electropolish successfully removed the rolling lines present on the as-received material resulting in a smooth surface whilst an 8 min electropolish produced preferential grain etching (Figure 3.1.C); causing height differences between adjacent grains. From these results a time of 4 mins was selected for subsequent electropolishing of the copper substrates.

In addition, the electropolishing process produced a sufficiently smooth polish to allow crystallographic analysis by Electron Back-Scatter Diffraction (EBSD) to be undertaken. This then allowed the mapping, and the creation of an inverse pole figure, of crystallographic grain orientation as shown in Figure 3.2.

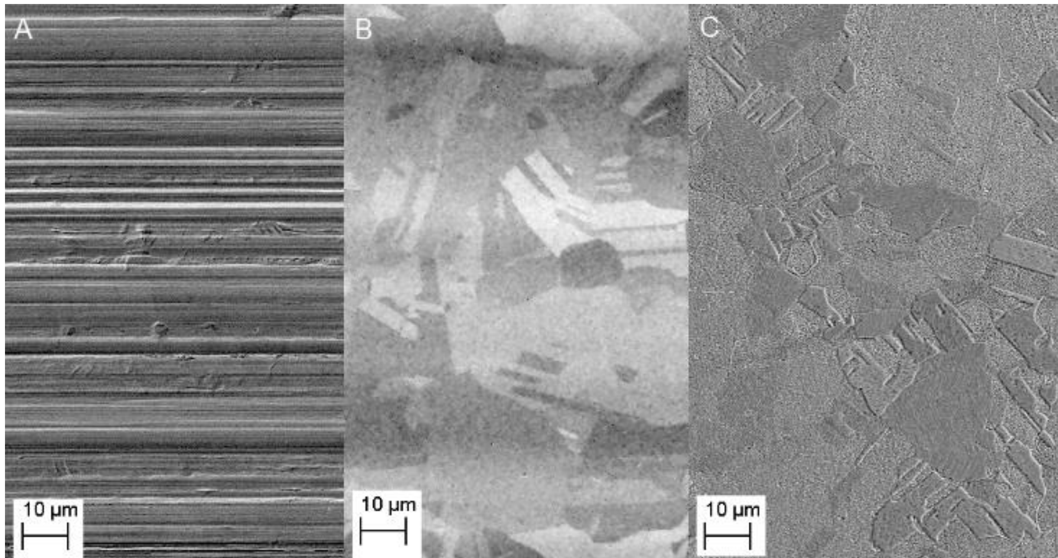


Figure 3.1: SEM micrographs showing copper substrate surfaces; 'A' as–received, 'B' electropolished for 4 mins, 'C' electropolished for 8 mins

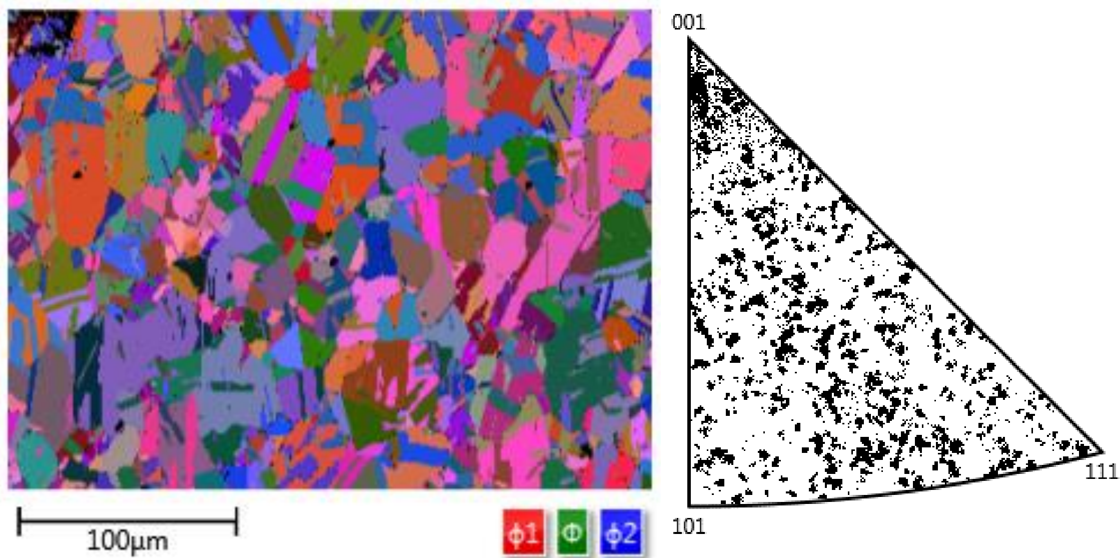


Figure 3.2: SEM-EBSD crystallographic Euler orientation map and corresponding Z-fold (surface normal) inverse pole figure for electropolished copper substrate

The results illustrated in Figure 3.2 show that the copper substrates used in this study possess a slight preferred texture towards the 001 orientation with respect to the z-axis, but with the majority of grains depicting a random distribution of orientation.

Additionally, prior to any electrochemical procedures to reduce contamination and increase deposition uniformity all counter and working electrodes were degreased in acetone (99+% pure, SpeciFied) and deoxidised in 20 vol.% S.G. 1.83 sulphuric acid (Fisher Scientific) followed by rinsing in deionised water.

This technique of electropolishing rolled copper substrates for 4 mins produced a highly polished surface finish in a relatively short time period. These investigations then allowed for two substrates to be used in the later experiments; both electropolished and rolled substrate illustrated minimal and sizeable surface roughness respectively. Therefore, allowing improved EBSD results for thin coatings utilising the electropolished substrates and a manipulation of the IMC distribution and morphology by varying surface roughness.

3.2 Electrodeposition from an aqueous electrolyte

The electrodeposition of tin was undertaken from a bright acid sulphate proprietary bath containing 0.28 M $SnSO_4$ (97% purity, Acros Organics), 1.29 M H_2SO_4 (S.G. 1.83, 95% purity, Fisher Scientific) and 40 ml/L of a proprietary additive; Tinmac (MacDermid). A three electrode cell was used with the aforementioned rolled copper substrates, 0.25 mm thick 4 cm^2 (2 x 2 cm) tin sheet (99.95 % purity, Advent Research Materials) and saturated calomel electrode (SCE), being the working, counter and reference electrodes respectively. Additionally, electrodeposition was achieved using a potentiostat (Solartron ECI 1286) to control and record electrode potentials.

3.2.1 *Effect of deposition current density on deposit structure*

The tin coatings were electrodeposited galvanostatically with and without slow agitation, by magnetic stirring, at current densities of 5, 10, 20, 30, 40 and 50 mA/cm^2 onto as-received (not electropolished) copper substrates. The tin coatings were deposited to a thickness of 10 µm and characterised using SEM and EBSD, to determine surface topography, crystallographic orientation and grain size.

3.3 Electrodeposition from non-aqueous deep eutectic solvents

Tin electrodeposition was carried out from three DESs, commonly referred to as; Reline, Ethaline and Propeline. The compositions of these three solvents are described in Table 3.1.

Table 3.1: Chemical constituents, linear formulae, purities, sources of procurement and concentrations for Reline, Ethaline and Propeline.

Common name	Chemical constituents (name, linear formula, purity, source) (ChCl)	(X)	Molar ratio (ChCl:X)
Reline	Choline Chloride, $(CH_3)_3N(Cl)CH_2CH_2OH$, 99%, Acros Organics	Urea, NH_2CONH_2, 99+%, FisherScientific	1:2
Ethaline		Ethylene glycol, $HOCH_2CH_2OH$, 99.8%, Sigma-Aldrich	1:2
Propeline		Propylene glycol, $CH_3CH(OH)CH_2OH$, 99.5%, Sigma-Aldrich	1:2

Tin, in the form of tin (II) chloride dihydrate ($SnCl_2.2H_2O$, 96+% purity, Fisher Scientific), was then added to all three solvents; Reline, Propeline and Ethaline, at varying concentrations until fully dissolved, taking up to 30 min. These three liquids were then used for various electrochemical experiments.

3.3.1 *Non-aqueous electrolyte DC voltammetry for tin electrodeposition*

Voltammetry of the three DESs was undertaken using a three electrode cell, using; platinum gauze for the working and counter electrodes and a silver/silver chloride quasi-reference electrode.

The DESs (Reline, Ethaline and Propeline) underwent voltammetry with additions of $SnCl_2$ at concentrations of 20 g/L, equivalent to 0.088 M. For each bath formulation, cyclic voltammetry was undertaken with a potentiostat (Solartron ECI 1286) at a rate of 50 mV/s for eight cycles. This was undertaken at 70°C to reduce viscosity and improve ion diffusion rates, between potentials vs. Ag/AgCl of -3 V and +3.5 V, -2.5 V and +4 V, -3 V and +3.5 V, respectively, for Reline, Ethaline and Propeline.

Additionally, Reline, Ethaline and Propeline were analysed with concentrations of 0, 10, 80 and 160 g/L $SnCl_2$ (0 M, 0.044 M, 0.35 M, and 0.71 M, respectively). Cyclic voltammetry was undertaken with a potentiostat (Solartron ECI 1286) at a rate of 50 mV/s for five cycles for each electrolyte. These electrolytes were analysed at 70°C between potentials vs. Ag/AgCl of -1.75 V and +1.35 V, -1 V and +1.8 V, -1 V and +1.5 V, respectively.

Moreover, potentiostatic polarisation of Propeline was undertaken at 1.5 V vs. Ag/AgCl for 60 mins followed by additions of 0.088 M $SnCl_2$. This was undertaken to determine if any oxidation or reduction products caused electrochemical affects. This arrangement utilised a divided 'H' cell (3M), keeping cathodic and anodic products separated, with platinum gauze electrodes for the working and counted electrodes and Ag/AgCl as the reference electrode. This electrolyte was then compared

to an identical one which had not undergone potentiostatic polarisation. This comparison was undertaken by cyclic polarisation being carried out on both formulations between -3 V and +3.5 V vs. Ag/AgCl at 50 mV/s for 10 cycles.

3.3.2 *Optimisation of electrodeposition parameters*

A two electrode cell was used with 4 cm x 2 cm as-received (un-electropolished) copper substrates and 0.25 mm thick 4 cm^2 tin sheet (99.95 % purity, Advent Research Materials) used as the working and counter electrodes respectively.

Electrodeposition was undertaken from 100 ml of electrolyte at 70°C to encourage mass transport while using concentrations of $SnCl_2$ of; 10, 80 and 160 g/L (equivalent to; 0.044 M, 0.35 M and 0.71 M, respectively). Electrodeposition was carried out using a potentiostat (Solartron ECI 1286), applying galvanostatic current densities of 0.5, 1, 5, 10 and 20 mA/cm^2 for each electrolyte composition. Agitation at 60 RPM was employed during electrodeposition using a 2.5 cm magnetic stirrer. Moreover, each deposition passed 12 C of charge with predicted uniform thickness of approximately 2.5 µm, assuming a 100% cathode current efficiency. A flow diagram describing the processes is shown in Figure 3.3.

To measure the thickness of the resulting electrodeposits, assuming uniform electrodeposition, and to calculate the cathode current efficiency, weight gain was recorded using a microbalance (Mettler Toledo XS 205 dual range) measuring mass before and after electrodeposition.

The electrodeposits were then evaluated individually using the criteria of cathode current efficiency and electrodeposit uniformity. The variables that produced the most uniform tin coatings were then studied further by attempting to increase the coating thickness using increased electrodeposition times of 1, 2, 5, 10, 15, 20, 25, 30, and 35 min. When the uniformity of the electrodeposit was seen to be unacceptable, featuring clear dendritic or needle-like growth, no further deposition times were undertaken, thus defining the limiting thickness of the coating. This took place for three electrodeposits; two from Propeline with $SnCl_2$ concentrations of 0.044 M at 20 mA/cm^2 and at 0.71 M at 0.5 mA/cm^2 and one from Ethaline with 0.044 M at 10 mA/cm^2. The description of this process is seen in the flow diagram in Figure 3.4.

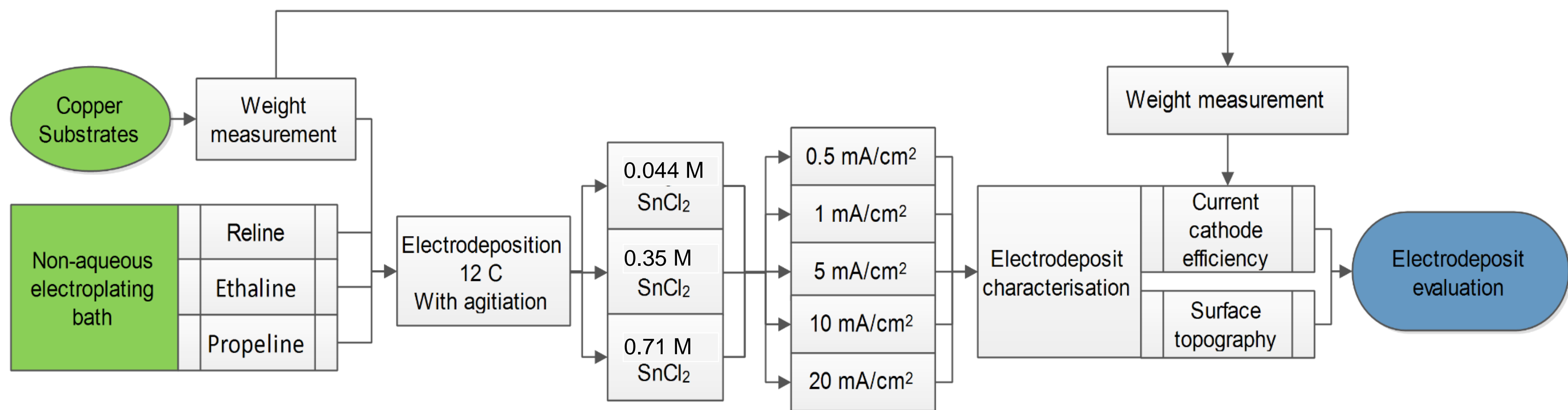


Figure 3.3: Flow diagram for the processes undertaken for electrodeposition from three deep eutectic solvents, three $SnCl_2$ concentrations and five current densities

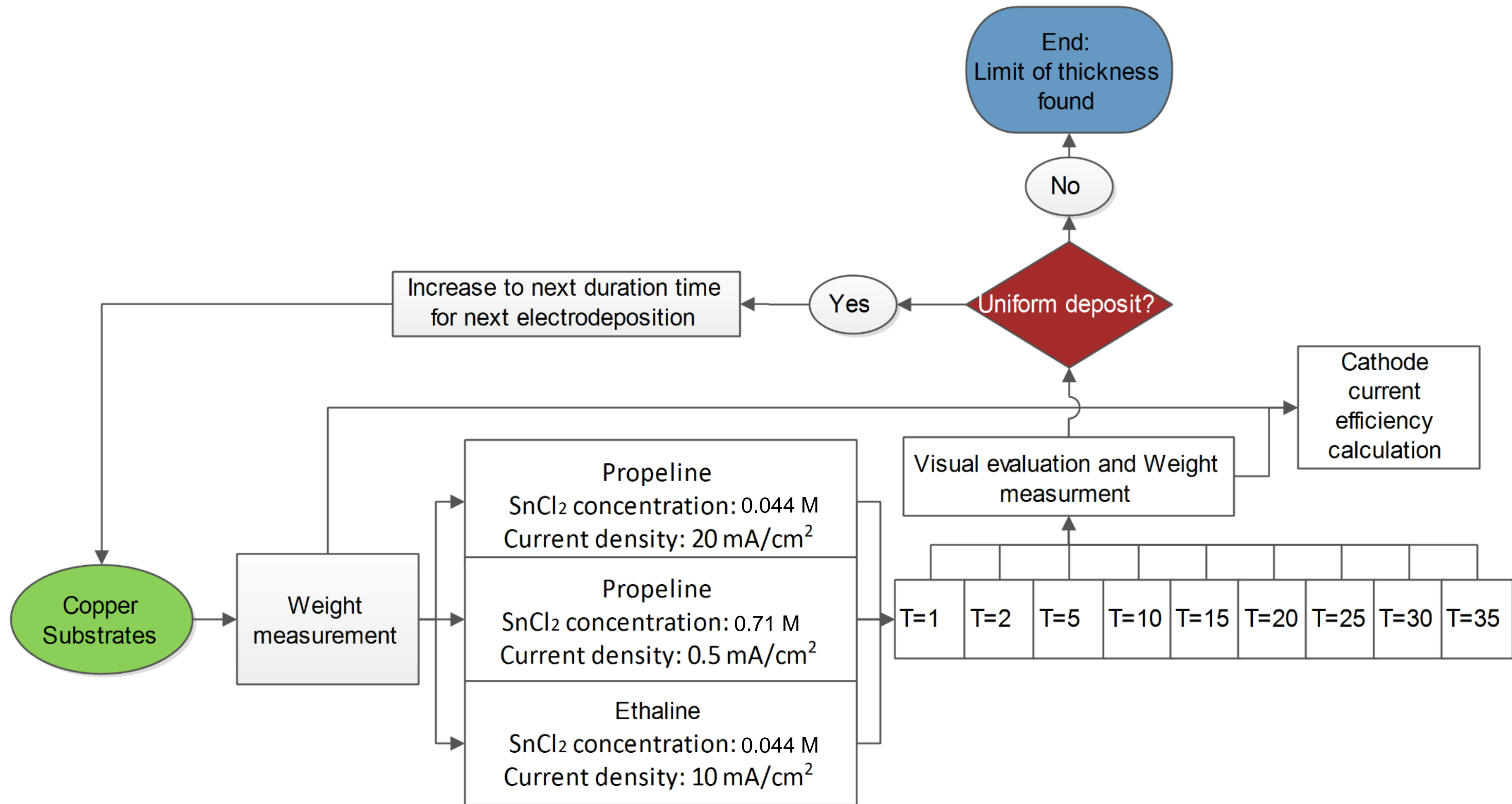


Figure 3.4: Flow diagram of the processes undertaken for electrodeposition from three specific non-aqueous tin electroplating baths to increase deposition thickness. Where 'T' is the time period of deposition in minutes

3.4 Comparison of electrodeposits from aqueous and non-aqueous electrolytes

The electrodeposits that were selected for further study from the process of electrodeposition parameter refinement were then undertaken from Propeline at $SnCl_2$ concentrations and current densities of 0.044 M at 10 mA/cm^2 onto electropolished copper substrates to thicknesses of 1 and 2 µm for both formulations. These electrodeposits, as illustrated in Figure 3.5, were then used to evaluate: current cathode efficiency, surface topography (SEM), crystallographic texture (EBSD), IMC growth after 7 and 177 days storage (electrochemical anodic oxidation), absorbed electrolyte contamination (XPS sputter profile), cross-section (FIB/mechanical) and whisker propensity (SEM). Mechanical cross-sectioning was carried out by mounting the sample in non-conductive cold mounting resin (EpoFix, Struers), grinding with silicon carbide papers (240 grit through to 1200 grit), polishing with 6 µm and 3 µm diamond suspensions and finally, polishing with pH 9.8, 0.04 µm colloidal silica.

3.5 Electrochemical anodic oxidation of tin

The study of IMC growth rate, morphology and distribution has been shown to be of significant interest for whisker growth and subsequent component reliability. Other chemical methods for removing the tin coatings formed from the aqueous electroplating bath were unsuccessful [254]. Therefore, an alternative method was developed and utilised.

3.5.1 *Procedure for technique development*

As illustrated in Figure 3.6, removal of 5 and 10 µm thick electrodeposited tin layers were undertaken via an electrolytic process. Two electrolyte solutions; S.G. 1.83 H_2SO_4 and S.G. 1.16 HCl, both at 5 vol.-% (0.9 M and 0.5 M, respectively), and 1.2 % (0.2 M and 0.1 M, respectively), were examined along with two methods of inducing the anodic oxidation reaction i.e. potentiostatically and galvanostatically, both utilising a potentiostat (Solartron ECI 1286).

Voltammetry for both 0.25 mm thick 4 cm^2 tin sheet, and 100 µm thick 4 cm^2 copper was also undertaken by varying the potential from -1.0 V to +1.0 V (vs. SCE) for tin and -0.5 V to 0.5 V (vs. SCE) for copper in both electrolytes at a rate of 50 mV/s.

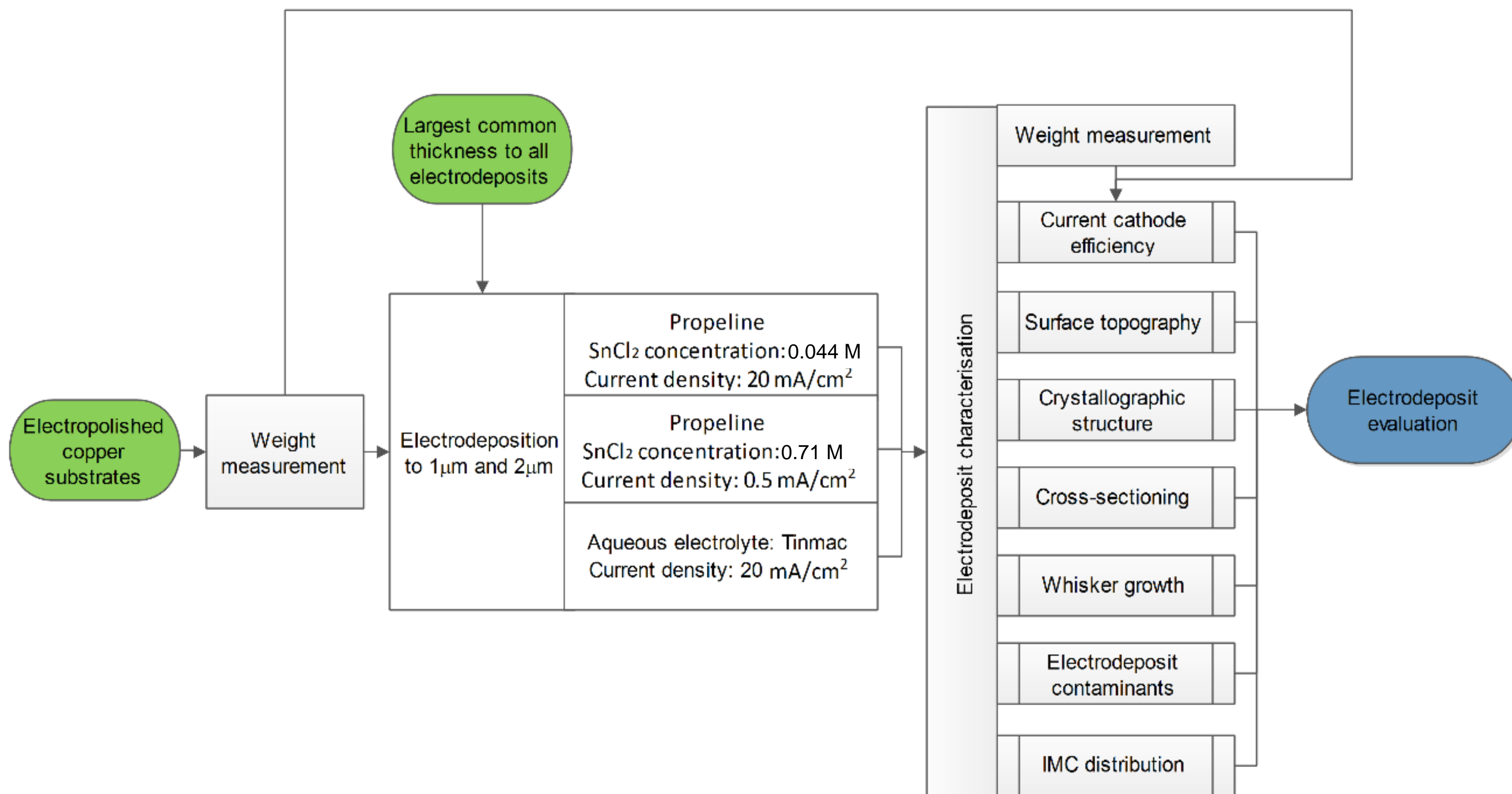


Figure 3.5: Flow diagram illustrating the process of creating electrodeposits from Propeline and Tinmac at equivalent thicknesses for characterisation and evaluation

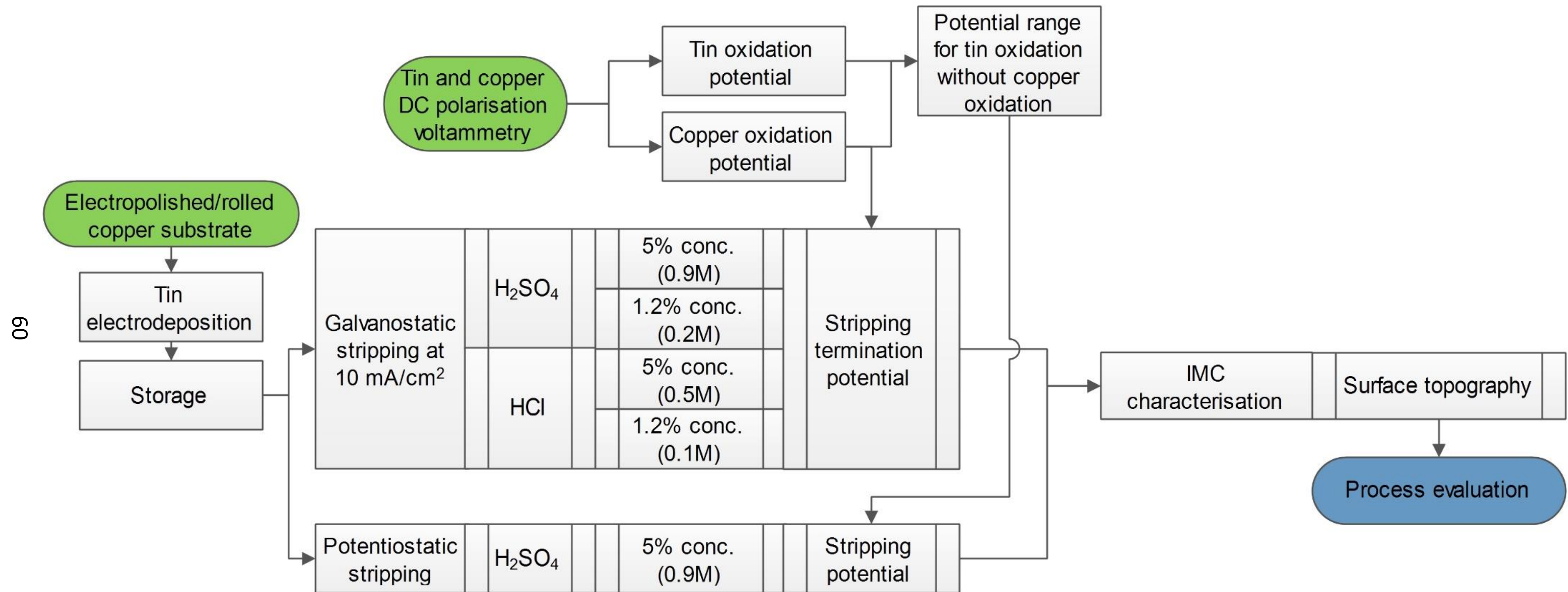


Figure 3.6: Flow diagram representing the processes undertaken to develop the electrochemical anodic oxidation of tin for tin stripping

For galvanostatic stripping, tin electroplated samples were utilised as the working electrode in a three electrode cell with copper sheet and a saturated calomel electrode (SCE) used as the counter and reference electrodes, respectively. The electroplated samples were oxidised at 10 mA/cm^2 in both electrolytes. The oxidation reactions were deemed complete when the working electrode potentials exceeded the oxidation potential for tin in the respective electrolyte solutions.

For the potentiostatic technique, tin stripping was performed in 5 % H_2SO_4 (S.G. 1.83) at a potential of -0.26 V (vs. SCE) using the same experimental arrangement as outlined for the galvanostatic stripping. The samples were anodically polarised until the current reduced to approximately zero indicating the complete oxidation of tin.

3.5.2 *Electrochemical anodic oxidation procedure of tin for IMC study*

In the early stages of using these techniques for IMC analysis potentiostatic tin stripping was undertaken for samples that had been electrodeposited from the aqueous tin bath at 20 mA/cm^2 to 5 µm thick, as seen in Figure 3.7-'Original technique'.

These electrodeposits were formed onto a range of substrates; electropolished, mechanically polished and rolled. These samples were stripped after a storage period at room temperature of 1 day, 17 days, 55 days and 445 days, or 7 days with heat treatment at 150°C for 30 mins. Additionally, mechanical polishing of substrates was performed by mounting copper substrates in non-conductive cold mounting resin (EpoFix, Struers), grinding with sequential silicon carbide papers (240 grit through to 1200 grit), polishing with 6 µm and 3 µm diamond suspensions and finally polishing with pH 9.8, 0.04 µm colloidal silica to verify that the electropolishing process was not affecting IMC growth.

In the later stages of this project, both galvanostatic and potentiostatic techniques were used sequentially to characterise the IMC growth of the electrodeposit formed from the DES and aqueous proprietary tin bath, as seen in Figure 3.7-'Sequental technique;. Firstly samples were oxidised at 10 mA/cm^2 in 5 % H_2SO_4 (S.G. 1.83) until the potential rose rapidly exceeding 0 V (vs. SCE), followed by potentiostatic oxidation at a potential of -0.26 V (vs. SCE), until the current dropped to approximately zero.

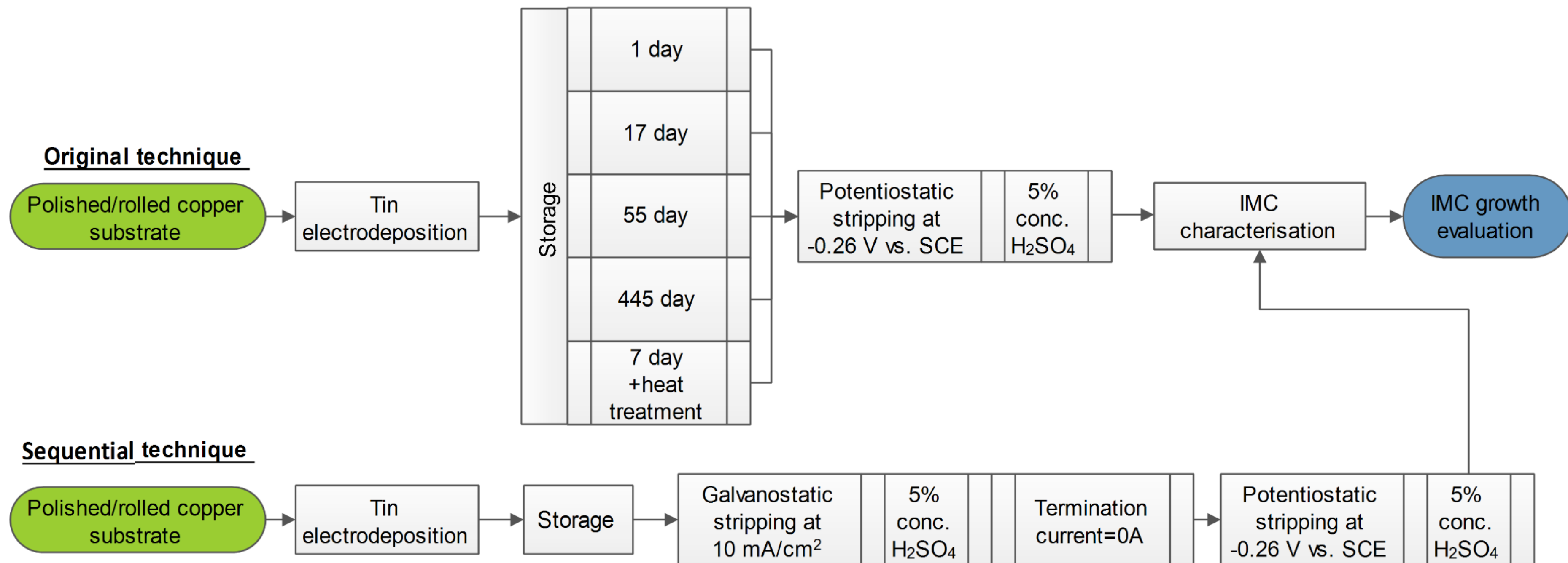


Figure 3.7: Flow diagram of original technique used in the early stages of the project and sequential technique used in later stages for the anodic oxidation of tin from electrodeposited tin on copper substrates

3.6 Surface and subsurface characterisation techniques

3.6.1 *Scanning electron microscopy*

Scanning electron microscopy (SEM), energy-dispersive X-ray spectroscopy (EDS) and EBSD were performed using a high resolution Carl Zeiss Leo 1530 VP FEG SEM equipped with an Oxford Instruments X–Max 80 mm^2 EDS detector and a HKLNordlys F high speed camera. SEM was used to analyse the finish of each sample, such as, electrodeposit surface topography, IMC growth distribution and whisker formation. This technique utilised two secondary electron detectors; one situated in the chamber with the specimen, and the other situated in the pole piece called the in-lens detector. The in-lens detector was able to achieve increased brightness and contrast at small working distances and high magnifications, and accentuated contrast within images compared to the conventional secondary electron detector. The SEM was operated at an accelerating voltage of 15 kV or 20 kV.

EBSD allowed for crystallographic information to be collected for both the tin electrodeposit and the copper substrate. The specimens, being placed in a 70 ^{o} sample holder, required a low level of surface roughness and contamination for effective analysis. This technique analysed each point and, by Kikuchi pattern matching, defined the crystallographic orientation of that location. If the pattern was unidentified, which is caused by surface roughness, internal stress or surface contamination, the analysis point remains unrecorded. Moreover, Kikuchi patterns can be indexed wrongly, which is highlighted in a large mean angular deviation (MAD) value. Due to these limitations of mis-indexing, EBSD data was refined and analysed using software; HKL CHANNEL5 (Oxford Instruments). In addition, HKL CHANNEL5 was used to calculate grain size enabling the grain size distribution to be evaluated. The software was also used to generate folded pole figure and folded pole figure density maps.

The distributions of grain sizes were created from an average of two maps from each sample; at 10 kX and 5 kX magnification. The data produced were then refined in HKL CHANNEL5. This distribution was then interpreted by creating a histogram of grain sizes. In addition, many grains measured were seen to have an area of less than 4 pixels, and were deemed erroneous high error results. Therefore, grains seen to have an area of less than 4 pixels were removed from the distribution.

3.6.2 *X-ray photoelectron spectroscopy depth profiling*

X-ray photoelectron spectroscopy depth profiling (XPS) was performed using a monochromatic small-spot K-Alpha X-ray photoelectron spectrometer with an aluminium source (Thermo Scientific). XPS was carried out initially on the surface of the electrodeposit, followed by sputtering, and spectroscopy of the subsurface. XPS analysis was carried out using a spot size of 300 µm and pass energy of 151.2 eV. A 100 second sputter using accelerated argon ions at 4 KeV was used to remove surface contaminants. Both survey scans and high resolution scans were used to analyse the deposit surface. The survey scans were undertaken using a step size of 1 eV from 0 to 1341 eV and the high resolution scans utilised a step size of 0.1 eV, over element specific energy ranges at a pass energy of 50 eV.

3.6.3 *Dual beam microscopy*

A Nova 600 Nanolab Dual Beam microscope combining a focused ion beam and FEG-SEM (FEI) was used for cross-sectional thickness measurements. By depositing platinum over a selective area, sufficient protection was given so that ion milling could take place without damaging the region of interest. The digging of trenches by ion milling allowed for electron and ion beam images to be captured of the cross-sections of electrodeposits. This technique allowed for small thicknesses of tin electrodeposit to be cross-sectioned with minimal edge effects commonly seen during mechanical cross-sectioning.

3.6.4 *Whisker measurement techniques*

Whisker measurement was undertaken using a SEM from which whisker densities were calculated. This was undertaken by counting visible whiskers while sweeping across the sample at 1000 times magnification. These sweeps across the surface of the samples analysed an area of 4 mm^2 (200 µm by 2 cm) for each sweep. Ten sweeps of such sized areas were undertaken for each sample in random locations, culminating in a total analysed area of 40 mm^2. This therefore allowed for a whisker density to be calculated.

3.7 Experimental summary

In conclusion, several techniques have been developed to facilitate the creation and analysis of tin electrodeposits created from both aqueous and non-aqueous electroplating baths. Due to the novelty of the non-aqueous electrolytes, process variable refinement had to be undertaken to create the thickest and most uniform electrodeposits. These electrodeposits were then characterised by various means evaluating their cross-section, incorporated contaminants, current efficiencies during

electrodeposition, surface topography and crystallographic structure. Furthermore, by developing a substrate electropolishing and tin electrodeposit stripping process, it was possible to analyse the IMC growth independent of substrate roughness effects. In addition, further study utilising electropolished substrates, electrodeposits from the aqueous electroplating bath, and the electrochemical anodic oxidation of tin enabled the study of IMC growth and distribution with respect to time.

Chapter 4 Results and Discussion

4.1 Electrodeposition from an aqueous electrolyte

10 µm thick deposits of pure tin were produced from a bright acid sulphate electroplating bath incorporating Tinmac additives (MacDermid Corporate) at varying current densities, with and without agitation using as-received rolled copper substrates. These parameters were selected due to their frequent intentional or unintentional variation during electrodeposition, e.g. intentional variations such as the electroplater specifying different current densities and levels of agitation, and unintentional variations such as those caused by electrode geometry. Selected photographs of these electrodeposits of specific interest are shown in Figure 4.1.

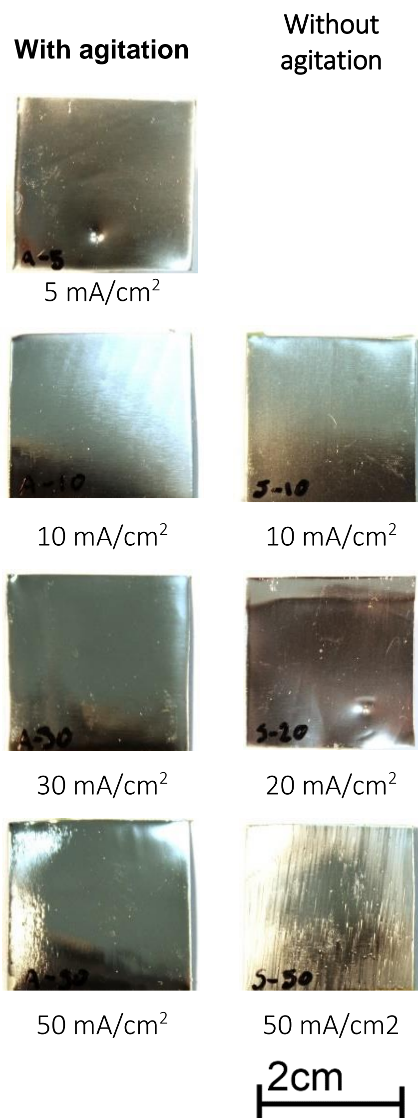


Figure 4.1: Photographs of tin electrodeposits produced from Tinmac with varying current densities, and without agitation

All electrodeposits produced were uniform, exhibited no dendritic growth, and were bright or satin in finish. Deposits produced at 5 mA/cm^2 with agitation, and at 5 and 10 mA/cm^2 without agitation were satin in finish. This reduction in brightness at low current densities could be attributed to inadequate additive interactions, or encouraged faceted deposit growth due to low reduction over potentials, but due to the complex nature of electrolyte dynamics distinguishing between these possibilities would be difficult.

In addition, electrodeposits produced at current densities above 20 mA/cm^2 without agitation showed an increasing level of vertical streaks across their surface. Whereas, with agitation these streaks were only slightly visible for the electrodeposit produced at 50 mA/cm^2. This streaking effect was attributed to hydrogen evolution on the deposit surface, bubbles of which coalesced and rose vertically leaving this patterning. Samples that experienced agitation at equivalent current densities showed fewer predominant vertical streaks. This was attributed to agitation increasing the removal rate of hydrogen from the surface of

the deposit, reducing the probability for large bubble formation and the affect which they caused while rising up the deposit.

These results agree with the supplier's recommended current densities and agitation for rack electrodeposition using this electrolyte. The specification for this electrolyte suggests current densities of 10 and 45 mA/cm^2 with or without agitation, but with further commendation towards current densities between 15 and 20 mA/cm^2 with slow agitation (3 m/min) [257].

Additionally, the electrodeposits were analysed by EBSD producing crystallographic maps. An overview of these are shown in Figure 4.2. Moreover, inverse folded pole figures for the maps in Figure 4.2 show preferred texture towards the 001 direction with respect to the surface normal (Z). An example pole figure for these electrodeposits is shown in Figure 4.3.

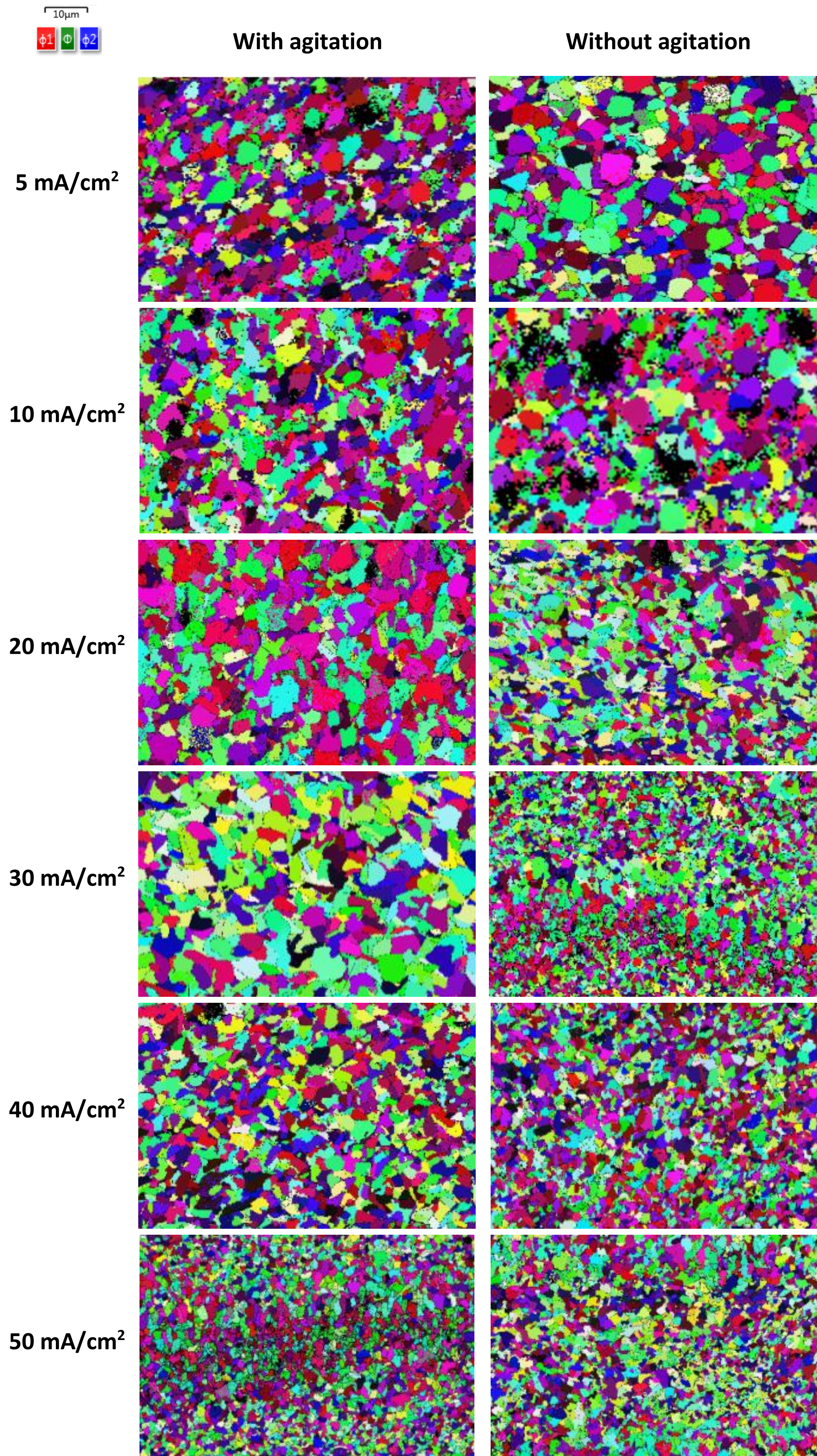


Figure 4.2: EBSD crystallographic Euler maps for 10 μm thick tin electrodeposits produced from a Tinmac proprietary bath at various current densities with and without agitation in plain view

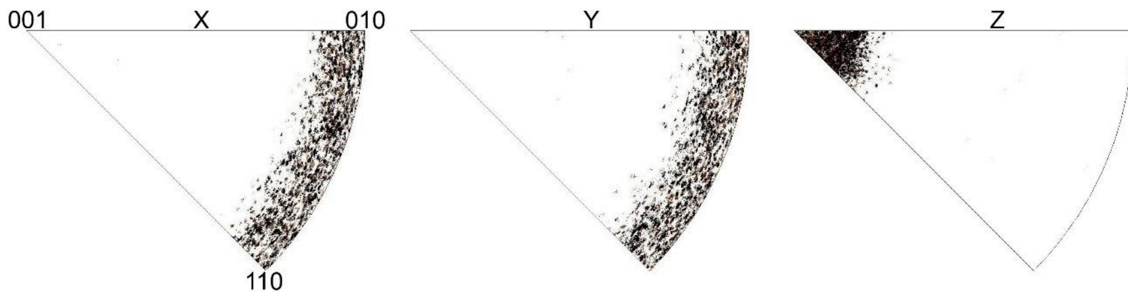


Figure 4.3: Inverse folded pole figure in the X, Y, and Z directions (Z=surface normal) for a tin electrodeposit produced from a Tinmac electrolyte at 5 mA/cm² with agitation

All the EBSD maps and corresponding pole figures collected for these deposits during this project are shown below in Figures 4.4 to 4.32. These Figures are arranged by: agitation, then current density and finally by varying magnifications

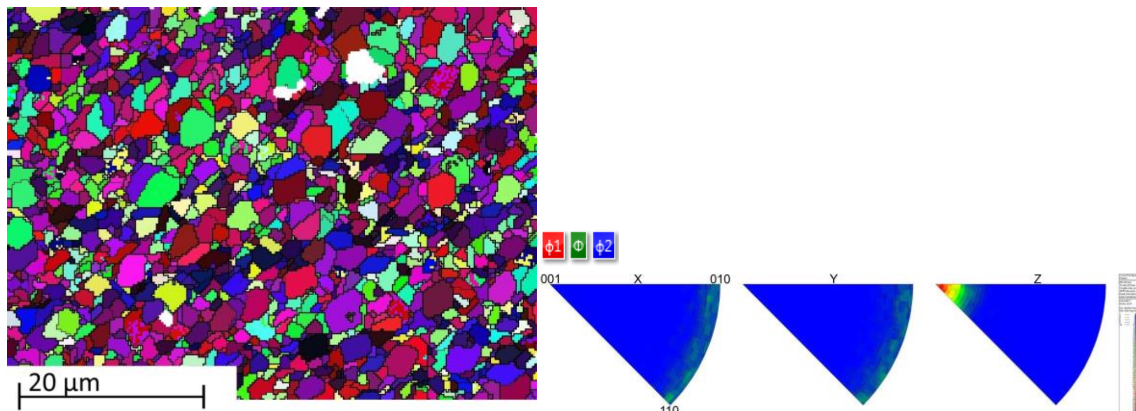


Figure 4.4: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 5 mA/cm²

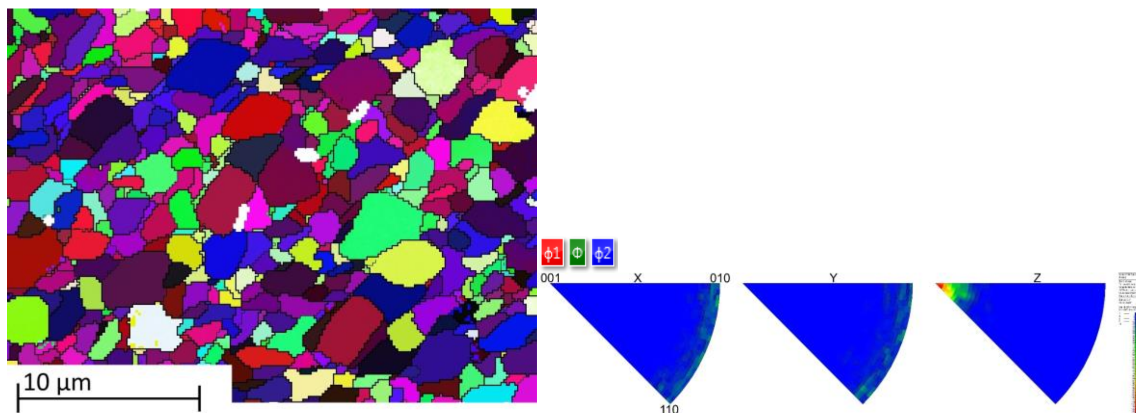


Figure 4.5: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 5 mA/cm²

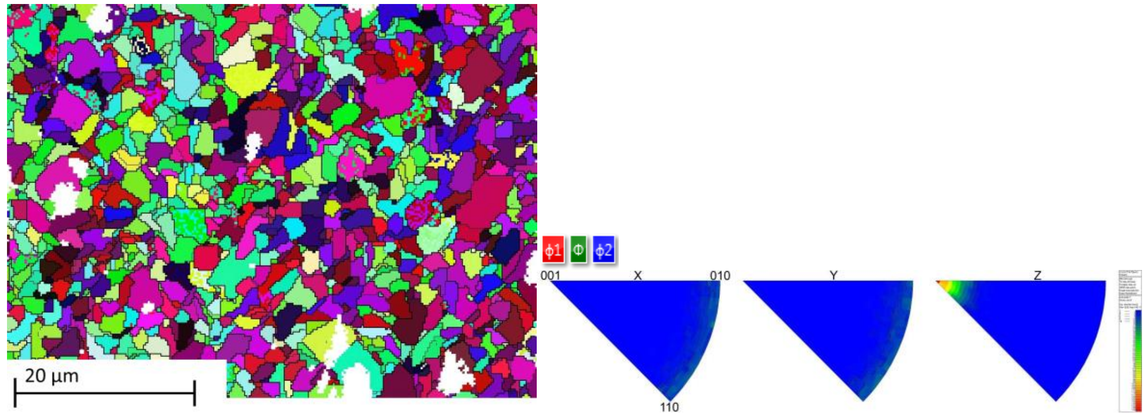


Figure 4.6: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 10 mA/cm²

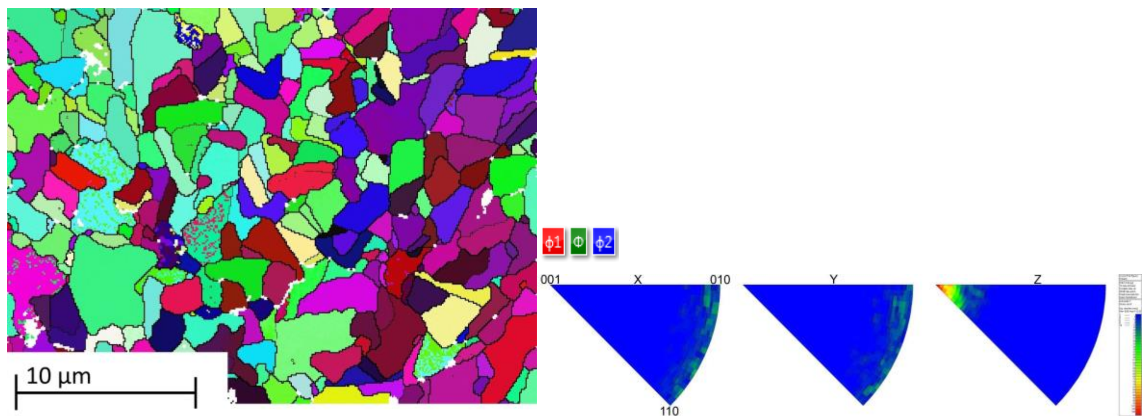


Figure 4.7: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 10 mA/cm²

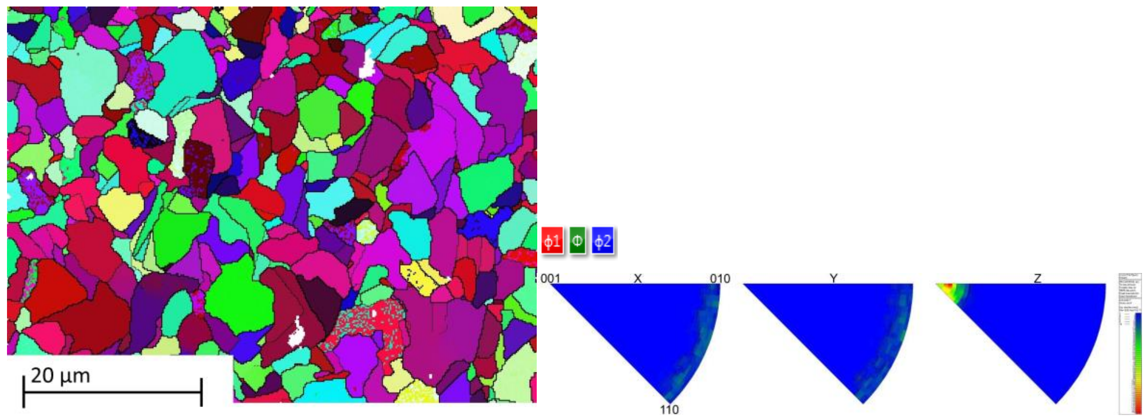


Figure 4.8: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 20 mA/cm²

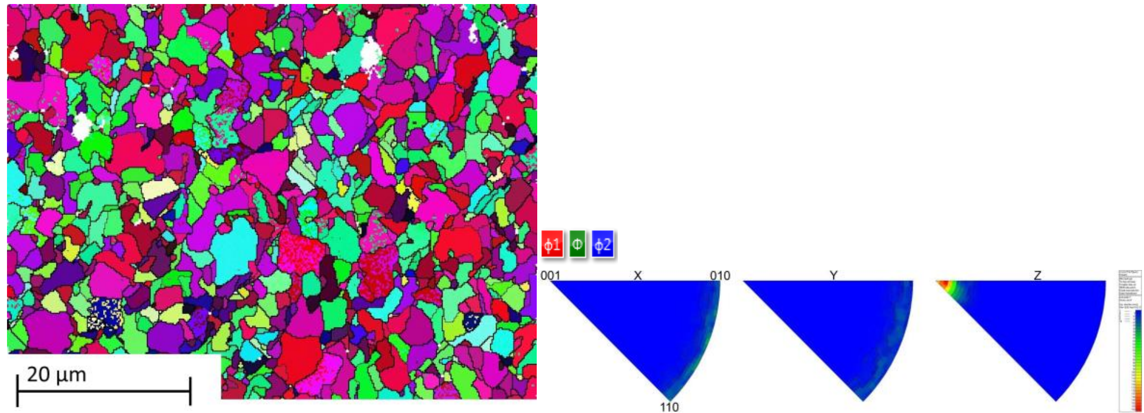


Figure 4.9: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 20 mA/cm^2

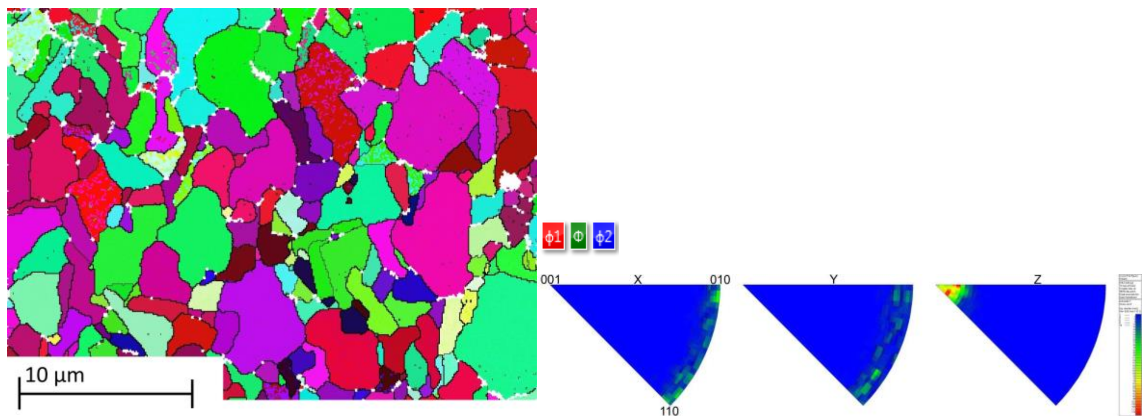


Figure 4.10: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 20 mA/cm^2

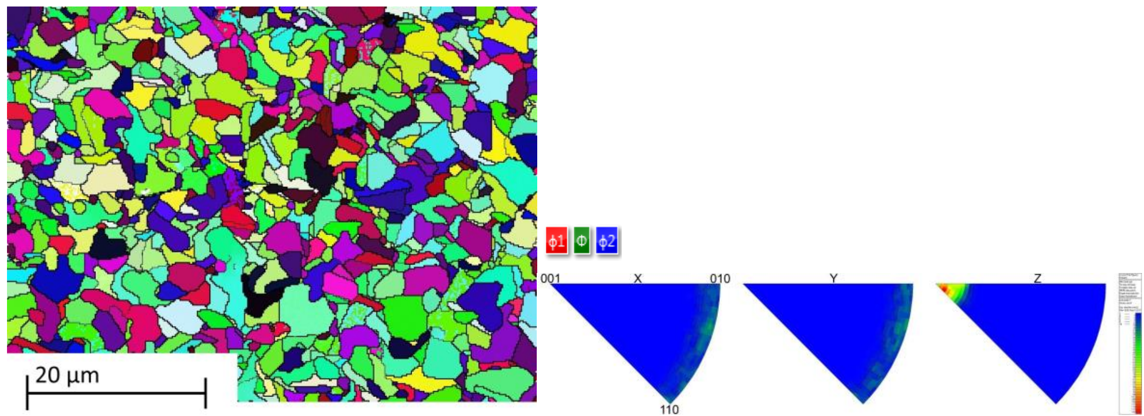


Figure 4.11: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 30 mA/cm^2

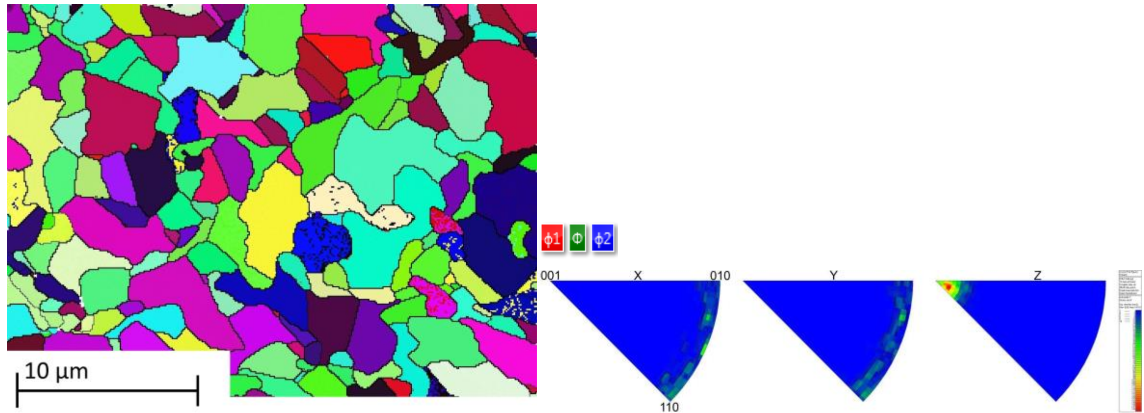


Figure 4.12: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 30 mA/cm^2

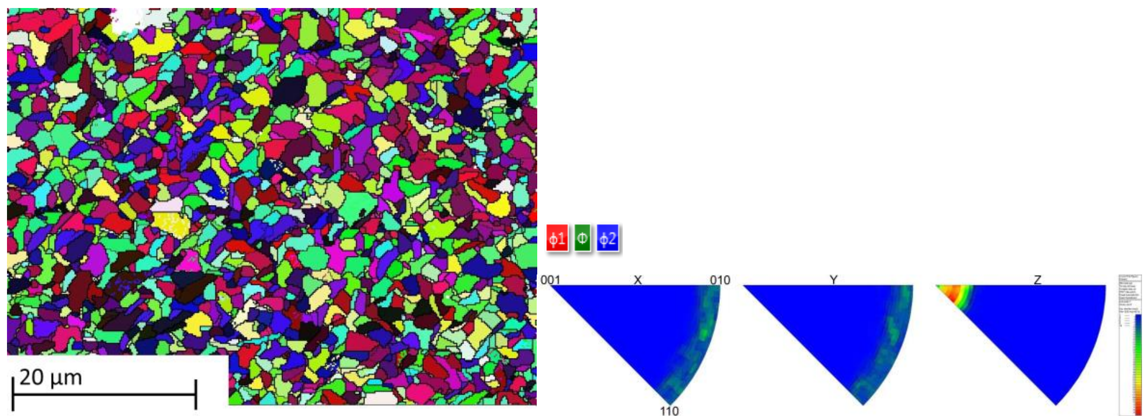


Figure 4.13: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 40 mA/cm^2

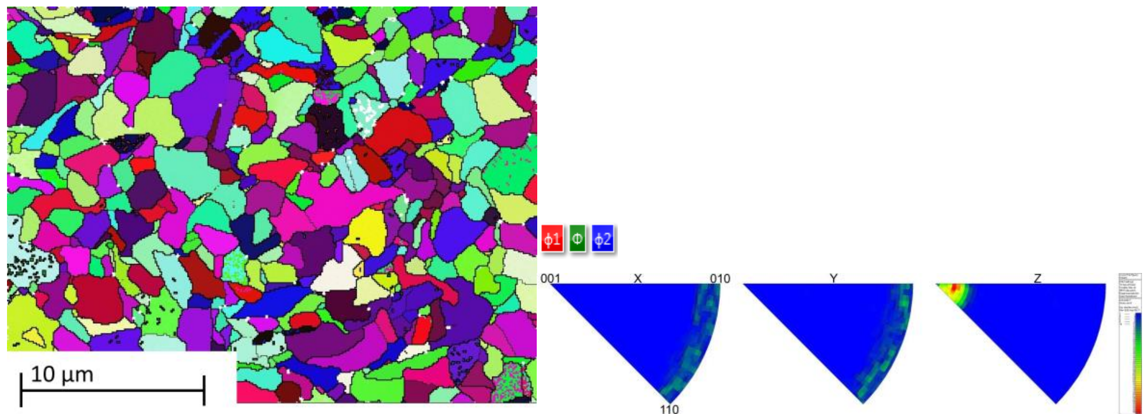


Figure 4.14: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 40 mA/cm^2

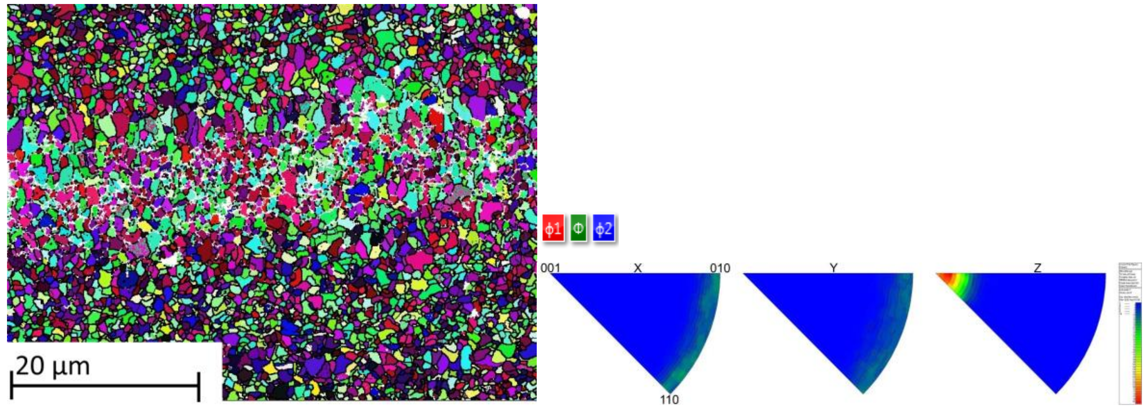


Figure 4.15: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 50 mA/cm^2

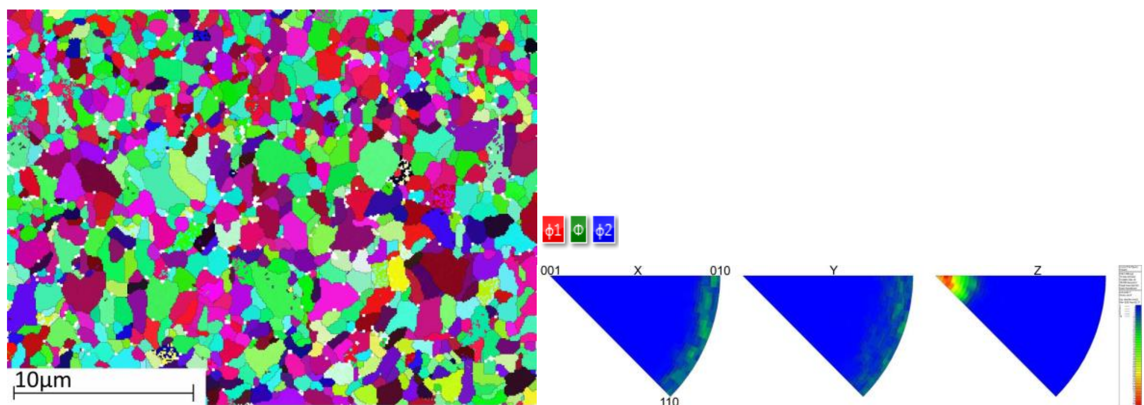


Figure 4.16: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac with agitation at 50 mA/cm^2

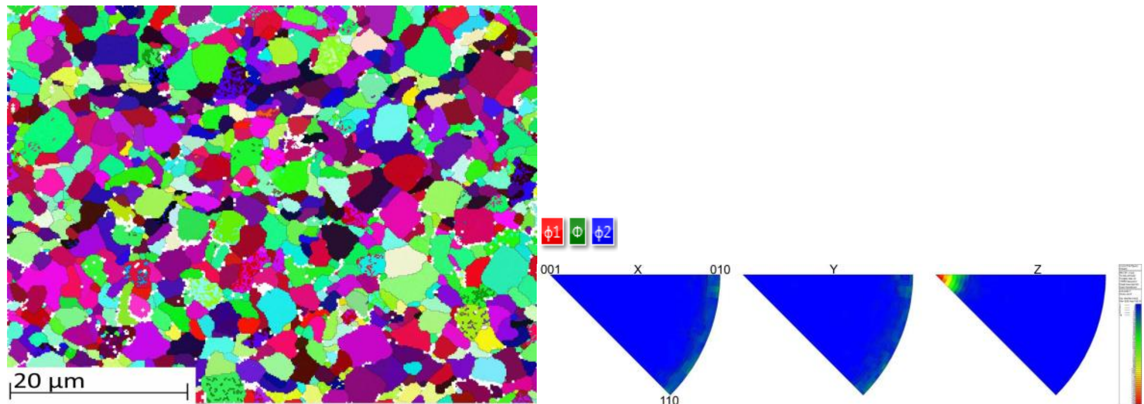


Figure 4.17: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 5 mA/cm^2

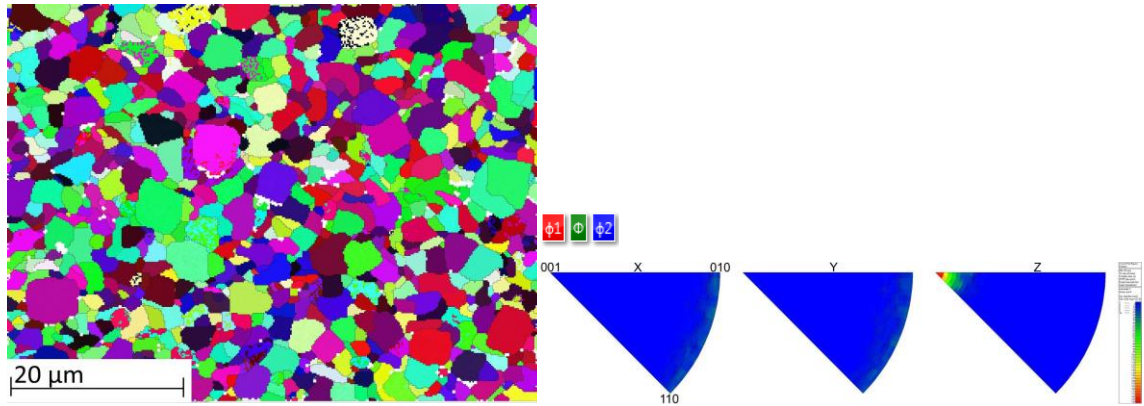


Figure 4.18: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 5 mA/cm^2

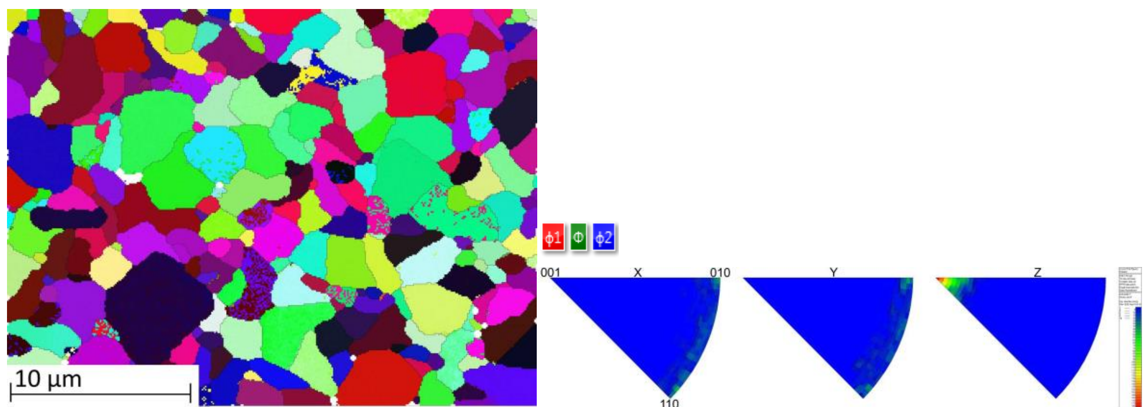


Figure 4.19: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 5 mA/cm^2

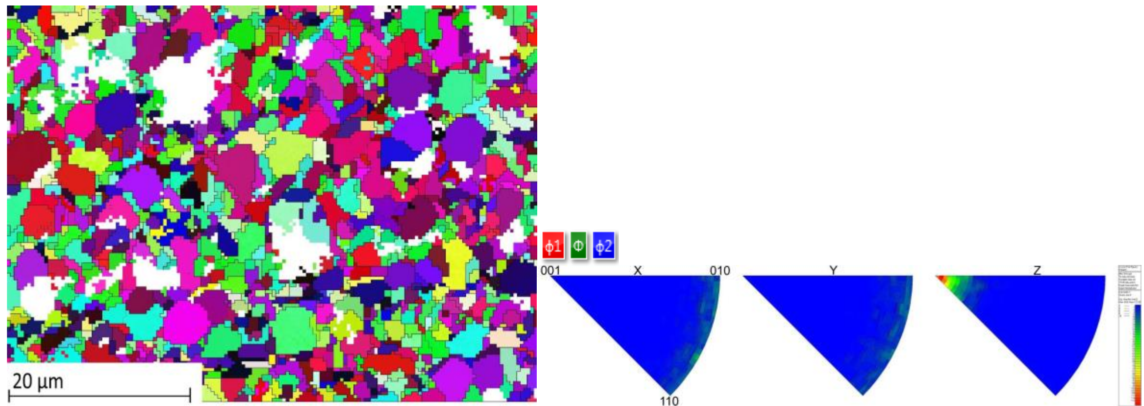


Figure 4.20: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 10 mA/cm^2

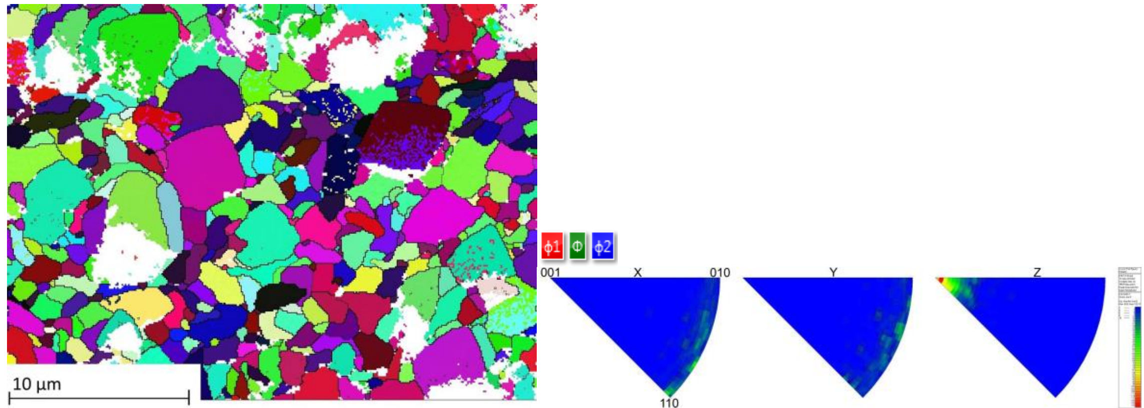


Figure 4.21: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 10 mA/cm²

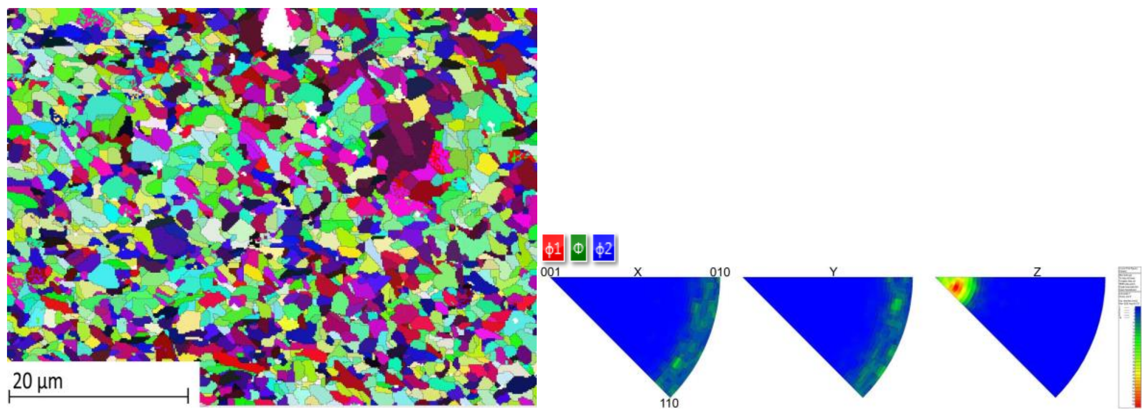


Figure 4.22: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 20 mA/cm²

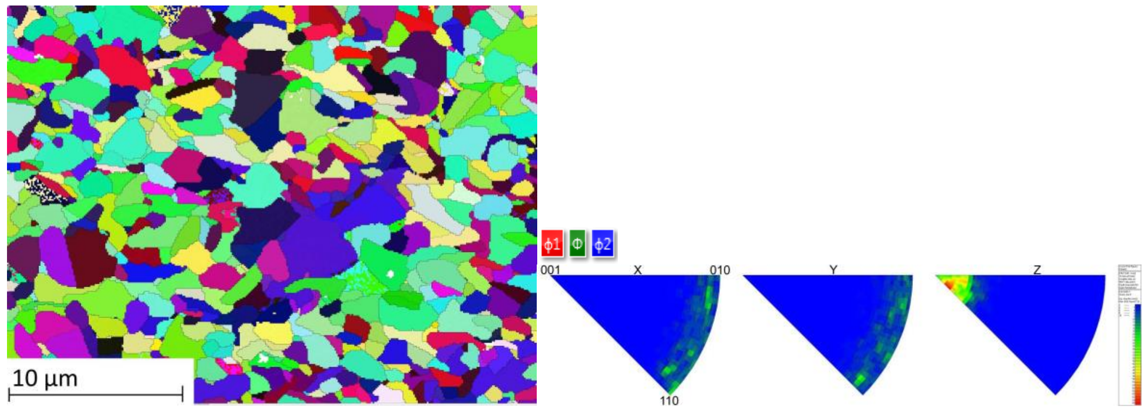


Figure 4.23: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 20 mA/cm²

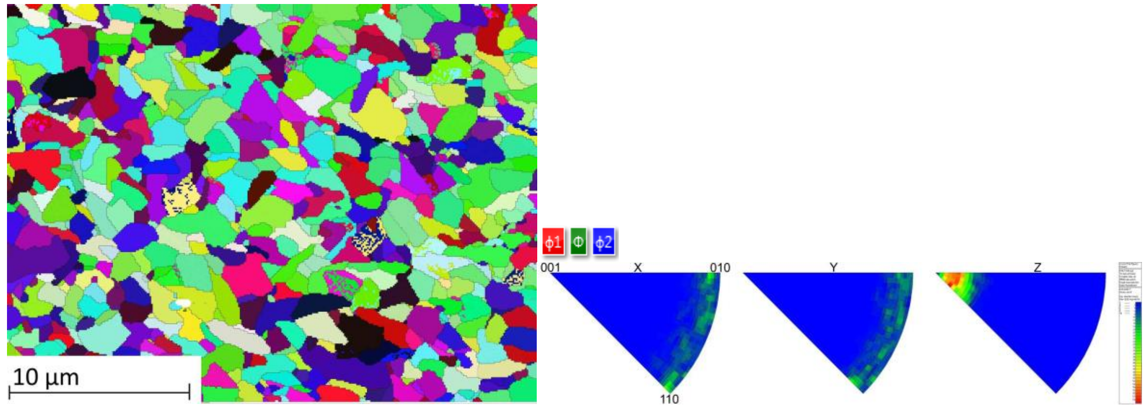


Figure 4.24: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 20 mA/cm²

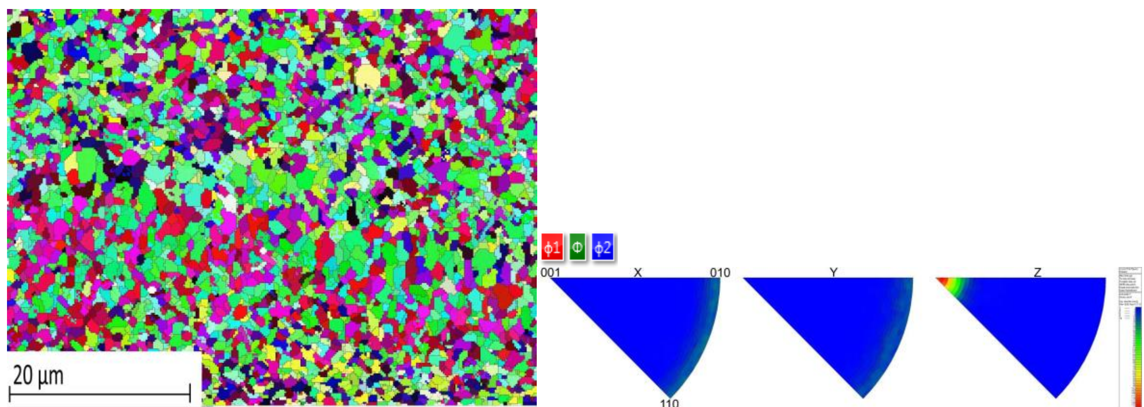


Figure 4.25: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 30 mA/cm²

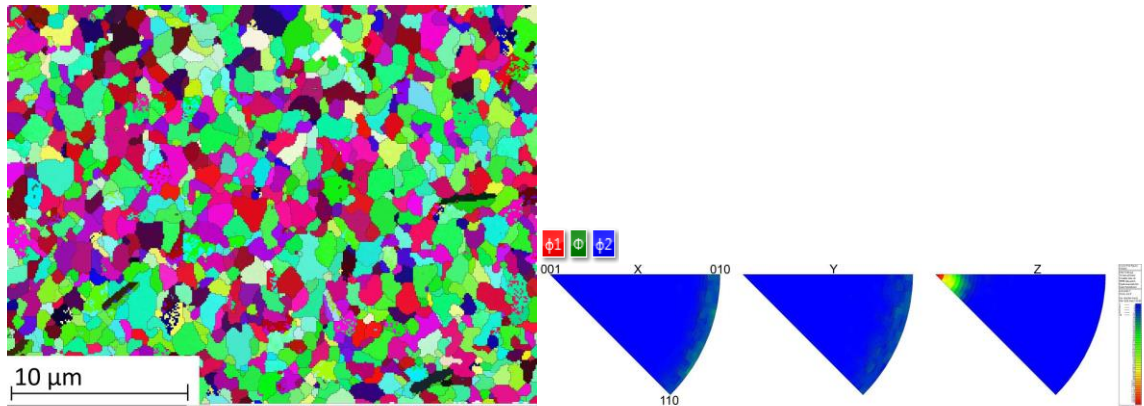


Figure 4.26: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 30 mA/cm²

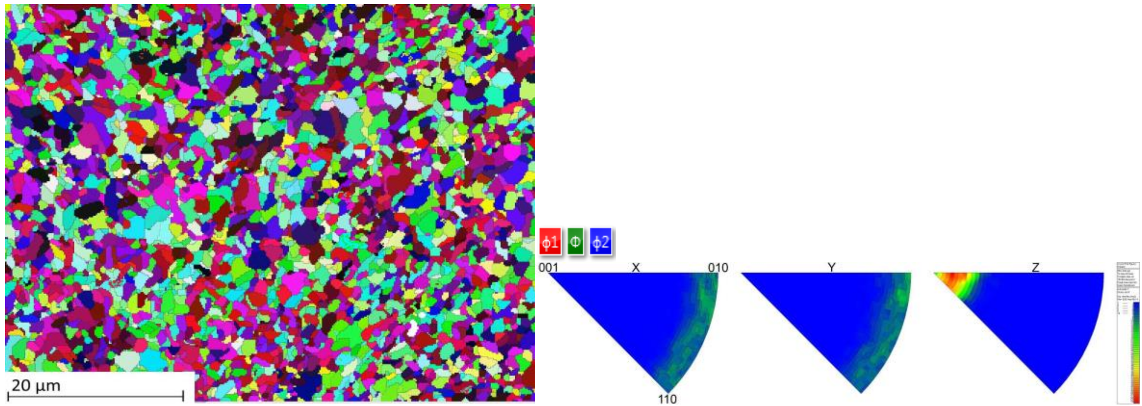


Figure 4.27: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 40 mA/cm^2

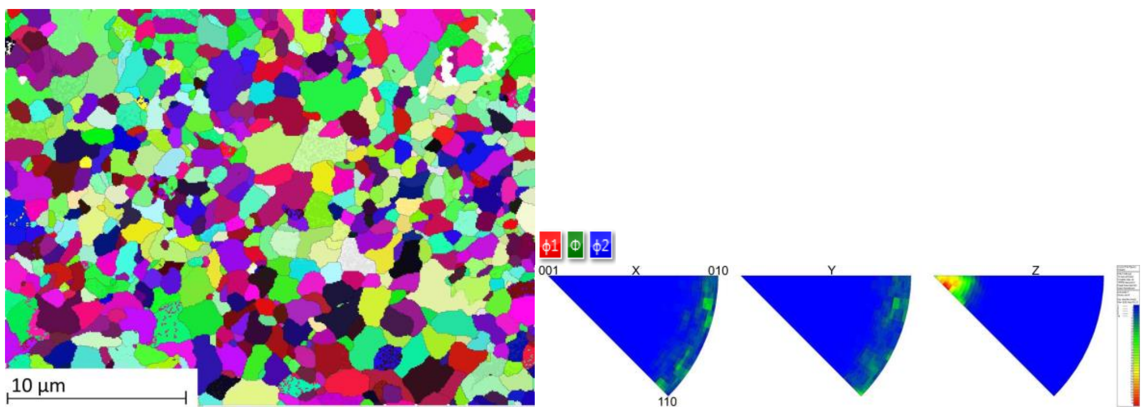


Figure 4.28: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 40 mA/cm^2

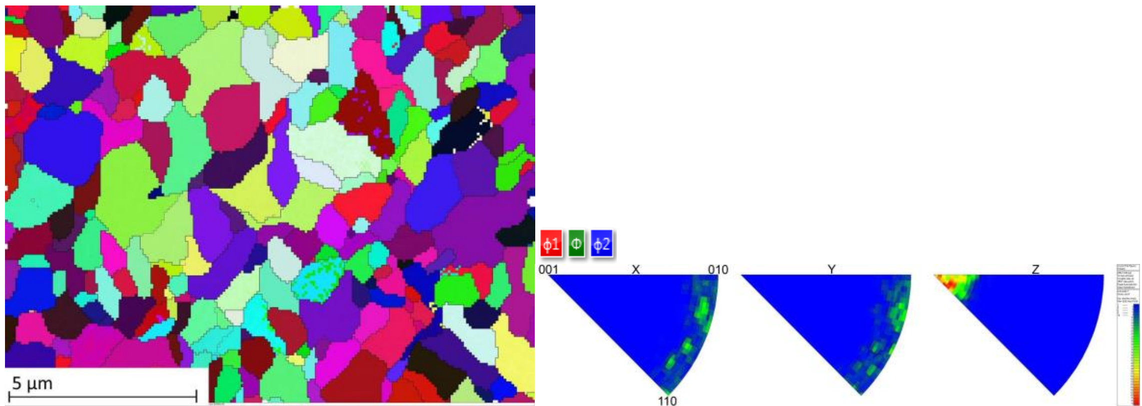


Figure 4.29: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 40 mA/cm^2

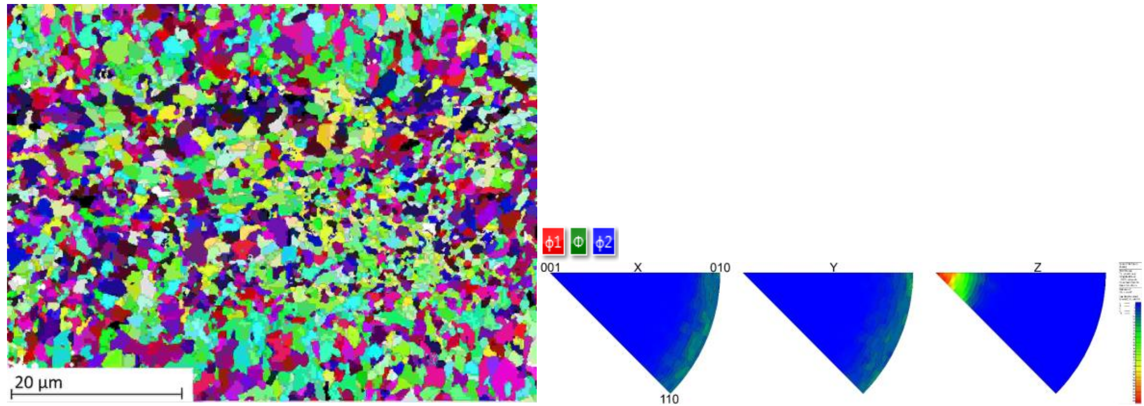


Figure 4.30: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 50 mA/cm^2

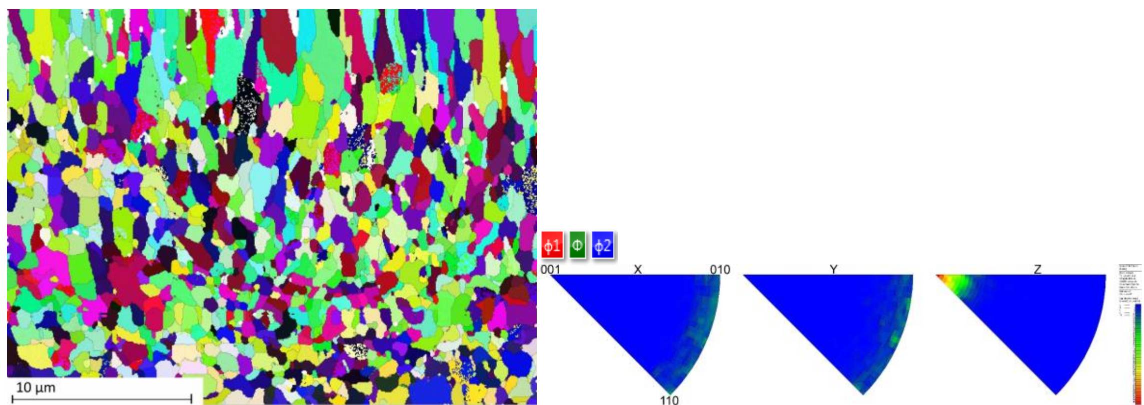


Figure 4.31: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 50 mA/cm^2

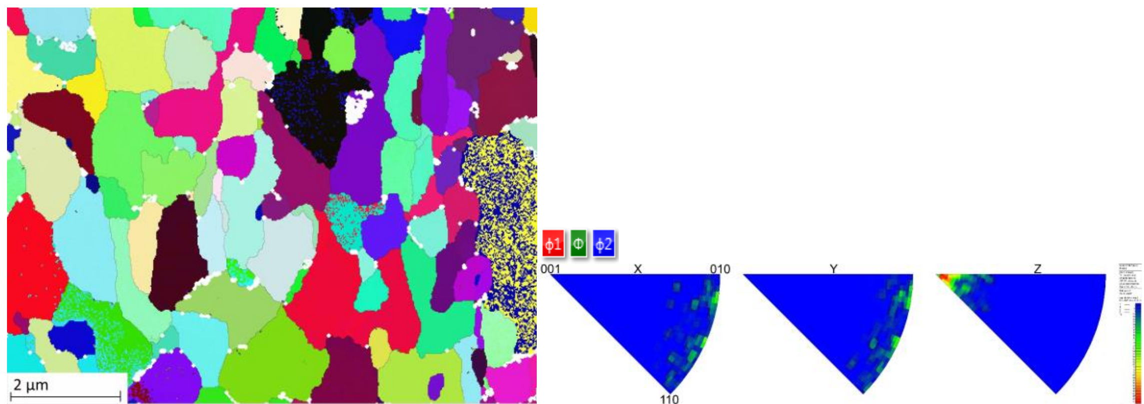


Figure 4.32: EBSD Euler map and folded pole figures for tin electrodeposit produced from Tinmac without agitation at 50 mA/cm^2

Additionally, the EBSD maps in Figures 4.4 to 4.32 allowed the calculation of mean grain size for each electrodeposit as shown in Figure 4.33. Moreover, the standard deviation of the mean grain size for each deposit provided for the error bars in Figure 4.33

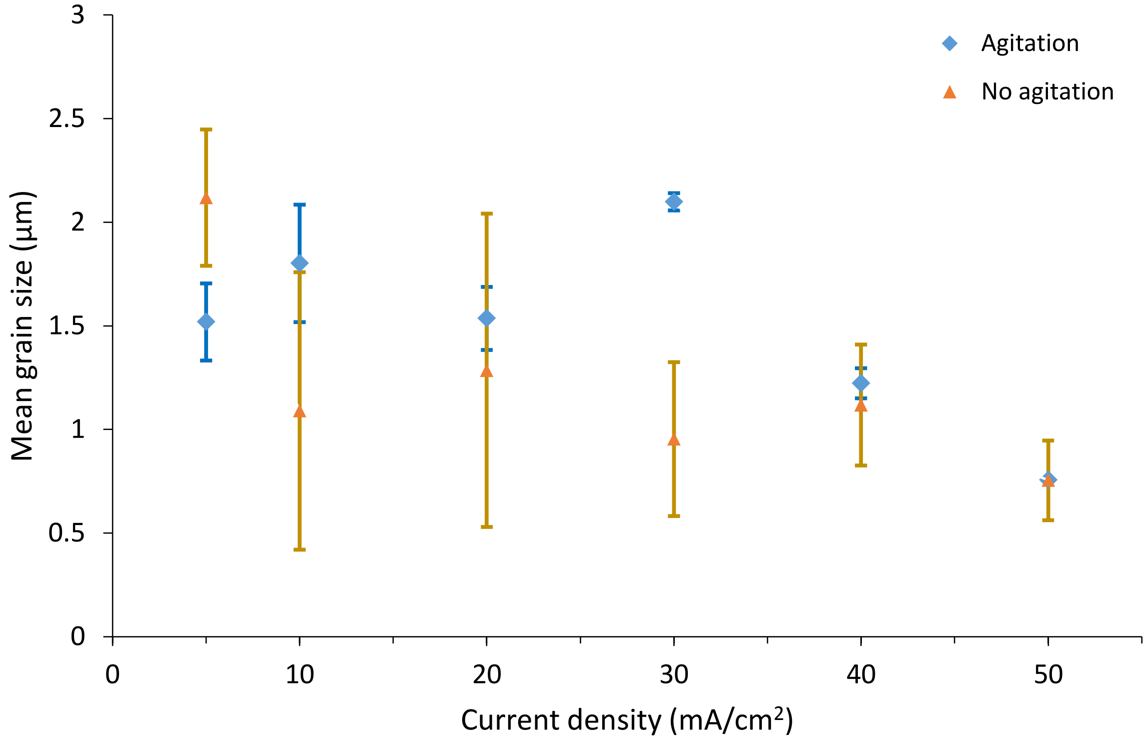


Figure 4.33: Mean grain size from tin electrodeposits produced from a Tinmac electrolyte at various current densities with and without agitation

The result in Figure 4.33 quantify and confirm the visual interpretation of the EBSD maps in Figure 4.2. These results, suggest that deposits produced without agitation exhibit smaller mean grain sizes compared to those formed using agitation (except for those produced at 5 mA/cm^2). Furthermore, at current densities of 40 and 50 mA/cm^2 mean grain sizes for agitated and non-agitated electrodeposits approach similar values.

The reduction of grain size to equivalent values at 50 mA/cm^2 could be attributed to either the nature of the electroplating bath, or that hydrogen evolution, present in the non-agitated samples caused comparable agitation to that of the agitated deposits [258]. The trend of reducing grain size with respect to increasing current density suggests that with increasing over potential grain nucleation is becoming more favourable compared to grain growth, resulting in the formation of small grains at higher current densities. Moreover, grain nucleation is discouraged by the application of agitation to the electrolyte, highlighted by the comparison of EBSD maps in Figure 4.2 for agitated and non-agitated electrodeposition at 30 mA/cm^2. These effects may be caused by an increased thickness to the diffusion layer, increased by the lack of agitation and by increased current density,

causing preferential grain nucleation [30], or it may be due to the specific additives within the electrolyte behaving differently at increasing current density and with and without agitation [259].

Additionally, electrodeposits were investigated using SEM, analysing their surface topography. Two examples of typical deposit topography are shown in Figures 4.3 and 4.4.

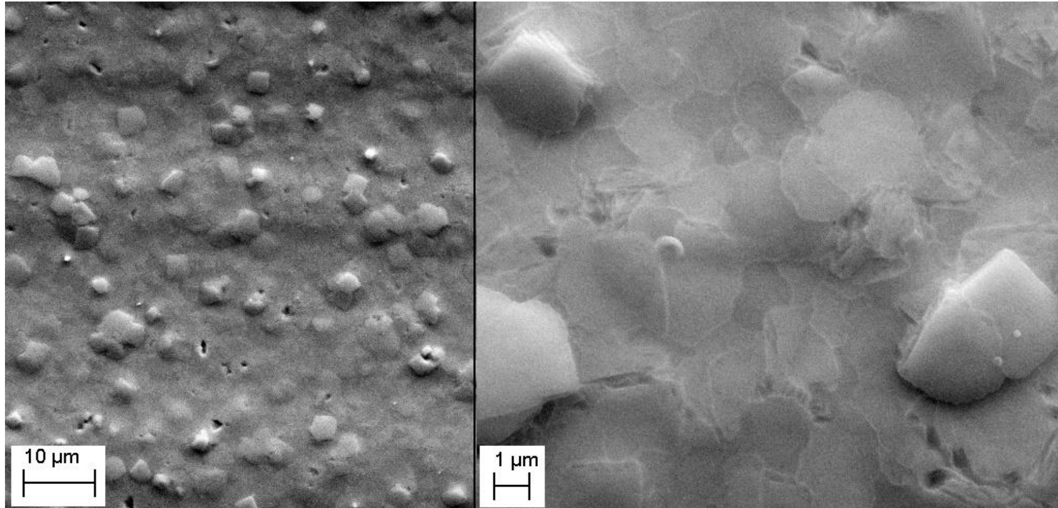


Figure 4.34: SEM micrograph of a tin electrodeposit produced at 5 mA/cm^2 with agitation showing a typical surface finish with faceted and irregular grain boundaries. Right inset shows a high magnification image

The electrodeposits either exhibited surface topography akin to Figure 4.34 or Figure 4.35. Figure 4.34 displays protruding grains with faceted boundaries within a matrix of flatter grains with irregular boundaries, whereas a high density of irregular and planar grain boundaries without protrusions are seen in Figure 4.35. For ease of classification these typical surfaces will be referred to as *faceted* and *irregular*. Electrodeposit topographical classification for deposits prepared with and without agitation are shown in Table 4.1.

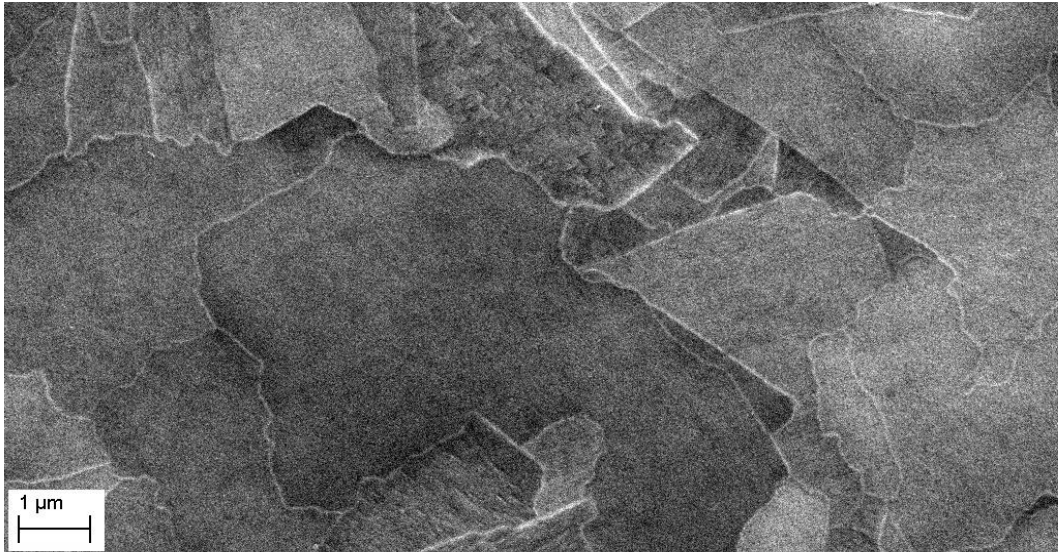


Figure 4.35: SEM micrograph of a tin electrodeposit produced at 20 mA/cm^2 with agitation showing a typical smooth surface finish with irregular grain boundaries

Table 4.1: Surface topography classification assessed by SEM for tin electrodeposits produced from a Tinmac electrolyte at varying current densities, with and without agitation.

	Current density (mA/cm^2)					
	5	10	20	30	40	50
Agitated	Faceted	Irregular	Irregular	Irregular	Irregular	Irregular
Non-agitated	Faceted	Faceted	Irregular	Irregular	Irregular	Irregular

Electrodeposits which exhibited faceted surface finishes were produced at low current densities. This was attributed to insufficient additive interactions at low current densities, or the growth of a specific grain structure due to low current density and slower growth deposition. In addition, agitation is seen to encourage deposits with irregular surface finishes for a wider range of current densities than those with no agitation.

Furthermore, if whisker growth is dependent upon deposit characteristics such as: grain size, shape, and orientation, then variations in current density and agitation may have an effect upon this phenomenon. Moreover, previous investigations of Tinmac showed increased whisker propensities at low deposition current densities [260]. This suggests that the faceted structures showing larger grains, present for deposits produced at low current densities, may increase whisker growth.

Furthermore, the natural variation in current density due to working electrode geometry will cause some areas to experience increased or decreased localised deposition rates. Areas experiencing lower current densities may run the risk of increased whisker growth. This therefore further encourages the use of larger current densities to ensure that these areas form a structure which reduces whisker propensity.

4.2 Outcome of technique development for the electrochemical anodic oxidation of tin

The electrodeposited tin was stored at room temperature for various time periods to allow the Cu–Sn IMCs to develop. The tin was then anodically dissolved from the samples leaving the copper and IMCs unaffected. In order to achieve this, a window of potential is required in which there is electrochemical activity solely due to the anodic oxidation of tin.

Figures 4.4 and 4.4 illustrate the anodic polarisation of tin and copper, respectively, in the H_2SO_4 and HCl electrolytes. As shown in Figure 4.36, the onset of tin oxidation occurred at approximately -0.5 V (vs. SCE) in both electrolytes. The onset of copper oxidation in HCl was found to be dependent upon concentration, as shown in Figure 4.37, with initial oxidation potentials of -0.17 V and -0.1 V (vs. SCE) for 5 % and 1.2 % HCl (S.G. 1.16), respectively. H_2SO_4 did not exhibit any concentration dependency for the two concentrations examined; it also produced a higher rate of oxidation for tin and had a constant oxidation onset potential for copper of +0.05 V (vs. SCE).

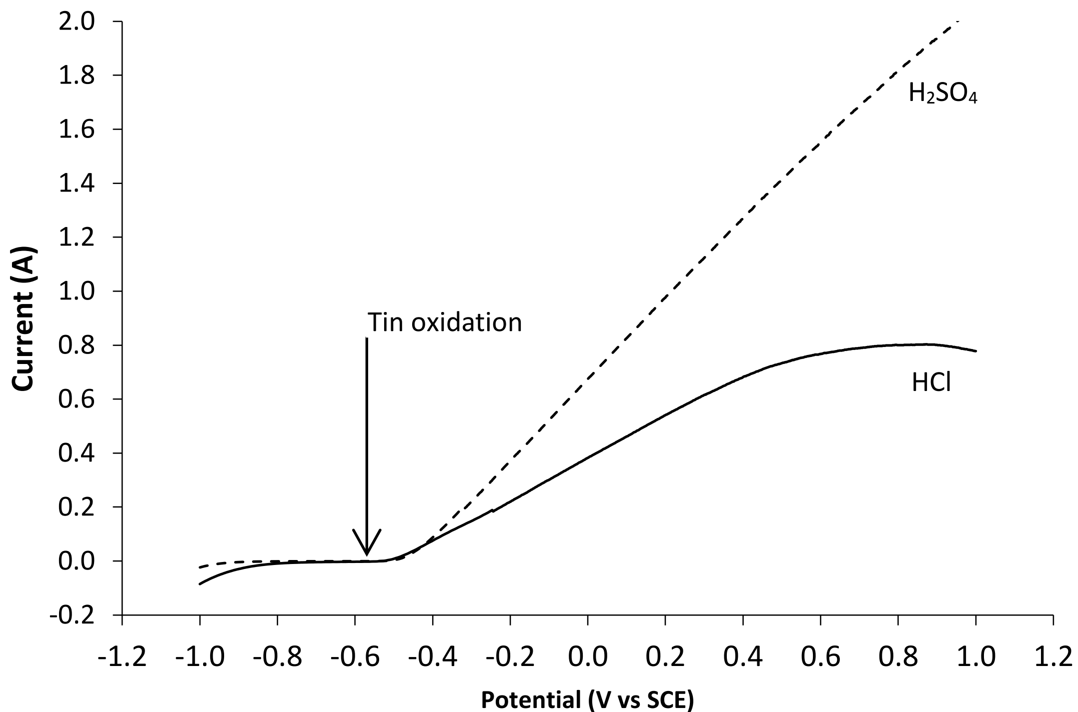


Figure 4.36: Anodic polarisation curves from -1.0 V to +1.0 V (vs. SCE) for a 4 cm^2 tin sheet electrode in 5 % H_2SO_4 (S.G. 1.83) (dashed line), and 5 % HCl (S.G. 1.16). The arrow indicates the onset of tin oxidation

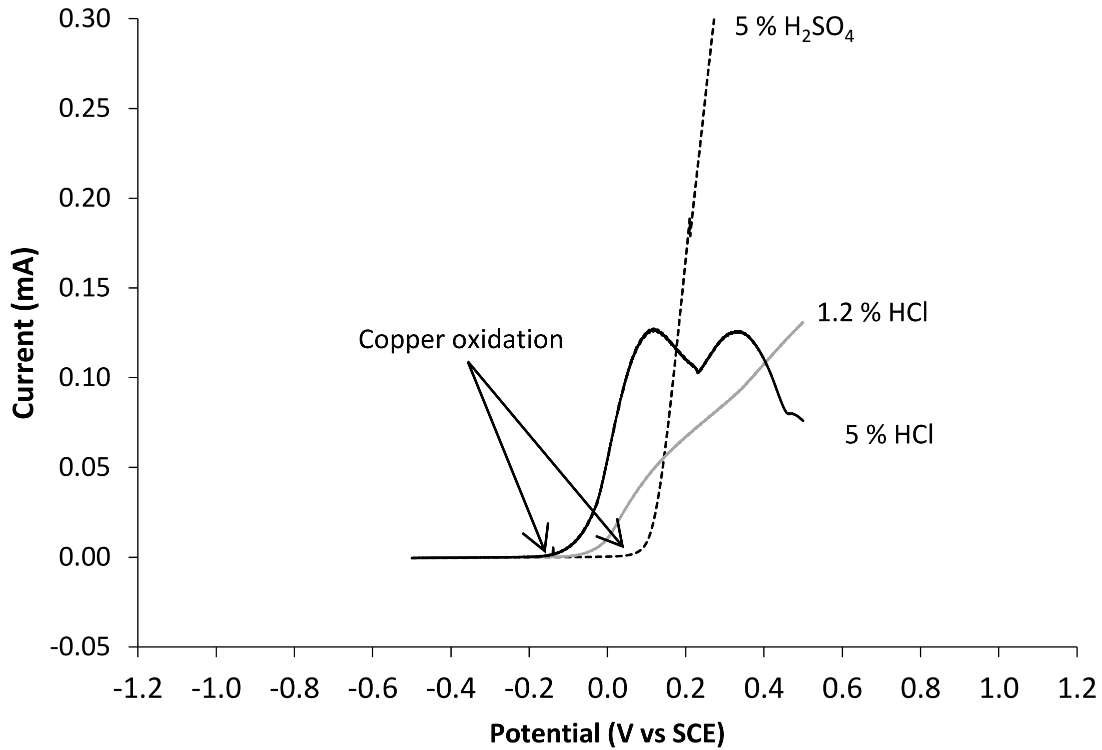


Figure 4.37: Anodic polarisation curves for a 4cm^2 copper electrode in 5 % H_2SO_4 (S.G. 1.83) (dashed black line), 5 % HCl (S.G. 1.16) (solid black line) and 1.2 % HCl (S.G. 1.16) (solid grey line). The arrows indicates the onset of copper oxidation

Figures 4.4 and 4.4 identify the potentials where only the oxidation of tin occurs i.e. potentials above the onset of tin oxidation and below the onset of copper oxidation, of 0.33 V for HCl and 0.55 V for H_2SO_4 for the 5 % concentration electrolytes. In addition to H_2SO_4 producing a larger potential window, it created a more uniformly oxidised sample surface, shown in Figure 4.38. For these reasons 5 % S.G. 1.83 H_2SO_4 was chosen as the more favourable electrolyte for the potentiostatic electrochemical stripping process.

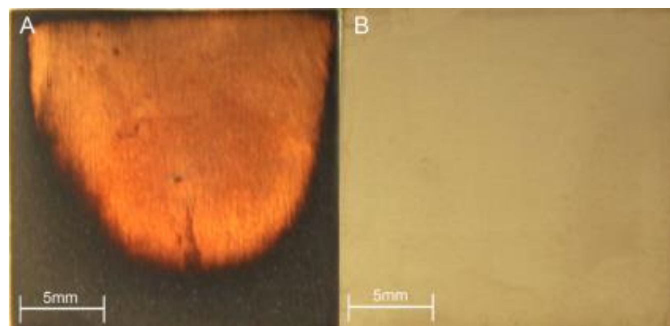


Figure 4.38: Surface appearance of tin electroplate copper coupons after electrochemical stripping using two different electrolytes: (A) 5 % HCl (S.G. 1.16) and (B) 5 % H_2SO_4 (S.G. 1.83)

Galvanostatic stripping was also undertaken in both HCl and H_2SO_4 solutions to determine the potential at which the process should be terminated i.e. the potential indicating that all the tin had been removed whilst leaving the copper and IMCs unaffected. Potential-time data used to calculate this potential is shown in Figure 4.39.

The potential-time data in Figure 4.39 for the 5 % H_2SO_4 electrolyte displayed two clear potential plateaux; representing tin (approximately -0.4 V vs. SCE) and copper (approximately 0.1 V vs. SCE) oxidation to Sn^{2+} and Cu^{2+} respectively, as shown in Table 4.2. In comparison, the data for 5 % HCl displayed three plateaux at; -0.4 V, attributed to tin oxidation and also two further plateaux at -0.06 V and +0.3 V (vs. SCE) the latter is highlighted in the magnified area in Figure 4.39. The reactions responsible for these latter plateaux have been identified in the literature as Cu_6Sn_5 and Cu_3Sn oxidation [255]. However, since the samples were stored at room temperature the formation of Cu_3Sn was unlikely as higher temperatures (>60 ˚C) are reported to be required for it to form [189].

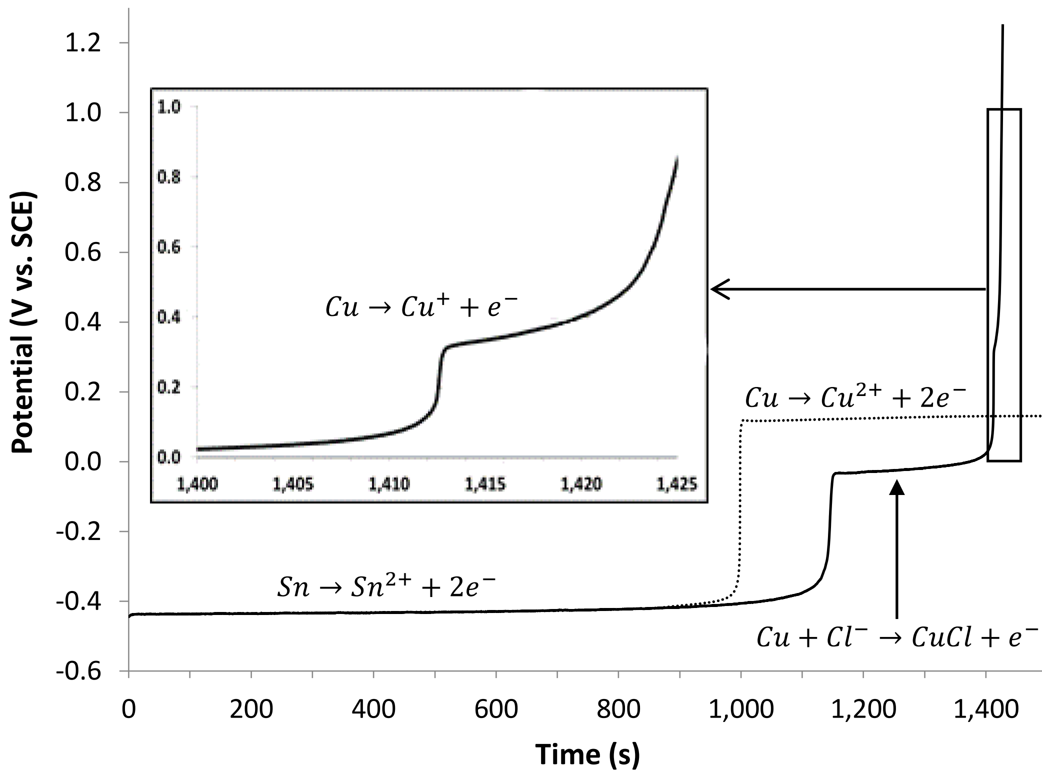


Figure 4.39: Potential-time data for the galvanostatic stripping of tin at 10 mA/cm^2 from a 10 μm electrodeposited coated copper sample in 5 % H_2SO_4 (S.G. 1.83, dashed line) and 5 % HCl (S.G. 1.18, solid line). The insert graph shows a magnified view of the plateau at 0.3 V vs. SCE for the latter solution

Additionally, in Figure 4.37 successive oxidation peaks at +0.08 V and +0.3 V (vs. SCE) are apparent for copper in HCl. Therefore, the two additional plateaux (-0.06 V and +0.3 V vs. SCE) found

during galvanostatic stripping with HCl are thought to correspond to copper oxidation to form Cu^+ and CuCl, as shown in Table 4.2, rather than dissolution of IMCs. Thus a solution of 5 % (S.G.1.83) H_2SO_4 was preferred for subsequent use because of reduced complications involving successive copper oxidation and a larger potential window between tin and copper oxidation. Therefore, the termination potential for this technique was defined as any value above 0 V vs. SCE.

Table 4.2: Relevant tin and copper oxidation standard potentials compared to measured potentials, illustrated in Figure 4.39 [256]

Electrolyte	Reaction	Standard Potential		Measured Potential	
		(V vs SHE)	(V vs SCE)	(V vs SHE)	(V vs SCE)
HCl, H_2SO_4	$Sn \rightarrow Sn^{2+} + 2e^-$	-0.14	-0.38	-0.16	-0.40
HCl	$Cu \leftrightarrow Cu^+ + e^-$	+0.52	+0.28	+0.54	+0.30
H_2SO_4	$Cu \leftrightarrow Cu^{2+} + 2e^-$	+0.34	+0.10	+0.32	+0.08
HCl	$Cu + Cl^- \leftrightarrow CuCl + e^-$	+0.12	-0.12	+0.18	-0.06

The surface structure that results from galvanostatic stripping a 5 µm tin electrodeposited coating for a copper substrate after 24 h can be seen in Figure 4.40. The micrograph shows that, using the galvanostatic method, ‘mossy’ features and IMC particles remain on the copper substrate. The mossy features that are shown in Figure 4.40 are thought to be residual tin. Although the IMC growth is distinct in these micrographs improvements to the process to remove the mossy features were thought to be beneficial for the analysis of smaller IMC particles.

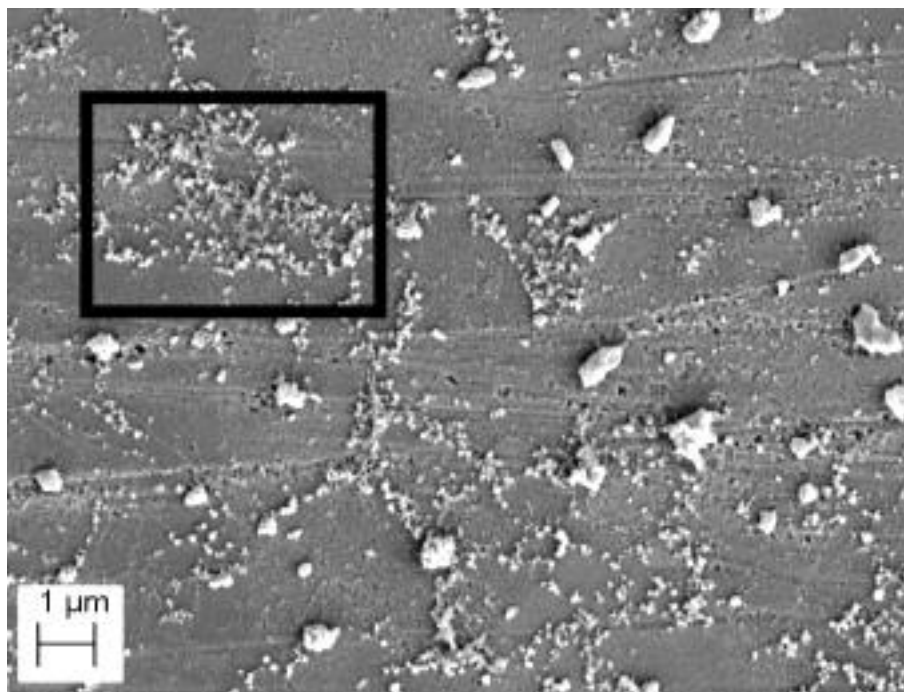


Figure 4.40: SEM micrograph of 24 h of IMC growth on an electrochemically polished copper substrate after the 5 µm electrodeposited tin coating was galvanostatically stripped. In the enclosed area shows an example of what is thought to be residual tin remaining on the surface after stripping

Tin electrodeposit stripping using a potentiostatic method requires a potential to be selected where there is solely tin anodic oxidation which allows for the complete removal of tin closest to the substrate even when its surface area is reduced in the final stages of stripping. Based on the data presented in Figures 4.4 and 4.4, potentials between the copper and tin oxidation (+0.05 V and -0.5 V vs. SCE) would rapidly oxidise the tin to the point of leaving just the copper and IMCs unaffected with close to zero current being passed. For this reason -0.26 V (vs. SCE) was selected for stripping, which is approximately midway between the potentials for the onset of tin and copper oxidation. Additionally, Figure 4.39 shows no additional potential plateau between the tin and copper stripping potentials in H_2SO_4. This would indicate that the IMC oxidation potential is not located between the tin and copper oxidation potentials but is greater than the copper oxidation potential. This, therefore, suggested that a potential of -0.26 V (vs. SCE) would not result in significant anodic oxidation of the IMCs. Figure 4.41 shows an example of a current–time plot for the potentiostatic stripping of a nominal 10 µm tin deposit at -0.26 V (vs. SCE).

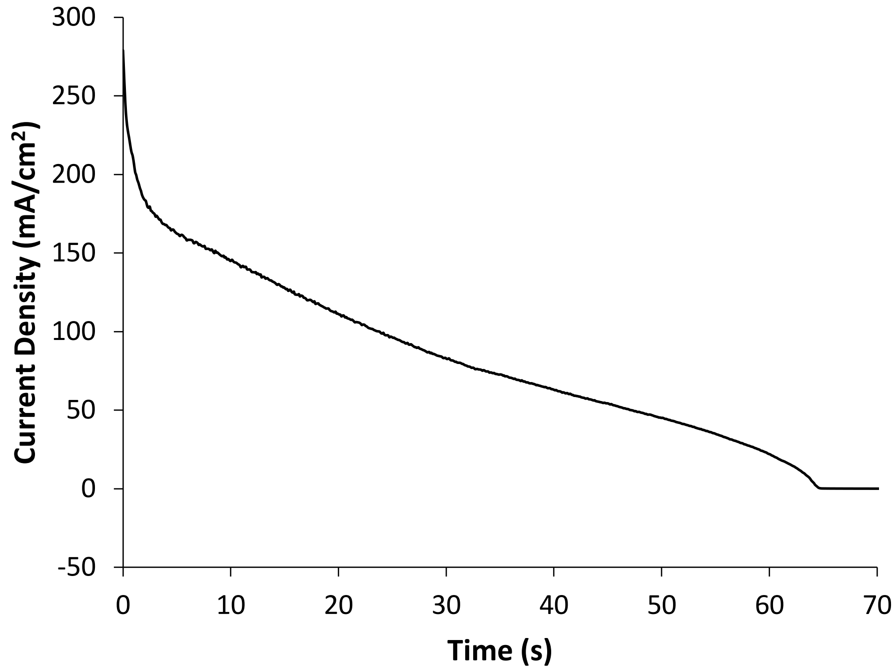


Figure 4.41: Current density vs time data for the potentiostatic stripping of a nominally 10 µm tin coated copper coupon maintained at -0.26V vs. SCE

An example of a potentiostatically stripped 5 µm tin electrodeposit after 24 h of IMC growth is seen in Figure 4.42. It is evident from this micrograph that the tin coating has been totally removed (cf. Figure 4.40) whilst leaving the IMCs unaffected and as such is a more advantageous way of coating removal compared to galvanostatic stripping. The greatest concern with using a potential of -0.26 V (vs. SCE) is that this results in current densities exceeding 250 mA/cm^2 (Figure 4.41). This high rate of oxidation may be beneficial in terms of the time required to undertake this process, but occasionally an insoluble copper sulphate, rich in copper, sulphur and oxygen (Cu: 37 wt-%, S: 17 wt-%, O: 43 wt-%, defined by EDS), formed on the surface of the coating from the electrochemical reaction of the sulphuric acid and copper which then prohibited IMC analysis. Copper sulphate often dissolves in acidic solutions, but due to the low acidity of the electrolyte this was not experienced in this case. This resulted in large areas which were not able to be analysed, as show in Figures 4.43 and 4.44.

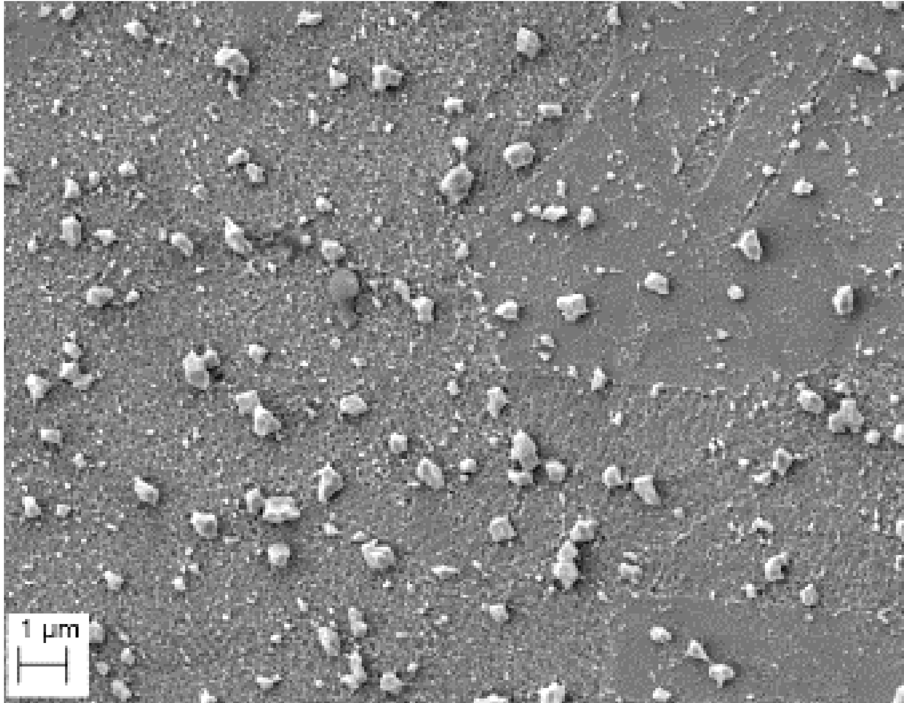


Figure 4.42: SEM micrograph of 24 h IMC growth on electrochemically polished copper substrate after 5µm of electrodeposited tin was potentiostatically stripped

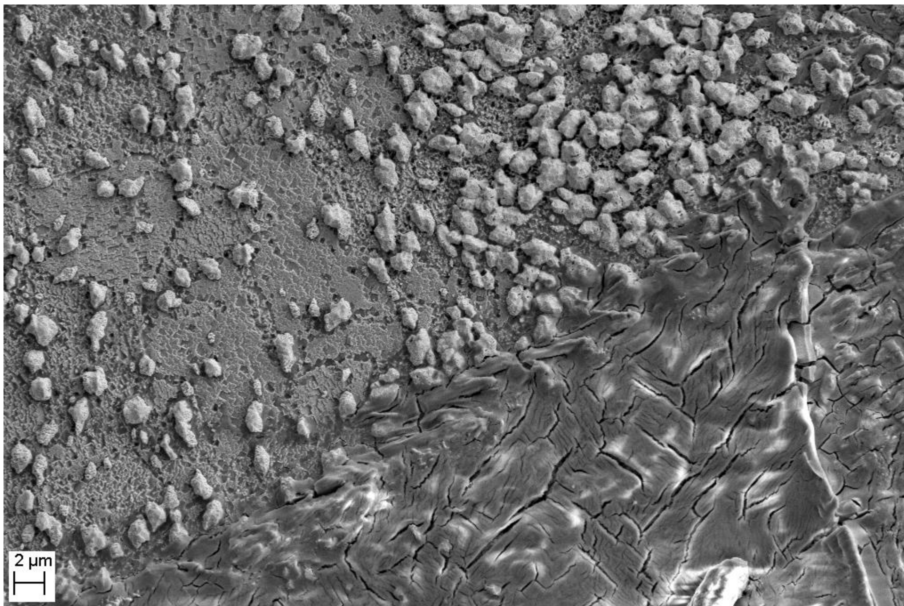


Figure 4.43: SEM micrograph of IMC growth on an electropolished copper substrate after 17 days in storage exhibiting damage by sulphate build up during potentiostatic stripping

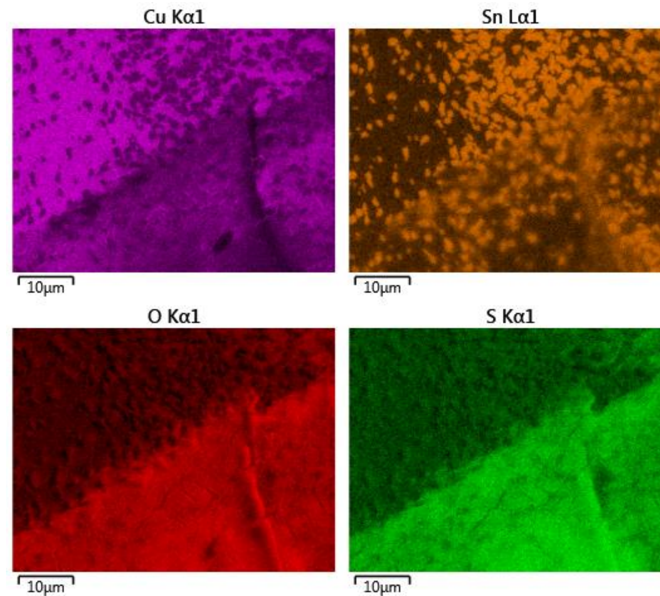


Figure 4.44: EDS map of Cu, Sn, O and S for the area shown in Figure 4.43

A solution to inhibit this damaging sulphate build up was the sequential combination of both techniques, i.e. galvanostatic stripping to remove the majority of the tin electrodeposit in a controlled manner that inhibits sulphate growth, followed by potentiostatic stripping to remove the remaining tin. This process resulted in equivalent results but with no issues arising from sulphate formation.

4.3 Intermetallic compound growth for deposit produced from an aqueous electrolyte at 20 mA/cm²

The effect of storage time upon the morphology and distribution of IMCs was investigated with periods of storage ranging between 1 to 55 days for electrodeposits produced from the aqueous electrolyte, Tinmac, on electropolished copper substrates. Moreover, there are two possible compositions of Cu–Sn IMCs, namely Cu_3Sn and Cu_6Sn_5 [189], [190]. Initial investigations were undertaken to confirm that Cu_6Sn_5 IMCs were formed. Elemental analysis of the 3 day old IMCs, using EDS, showed an average composition of 41±1 at.% copper and 58±1 at.% tin; identifying the IMC as Cu_6Sn_5.

Figure 4.45 shows the distribution and particle size of the IMCs after the sample had been storage at room temperature for 24 h and 17 days before being stripped if its metallic tin. After 24 h, the IMCs form small individual particles with a variation in particle density in patterns across the surface of the substrate.

These distinct patterns are comparable in shape and size to the copper grains seen in Figure 3.2.B and C. This indicates that initial IMC growth was related to the underlying copper grains as different copper grain orientations are producing different densities of IMC. After 17 days the IMC

particles have coarsened and begun to coalesce in some areas. In addition, small individual particles of IMC are visible after 17 days indicating that either IMC nucleation is still occurring or that there is a wide range of IMC particle coarsening rates taking place. The variation in IMC particle density that was evident after 24 h is also evident after 17 days. In addition, preferential IMC growth is also apparent at the copper grain boundaries. It is clear that the rate of IMC nucleation, coarsening and coalescence is increased at the copper grain boundaries. An example of this boundary specific nucleation and coarsening is shown in Figure 4.46.

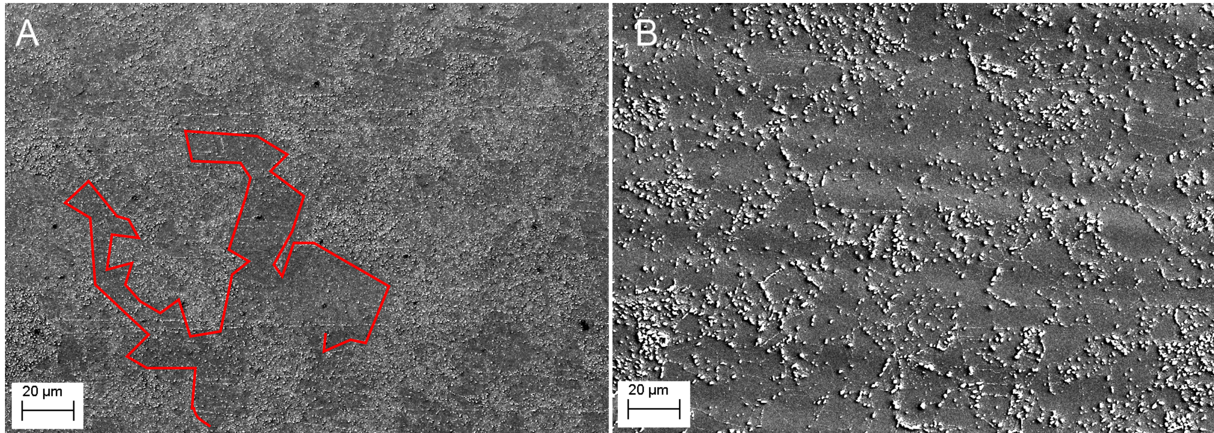


Figure 4.45: SEM micrographs of IMC growth formed from a Tinmac electrodeposit on electropolished copper substrates; A – Tin stripped after 24 h growth at ambient conditions showing superimposed red line depicting low-high IMC density boundary; and B - 17 day storage at ambient conditions

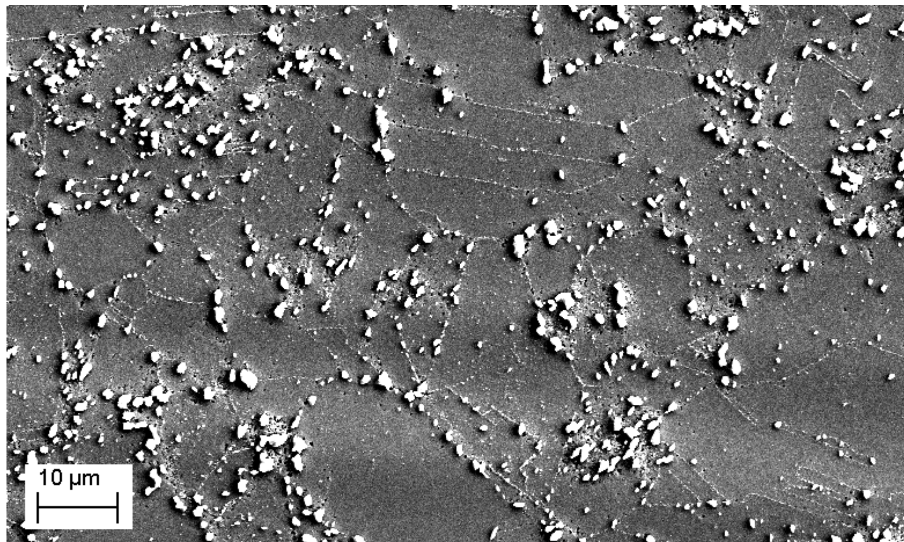


Figure 4.46: SEM micrograph of IMC growth formed from a Tinmac electrodeposit upon electropolished copper substrate and being tin tripped after 17 days storage at ambient conditions. Featuring IMC growth along copper grain boundaries

To eliminate the possibility that the variation in the density of IMC growth was either as a result of a localised change to micro-roughness or surface chemistry induced by the electropolishing

process, copper substrates were polished by an alternative method, i.e. mechanical polishing. From Figure 4.47 it is clear that the same variations in IMC density that were observed on electropolished samples (Figure 4.47.A and B) were also observed on samples that had been mechanically polished (Figure 4.47.C and D). This shows that the variation in IMC density and growth is independent of the polishing method used, which confirms that the heterogeneity in IMC density is due to the grain orientation of the underlying copper substrate.

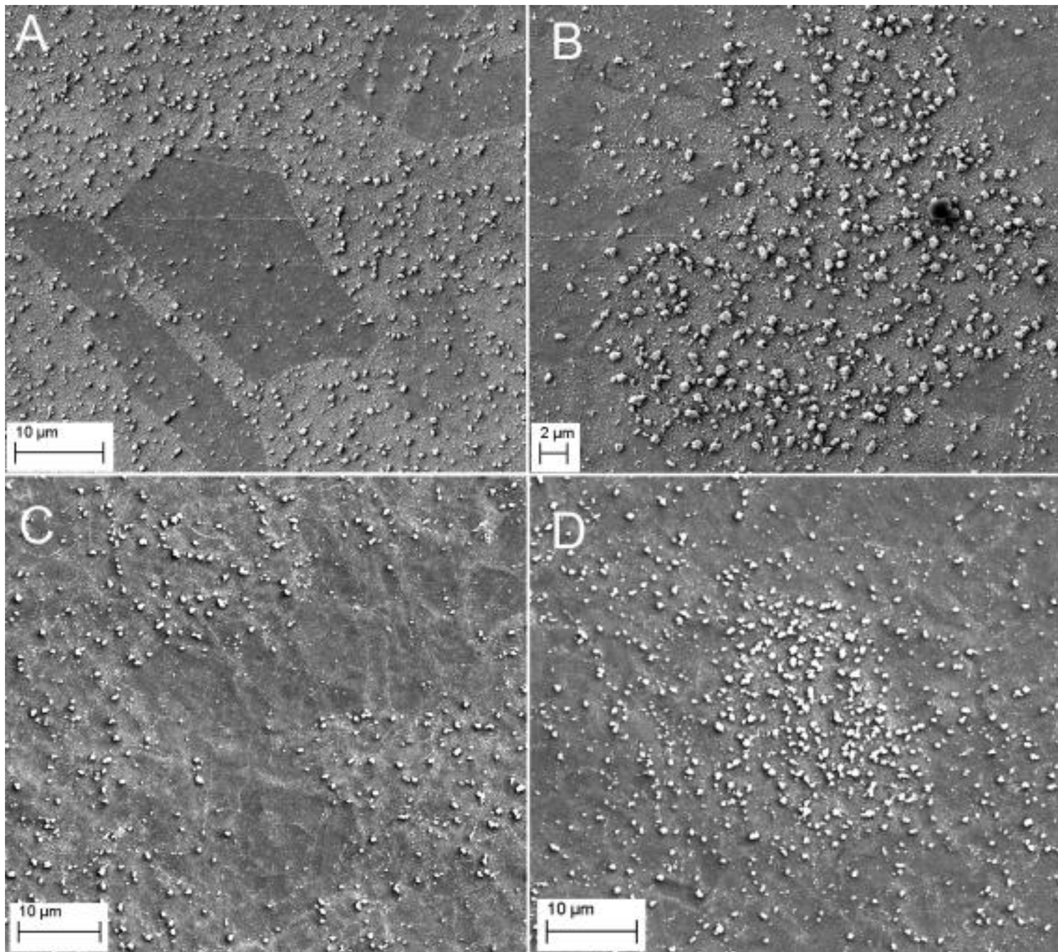


Figure 4.47: SEM micrographs of IMC growth formed from a Tinmac electrodeposit upon electropolished copper substrate after 24h storage at ambient conditions before tin stripping on; electropolished copper substrates; ‘A’ and ‘B’, and mechanically polished substrates; ‘C’ and ‘D’, showing areas of low density IMC growth; ‘A’ and ‘C’ and high density IMC growth ‘B’ and ‘D’

In addition, EBSD was undertaken for samples from which the tin was stripped after 17 days IMC growth to analyse the orientation of the copper substrate. Although a large number of data points were not able to be indexed due to topographic obstruction, sufficient data were collected to create the pole figure shown in Figure 4.48.

Therefore, by comparing the difference between inverse pole figures in Figure 4.48 and Figure 3.2 (as repeated next to Figure 4.48), we can infer that IMC growth was most pronounced in areas

which have points in Figure 3.2 but not in Figure 4.48, i.e. the pole figure maps for Figure 4.48 and Figure 3.2 should be identical as they are both electropolished copper from the same source, but due to IMC obstruction in specific locations on the substrate in Figure 4.48, data created from the analysis of the copper substrate after the removal of a 17 day old tin electrodeposit, data points are lost. Therefore, the data points which are shown in Figure 3.2 and not in Figure 4.48 are the ones which are obstructed from analysis by IMC growth.

For this comparison it is apparent that data points located in the region of 001 are mostly absent from Figure 4.48. Suggesting that the 001 copper grain orientation accelerated IMC growth more than other crystallographic orientations.

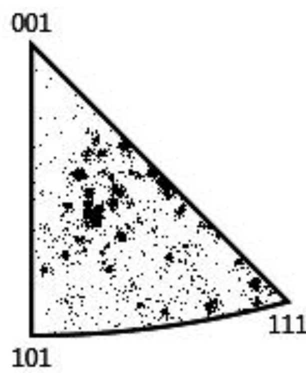


Figure 4.48: Inverse Z (surface normal) folded pole figure of accessible copper substrate with 17 day old IMC growth

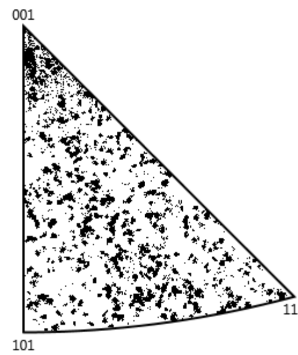


Figure 3.2 (repeated figure): Z-fold (surface normal) inverse pole figure for electropolished copper substrate used throughout this project

Moreover, closer inspection of the copper substrate reveals a further aspect to the IMC distribution, as shown in Figure 4.49. The micrograph shows small IMC particles aligning along an additional structure within the large copper grains. This IMC distribution could be related to either subgrain boundaries within the copper or grain boundaries in the removed tin electrodeposit.

Typically, the grain size of the copper in this study was much larger than 1 µm (e.g. Figure 3.2 and Figure 4.50) whereas the network of IMCs seen in Figure 4.49 suggests a feature size of less than 1 µm with a regular polygonal structure. Additionally, the EBSD analysis undertaken for the electropolished copper substrates, presented in Figure 4.50, demonstrates no fine grained substructure is present. This would suggest that the fine structure seen in Figure 4.49 is not due to copper subgrains, but rather the structure of the overlying tin coating.

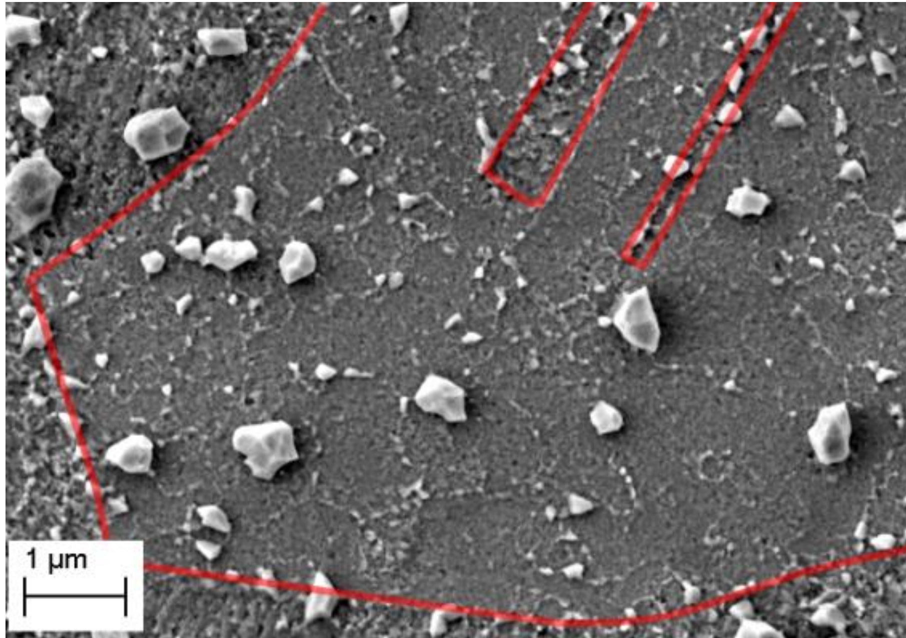


Figure 4.49: SEM micrograph of IMCs formed from a Tinmac electrodeposit upon electropolished copper after 24 h at ambient conditions before tin stripping showing finer IMC structure than that found on copper grain boundaries. The latter are highlighted

IMC growth after 55 days of storage was also investigated and an example is shown in Figure 4.51. This indicates an almost continuous layer of IMCs with small, sporadic patches distributed across the whole of the sample where IMC particles had not formed. EDS mapping indicates a lower tin concentration in these regions, which is consistent with the absence of IMCs (Figure 4.52).

Figure 4.50: SEM–EBSD Euler map of an electropolished copper substrates used throughout this study showing grain orientation

In addition to a storage period of 55 days, samples of electrodeposited tin on as-received rolled copper substrates were stored for 445 days and electrochemically stripped my applying an anodic

potential within an electrolyte of dilute sulphuric acid. Examples of the IMC growth are shown in Figure 4.53. These micrographs show that a complete layer of IMC particles has formed along the interface after 445 days. The size (c.a. 1 x 3 µm) and morphology of the IMCs after 55 and 445 days were similar. In both samples the IMCs displayed a facetted surface, which was more pronounced in the 455 day sample.

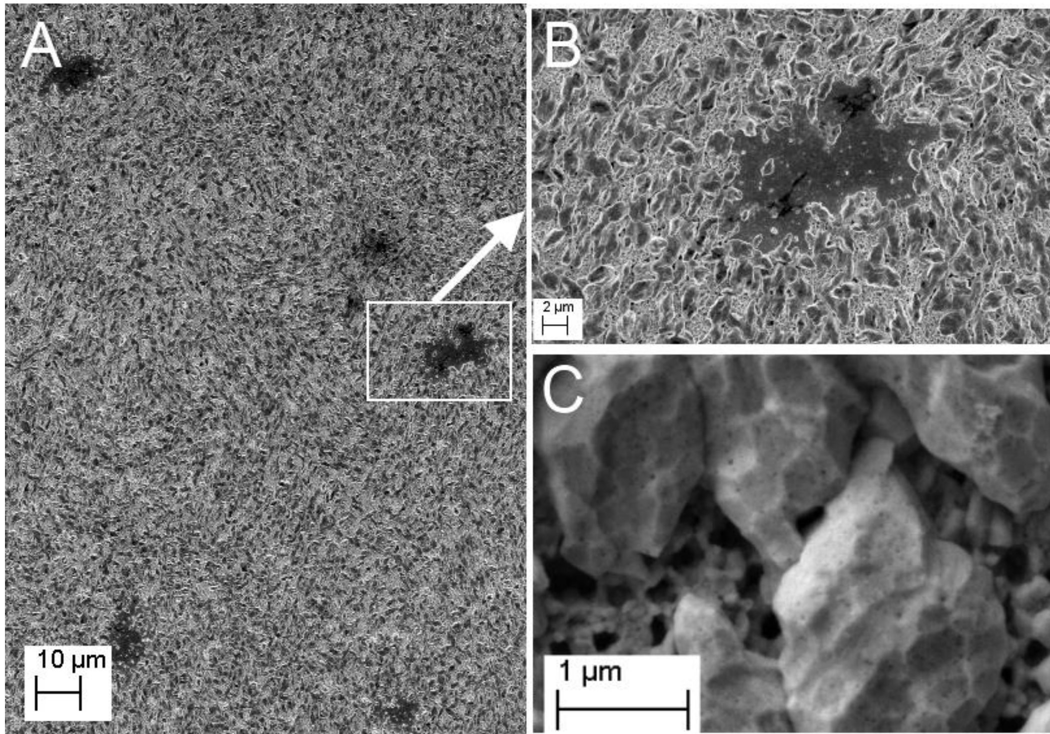


Figure 4.51: SEM micrographs showing IMC growth formed from a Tinmac deposit upon electropolished copper substrates after 55 days growth at ambient featuring an almost complete layer of IMC. 'A' and 'B' show areas with no IMCs at increasing magnification, and 'C' at high magnification showing detail of the IMC structure

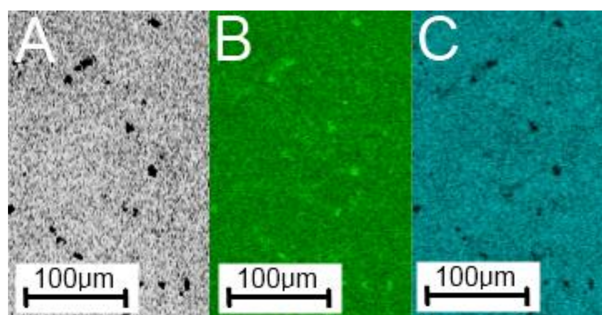


Figure 4.52: Secondary electron image and EDX maps of IMC growth after 55 days storage, as seen in Figure 4.51: A; secondary electron SEM micrograph, B; X-ray map for copper Kα and C; X-ray map for tin Lα

In addition, although this sample had tin electrodeposited onto an unpolished copper substrate containing rolling lines, there was no evidence of these features in the distribution of the IMC. This

would indicate that for samples of this age the distribution of the IMCs is independent of the rolling lines, copper grain orientation and copper grain boundaries.

Furthermore, samples which underwent tin electrodeposition from the Tinmac electrolyte on rolled copper substrates immediately followed a 30 min heat treatment at 150°C and then by 7 days storage at room temperature before being anodically tin stripped. These samples exhibited IMCs as seen in Figure 4.54.

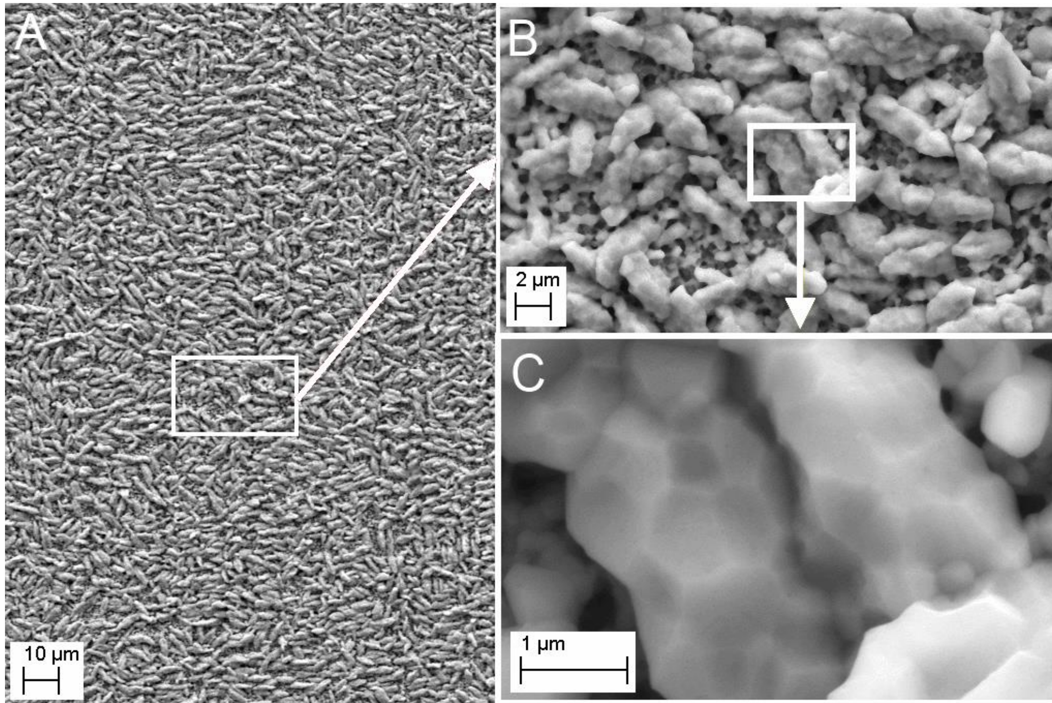


Figure 4.53: SEM micrographs of IMC growth from a Tinmac electrodeposit after 445 day storage at ambient prior to tin stripping on as–received rolled copper. 'A' showing complete coverage of the surface and 'B' and 'C' showing more detailed structure at higher magnifications

From Figure 4.54 the heat treated sample was completely covered by fine and facetted IMC growth. Comparing this to the IMC growth shown in Figure 4.46, stored for a longer time period (17 days), it is clear that the heat treatment has accelerated the coverage of IMCs, showing that a heat treatment creates a more uniform IMC structure within far shorter time periods. In addition, the morphology of the IMCs as seen in Figure 4.54 has clearly changed showing particles which are smaller in size and aspect ratio which could be due to the formation of Cu_3Sn at elevated temperatures.

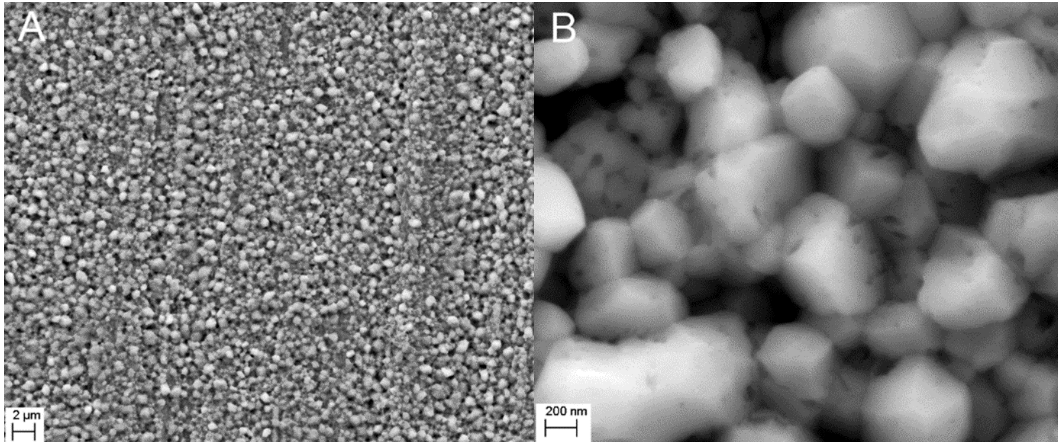


Figure 4.54: SEM micrographs showing IMC growth distribution formed from a Tinmac electrodeposit after heat treatment (at 150^{0}C for 30 mins) and 7 days at ambient prior to anodic electrochemical tin stripping on as–received rolled copper substrate. 'A' showing complete coverage of the surface and 'B' showing more detailed structure at higher magnifications.

After 17 days storage, IMC growth was dependent on the grain orientation and grain boundary structure of the copper substrate and the grain boundary structure of the tin coating. The larger IMC particles are due to the copper substrate rather than the tin coating structure which indicates that the rate of IMC growth is more dependent upon the copper substrate grain orientation and grain boundary structure than the structure of the tin deposit.

After 55 days storage near complete coverage of the interface was observed with only sporadic patches where IMCs did not grow. In comparison samples stored for 445 days exhibited 100% coverage. Moreover, if there was a continued increase in IMC coverage comparable to that seen between 1, 17 and 55 days then it is thought that soon after 55 days the interface will be essentially covered by IMCs.

By comparing these results to other studies focused upon IMC formation of tin electrodeposits on copper substrates [83], [191] shows the formation of a complete layer of IMC is uncommon. Moreover, the rate of which this layer is formed suggests a comparatively high rate of IMC formation for this specific coating. It has been shown previously that IMC growth is dependent upon deposit characteristics which themselves are produced by specific electrolyte formulations [83]. Therefore, this suggests that the IMC growth and distribution seen for deposits produced from a Tinmac electrolyte form rapidly into a complete layer fundamentally because of specific additives used within the electrolyte.

4.4 Electrodeposition from non- aqueous deep eutectic solvents

The study of electrodeposition from three DESs has been separated into: electrolyte voltammetry and cathode current efficiency and the optimisation of electrodeposit quality by: electrolyte formulation, $SnCl_2.2H_2O$ concentration and hydrogen bond donor; and deposition current density variation during deposition.

4.4.1 *Non-aqueous electrolyte DC voltammetry and cathode current efficiency measurements for tin electrodeposition*

4.4.1.1 *Cathodic reactions*

Voltammetry undertaken with additions of 0.088 M $SnCl_2.2H_2O$ at 70 ^oC in Reline, Ethaline and Propeline are shown in Figure 4.55. The voltammograms demonstrate four regions, summarised in Table 4.3, they are: tin reduction, electrolyte reduction, tin oxidation and electrolyte oxidation.

Table 4.3: tin and electrolyte reduction and oxidation onset potentials for Reline, Ethaline and Propeline with additions of 0.088 M $SnCl_2$

Electrolyte	Onset potential (V vs Ag wire)				Electrolyte potential window (V)	Tin reduction to electrolyte reduction potential window (V)
	Tin reduction	Tin oxidation	Electrolyte reduction	Electrolyte oxidation		
Reline	-0.48	-0.41	-1.50	+2.02	3.52	1.02
Ethaline	-0.32	-0.24	-1.33	+2.23	3.56	1.01
Propeline	-0.18	+0.01	-1.29	+1.76	3.05	1.11

The potential window between anodic and cathodic electrolyte degradation for all three electrolytes is seen to be large compared to aqueous alternatives (1.23 V [67]). Therefore, this also suggests the possibility of the electrodeposition of more noble metals from these solutions [91]. Furthermore, the potential at which tin reduction initiates produces a window between itself and the onset of electrolyte reduction. For each electrolyte examined this is also seen to be comparatively large (approximately 1 V) compared to aqueous electrolytes seen within literature (approximately 0-0.5 V)[44], [60], thus offering a wide range of potentials to be examined which permit the deposition of tin without electrolyte degradation.

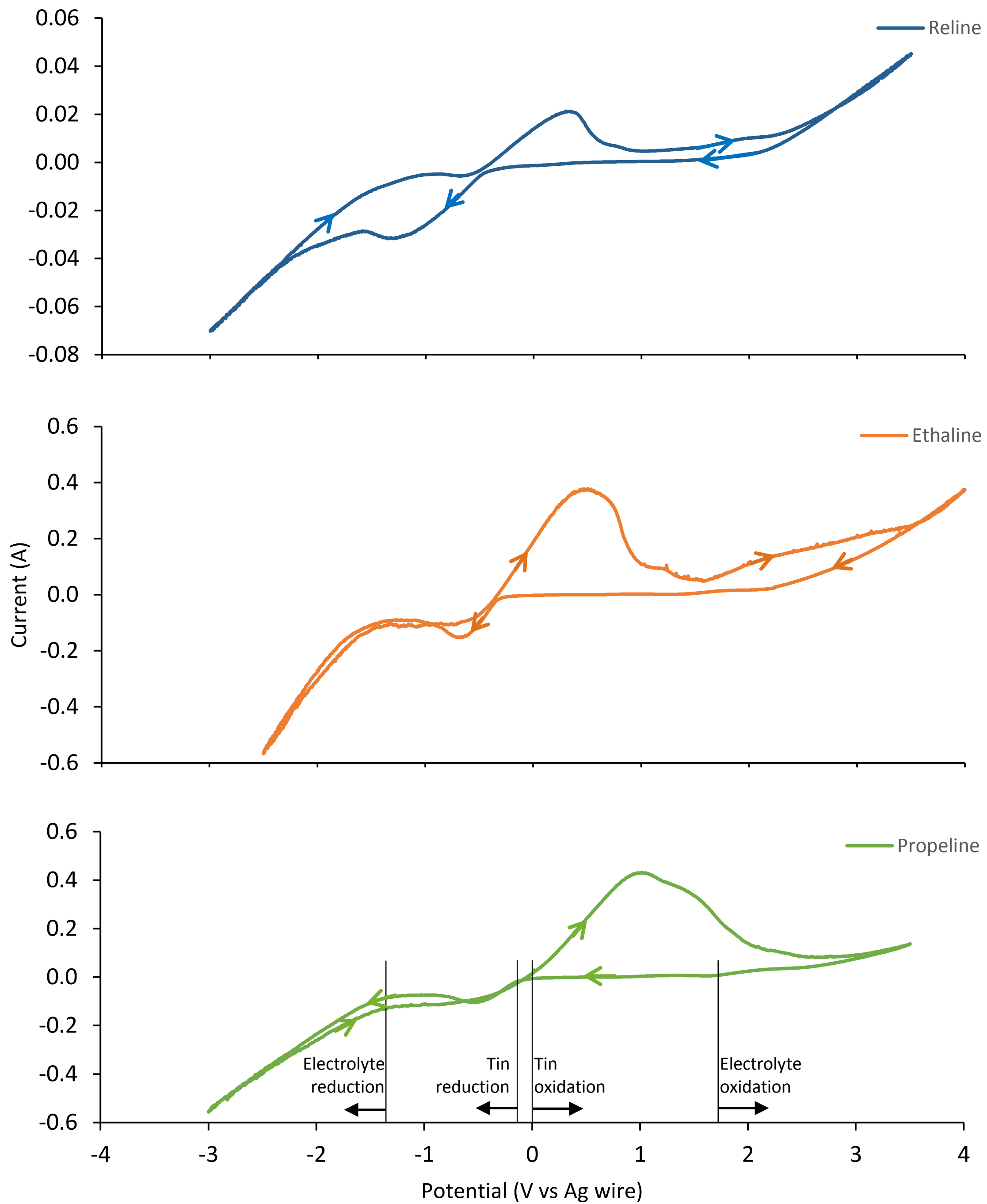


Figure 4.55: Voltammograms for Reline, Ethaline and Propeline with 0.088 M $SnCl_2.2H_2O$ at 50 mV/s using a platinum gauze working electrode. With example annotations for Propeline showing specific onset potentials. Voltammograms starting with open circuit and then becoming increasingly negative and terminating at open circuit.

Comparing voltammetry in Figure 4.55, Reline produces currents a magnitude smaller than with Ethaline and Propeline. This is most likely due to its ten times greater viscosity. Although the wide tin reduction-electrolyte reduction potential window suggests efficient high rate deposition, due to high viscosities the 1 V potential window produces a limited current increase. Enlarging the range of

possible currents over this 1 V window was achieved by increasing the electrolyte $SnCl_2.2H_2O$ concentration.

Cyclic voltammetry was undertaken for the three DESs whilst varying $SnCl_2.2H_2O$ concentrations. It was found that the three electrolytes behaved in the same manner, therefore the voltammetry in Figure 4.56 for Reline with additions of 0, 0.044, 0.35 and 0.71 M $SnCl_2.2H_2O$ summarises results seen from Ethaline and Propeline shown in Chapter 10 Appendix, 10.1 Non-aqueous electrolyte DC voltammetry for tin electrodeposition, page 179.

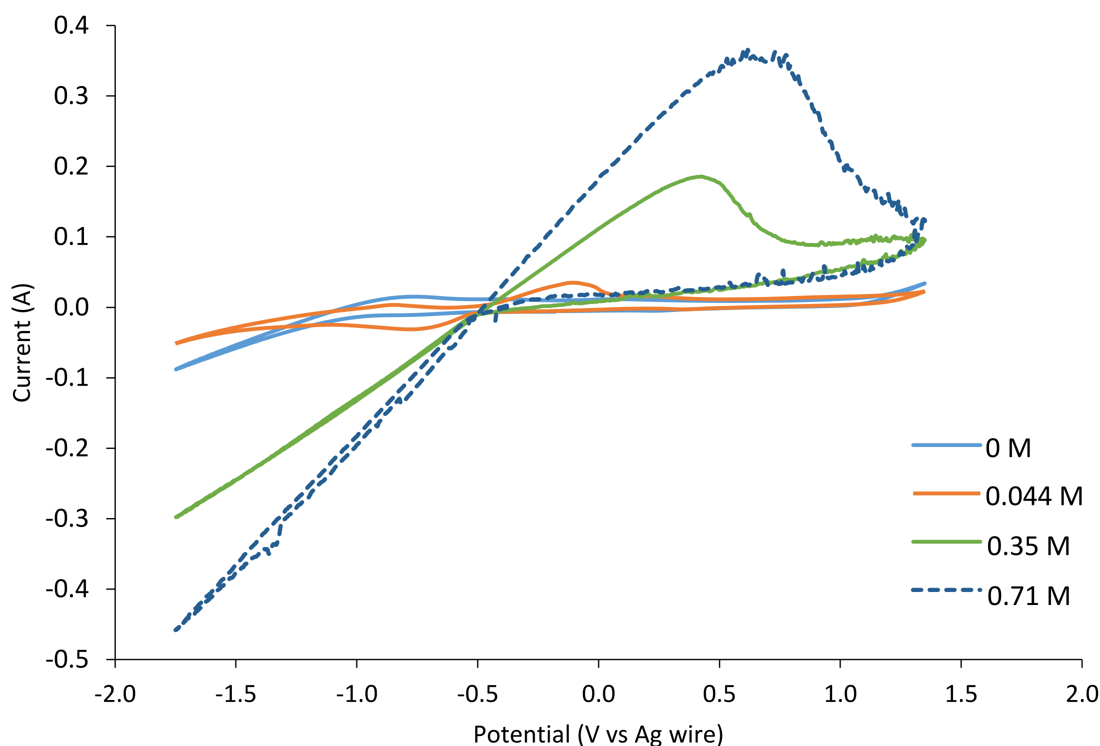


Figure 4.56: Voltammogram for Reline with additions of $SnCl_2.2H_2O$ to concentration of 0, 0.044, 0.35, 0.71 M at 70 ^{o}C and 50 mV/s

The addition of 0.044 M $SnCl_2.2H_2O$ to Propeline produced a reduction peak evident at approximately -0.7 V, thought to be Sn^{2+} reduction to Sn metal. Additionally, a decrease in current caused for electrolyte reduction, below -1 V, for 0.044 M $SnCl_2.2H_2O$ compared to 0 M $SnCl_2.2H_2O$ suggests that increasing $SnCl_2.2H_2O$ concentration decreases electrolyte reduction. This was most likely due to the change in surface chemistry during tin deposition for the electrolyte containing 0.044 M $SnCl_2.2H_2O$ as the electrodes surface is coated in tin decreasing the potential at which electrolyte degradation takes place. Furthermore, with greater $SnCl_2.2H_2O$ concentration there is an increasingly steepened ohmic relationship for tin reduction at 80 and 0.71 M $SnCl_2.2H_2O$. Therefore, by utilising these two-fold benefits from increased $SnCl_2.2H_2O$ concentrations, increased tin

reduction and decreased electrolyte reduction, it is possible to promote beneficial cathode current efficiencies during deposition for a larger range of current densities.

4.4.1.2 *Anodic reactions*

Although cathodic reactions are paramount, anode electrochemistry must be considered. Issues of electrolyte anodic oxidation pertain to both voltammetry and electrodeposition. Electrolyte anodic oxidation lessens the anodic current efficiency creating soluble, insoluble or gaseous oxidation products at the anode possibly leading to electrolyte contamination. Previously choline chloride decomposition by electrochemical oxidation has been studied [31]. This report suggested the formation of possibly toxic oxidation products at the anode, reducing the environmental favourability for electrolyte disposal.

Evidence of electrolyte contamination was studied using Propeline in an 'H' cell, separating any electrode reaction products. The electrolyte (containing no tin) appeared clear prior to and following electrolysis (1.5 V vs Ag/AgCl for 60 mins), but on the addition of $SnCl_2.2H_2O$, at 0.088 M, resulted in a colour change for the anodically polarised electrolyte, illustrated by Figure 4.57 and colour change seen in Figure 4.58. This colour change was only experienced after electrolysis and must have been caused by the interaction between $SnCl_2.2H_2O$ and an unknown oxidation product. This test was additionally undertaken polarising Propeline containing additions of $SnCl_2.2H_2O$ at 0.088 M but resulted in no colour change, whereas at lower concentrations (<0.088 M) exhibited small discolouration.

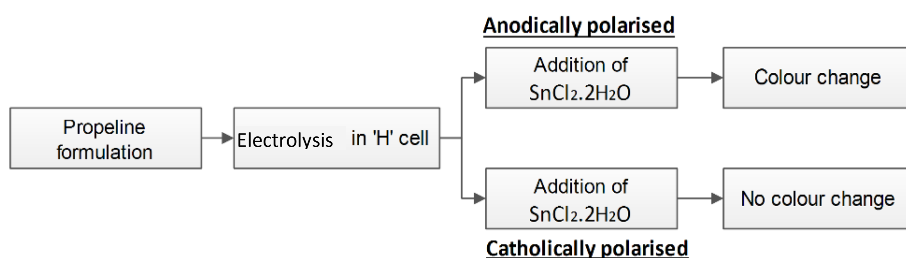


Figure 4.57: Flow diagram of Propeline polarisation and resulting colour change

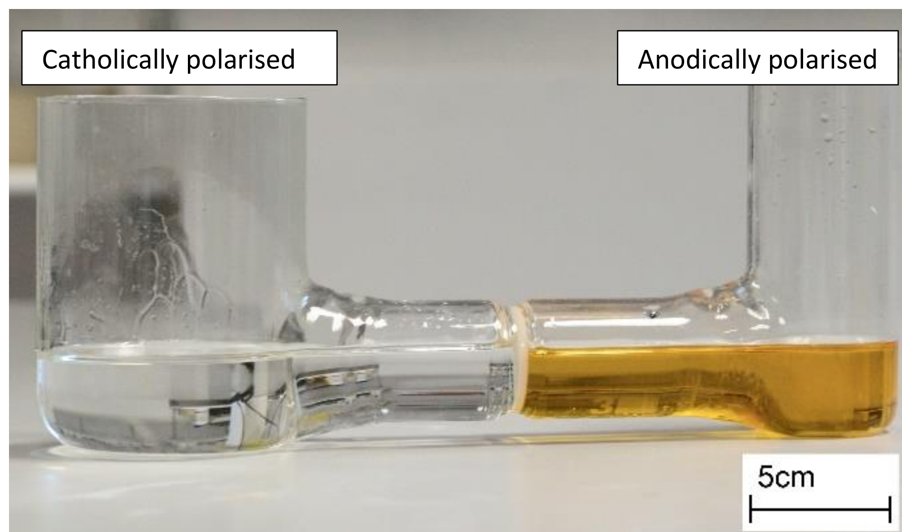


Figure 4.58: Electrochemical ‘H’ cell showing colour variation due to oxidation product contamination on the addition of $SnCl_2.2H_2O$ for Propeline after electrolysis at 1.5 V vs Ag/AgCl for 60 mins

Although there had been a colour change, it is unclear from appearance whether this indicated an alteration in electrochemical behaviour. The resulting voltammetry of the coloured and clear liquid is seen in Figure 4.59.

The two solutions produced after electrolysis, as seen in Figure 4.58, were then electrochemically characterised using cyclic voltammetry. The potential peak shifts, with respect to voltage and current, between voltammetry for the clear and coloured Propeline in Figure 4.59 indicate that the oxidation products causing a colour change also affect the electrolyte electrochemistry. Furthermore, from the repeated cycles in the coloured Propeline the polarisation during cyclic voltammetry is itself altering the electrolyte, whereas this is not seen for the clear Propeline with stable voltammetry. This changing voltammetry suggests that after electrolyte oxidation products have formed they encourage further electrochemical alterations during additional polarisation, whereas the constant voltammetry for the clear propeline indicates that the additions of $SnCl_2.2H_2O$ prior to polarisation inhibits the production of electrochemically active oxidation products. Therefore, since electrochemical changes are evident, altering cathodic reactions, thought must be given to anodic reactions as they result in electrolyte contamination which may lead to possible changes in an electrodeposit.

Due to the sizeable potential window between the onset of tin oxidation and the anodic degradation of the electrolyte of 3 V, and oxidation occurring ohmically until tin is fully removed, a tin anode should be able to sustain high anodic current efficiencies even at high current densities. Therefore, the benefits of using a tin anode would be twofold: high anodic current efficiencies

reducing the production of oxidation products other than that of Sn^{2+}; and $SnCl_2$ concentrations would be sustained utilising previously seen benefits associated with higher concentrations.

Anodic tin passivation has been previously documented [261] but because of the visible removal of tin occurring during anodic periods on the working electrode it is not thought to be taking place during oxidation in this system. Additionally, tin oxidation has been known to form Sn^{4+} during anodic oxidation [122], possibly explaining the small shoulders in the voltammetry in Figure 4.59 at +1.5 V. Although the presence of Sn^{4+} may be detrimental in prolonged electrodeposition, it was not seen to be such within these experiments due to the stability of the cyclic voltammetry in Figure 4.59 for clear propeline.

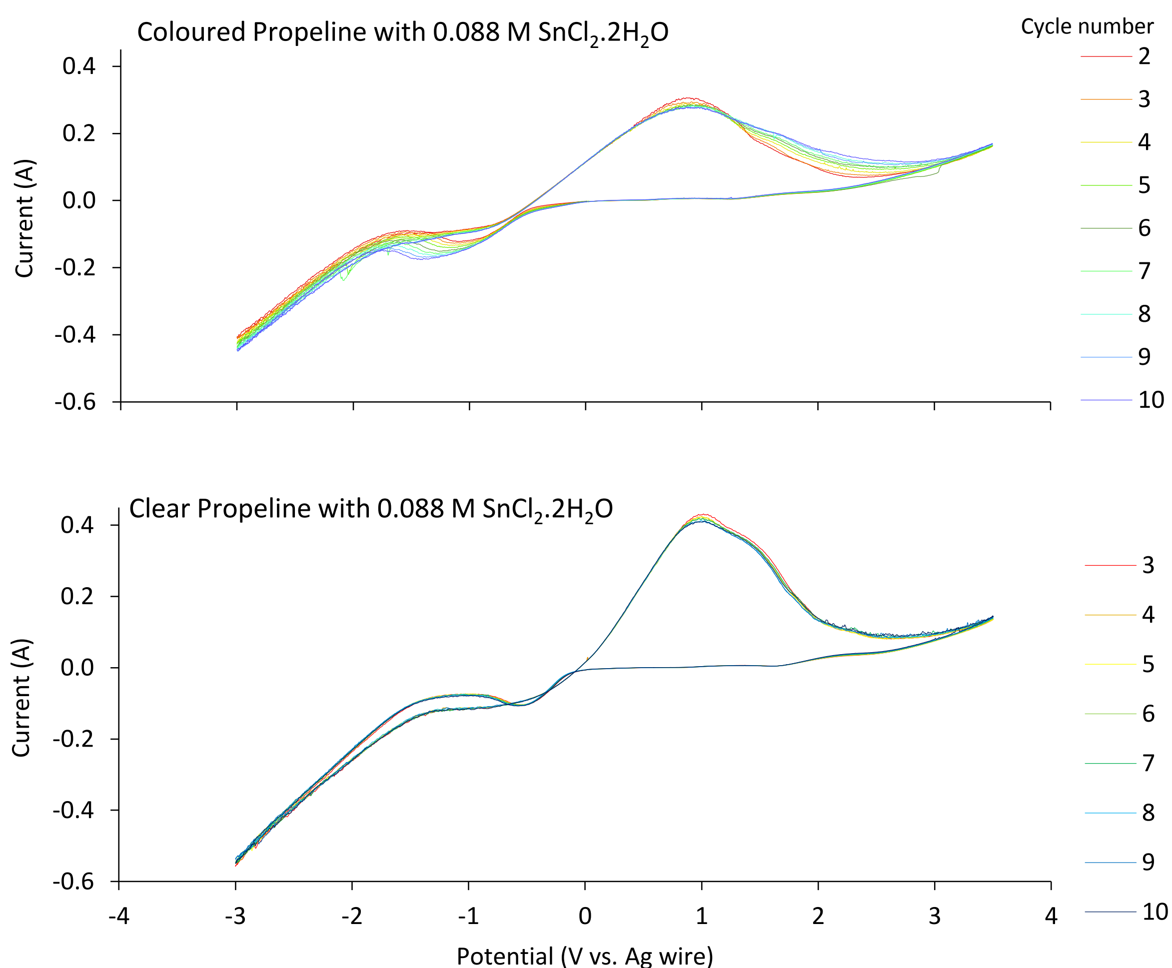


Figure 4.59: Potential-current voltammetry between -3 V and +3.5 V for repeated scan cycles at a scan rate of 50 mV/s for the coloured and virgin Propeline with 0.088 M $SnCl_2.2H_2O$ at 70^0C

4.4.1.3 *Cathode current efficiency*

Many of the remarks concerning electrolyte voltammetry, such as beneficial effects due to increased $SnCl_2.2H_2O$ concentrations and low viscosity electrolytes, were further substantiated

during measurements of cathodic current efficiency during electrodeposition. Although fragile dendritic loss and electrolyte gain between dendrites produced errors in mass gain measurements, cathode current efficiency graphs with respect to: electrolyte, $SnCl_2.2H_2O$ concentration and current density are shown in Figure 4.60.

Comparing the cathodic efficiencies in Figure 4.60, trends suggest reduced efficiencies by measuring deposit weight before and after deposition for Reline compared to Propeline and Ethaline; and decreasing $SnCl_2.2H_2O$ concentrations. These lowered efficiencies are attributed to poor mass transport, related to Reline's high viscosity, and decreased benefits from $SnCl_2.2H_2O$ additions. This is exemplified by cathodic efficiencies for Propeline, showing a clear correlation of $SnCl_2.2H_2O$ concentration which express high efficiencies even at high current densities for 0.71 M $SnCl_2.2H_2O$. Additionally, trends within Figure 4.60 provide an assessment of the accuracy for these single results, suggesting sufficiently low error, allowing these measurements to be considered reliable.

Additionally, it has been possible to calculate the resulting deposition rates caused by the applied current densities in Figure 4.61. Comparatively these results are close to those stated for the aqueous Tinmac electrolyte which exhibits a deposition rate of approximately 1 µm/min [257]. These results show many of the deposition processes to exhibit non-linear relationships between deposition rate and applied current density. These limiting effects are seen clearly in the Propeline with additions of 0.044 M $SnCl_2.2H_2O$, which, even though current density is increased, the resulting deposition rate is limited below 4 mA/cm^2. This effect is additionally seen for other low $SnCl_2.2H_2O$ concentration electrolytes. Therefore, this clearly shows that deposition from these electrolytes is diffusion limited at low $SnCl_2.2H_2O$ concentrations. On the other hand, high $SnCl_2.2H_2O$ concentrations, within these current density boundaries, there are linear deposition rates with respect to current density. Thus at high $SnCl_2.2H_2O$ concentrations the deposition rate is not limited by diffusion. Therefore it might be assumed that these high $SnCl_2.2H_2O$ concentration electrolytes may be able to attain even larger deposition rates than those seen in this study by utilising larger current densities.

Furthermore, the deposition rates shown for these high $SnCl_2.2H_2O$ concentration electrolytes, even for Reline, are comparable to the alternative aqueous Tinmac electrolyte used within this project [257]. Therefore, these results show that: although the electrolytes examined within this

study exhibit high viscosities, which commonly result in diffusion limited low deposition rates [108]–[110], by elevating $SnCl_2.2H_2O$ concentration it is possible to reduce this limitation.

The voltammetry of Reline, Ethaline and Propeline indicated large potential windows, of 3.5, 3.5, and 3.1 V respectively, compared to that of aqueous electrolytes of 1.23 [67]. Furthermore, these electrolytes produced significant potential windows between tin and electrolyte reduction, and beneficial effects from increased $SnCl_2.2H_2O$ concentration such as increasing deposition rate at a given potential and reduced electrolyte degradation. As a result of these large potential windows, these electrolytes become increasingly appealing to the electroplater as they are able to deposit novel metals with increased efficiency and deposition rate.

The increased deposition rate and reduced electrolyte degradation due to increased $SnCl_2.2H_2O$ concentration is thought to be caused by the increased current density limit at the working electrode. By producing an increased concentration of tin within the electrolyte bulk it is possible to increase concentration gradients between itself and the diffusion layer from which Sn^{+2} is reduced. These increased concentration gradients thus result in increased limiting current density, as limiting current density is proportional to reducing species concentration [33], and thus results in higher cathode current efficiencies prior to reaching this limit.

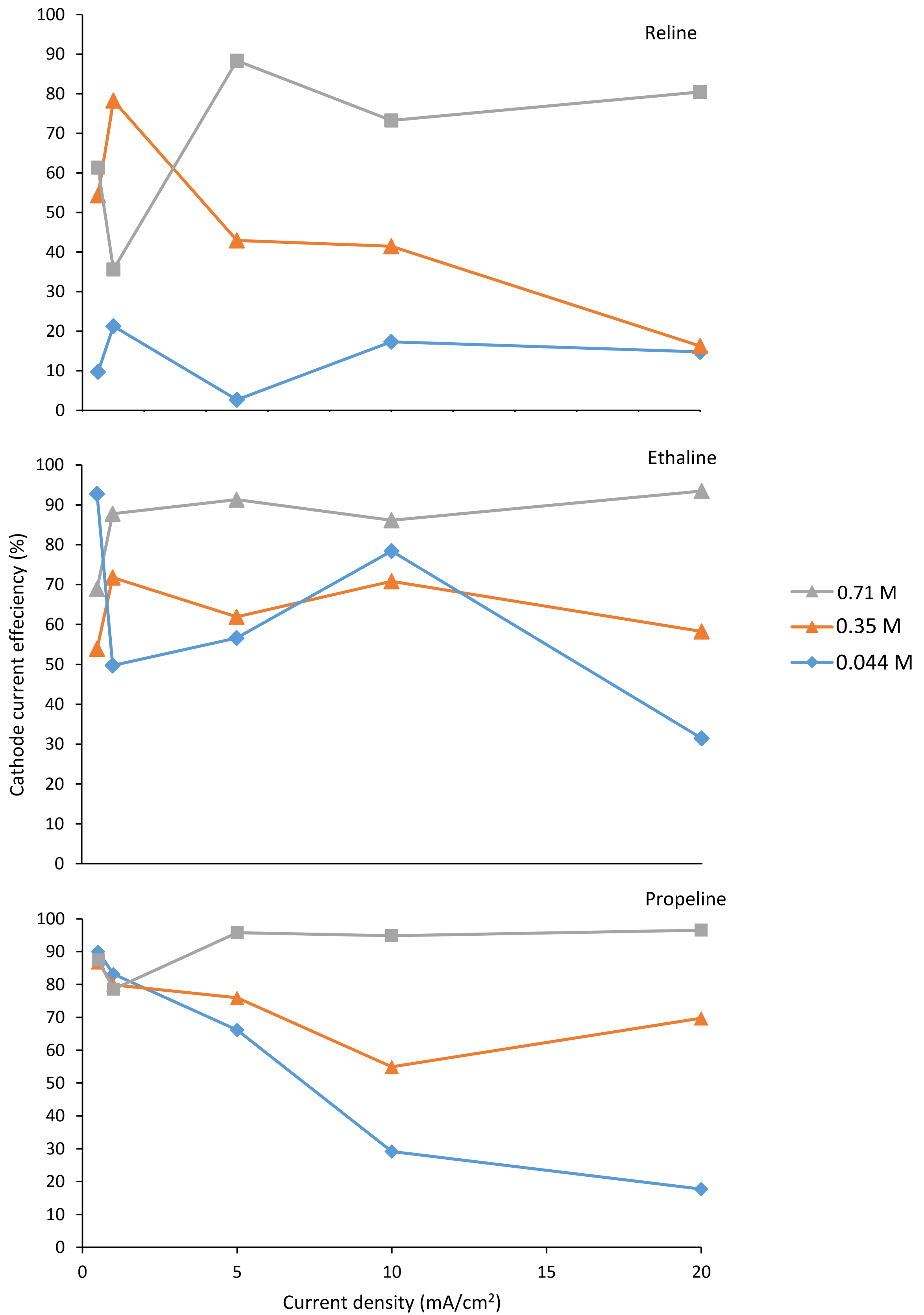


Figure 4.60: Cathode current efficiency- current density plots for tin electrodeposits produced from Reline, Ethaline and Propeline with additions of $SnCl_2.2H_2O$ at 0.044 M, 0.35 M and 0.71 M at 0.5, 1, 2, 5, 10 and 20 mA/cm^2. Point markers representing experimental results

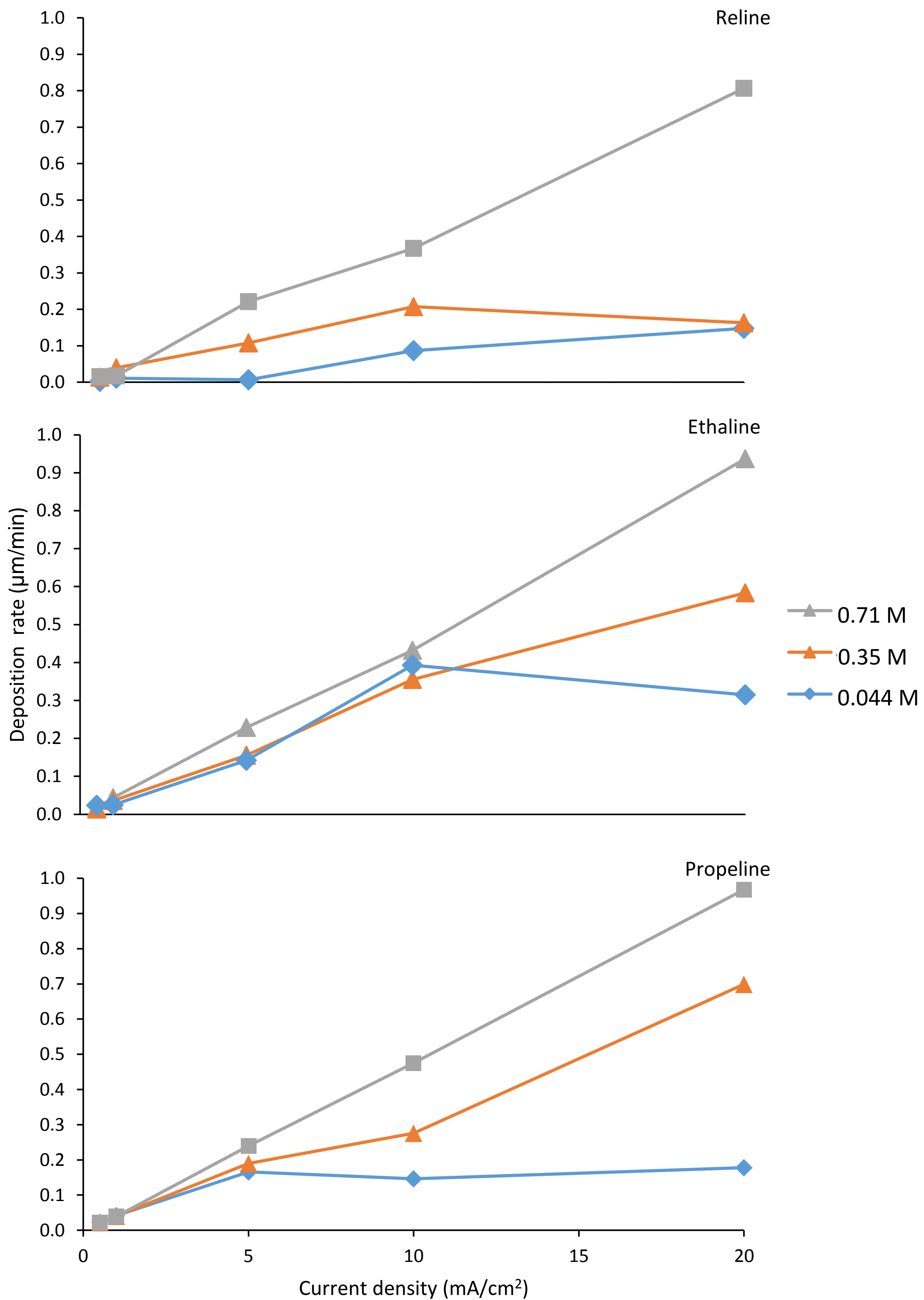


Figure 4.61: Deposition rate- current density graph for tin electrodeposits produced from Reline, Ethaline and Propeline with additions of $SnCl_2.2H_2O$ at 0.044 M, 0.35 M and 0.71 M at 0.5, 1, 2, 5, 10 and 20 mA/cm^2. Point markers representing experimental results

4.4.2 *Non-aqueous electrolytes: optimisation of deposition parameters*

The evaluation of electrodeposition using DESs was separated into two stages. Firstly, electrodeposit quality was assessed using a fixed charge passed (3 C/cm^2 ~2.5 µm at 100 % cathode current efficiency) with respect to variations in current density, electrolyte type and $SnCl_2.2H_2O$ concentration. From this study parameters which resulted in good quality electrodeposits were identified. Secondly, these parameters were subsequently used to investigate the effect of increasing deposition time, .i.e. increased deposition thickness and its effects on coating quality.

4.4.2.1 *Thin electrodeposit quality evaluation*

Evaluation of the coatings was carried out at both the edge and centre regions of the coupons (2x2 cm copper substrates). These locations, the edges and centres, often experience high levels of dendritic growth at the edges, whilst the centres experience less, allowing other deposit characteristics to be assessed. Similarities in topography between deposits were noted and were separated into nine categories (A-J) for edge formation, deposits topographies seen at the edges of coupons, with examples shown in Figure 4.62.

Figures 4.62.B, F, H and I, express traditional dendritic structures with a fern-like form, whereas other micrographs in Figure 4.62, excluding E, contain features which appear to be more unusual. Figures 4.62.A and C feature many unusual dendritic growths exhibiting lengths of several mm, widths of ~10 µm and, due to large aspect ratios, appear similar to whiskers unless viewed at high magnifications. Moreover, on closer inspection, as seen in Figures 4.63 and 4.64, these features are frequently hollow or flat, and exhibit rough surfaces that are absent of whisker-like striations along their circumference. It is these finer details, seen at high magnifications, which allow differentiation between these growths and whiskers. It is thought that due to their large lengths of these dendritic growths penetrate the electrolyte diffusion layer; further increasing the non-uniform growth rate of the deposit.

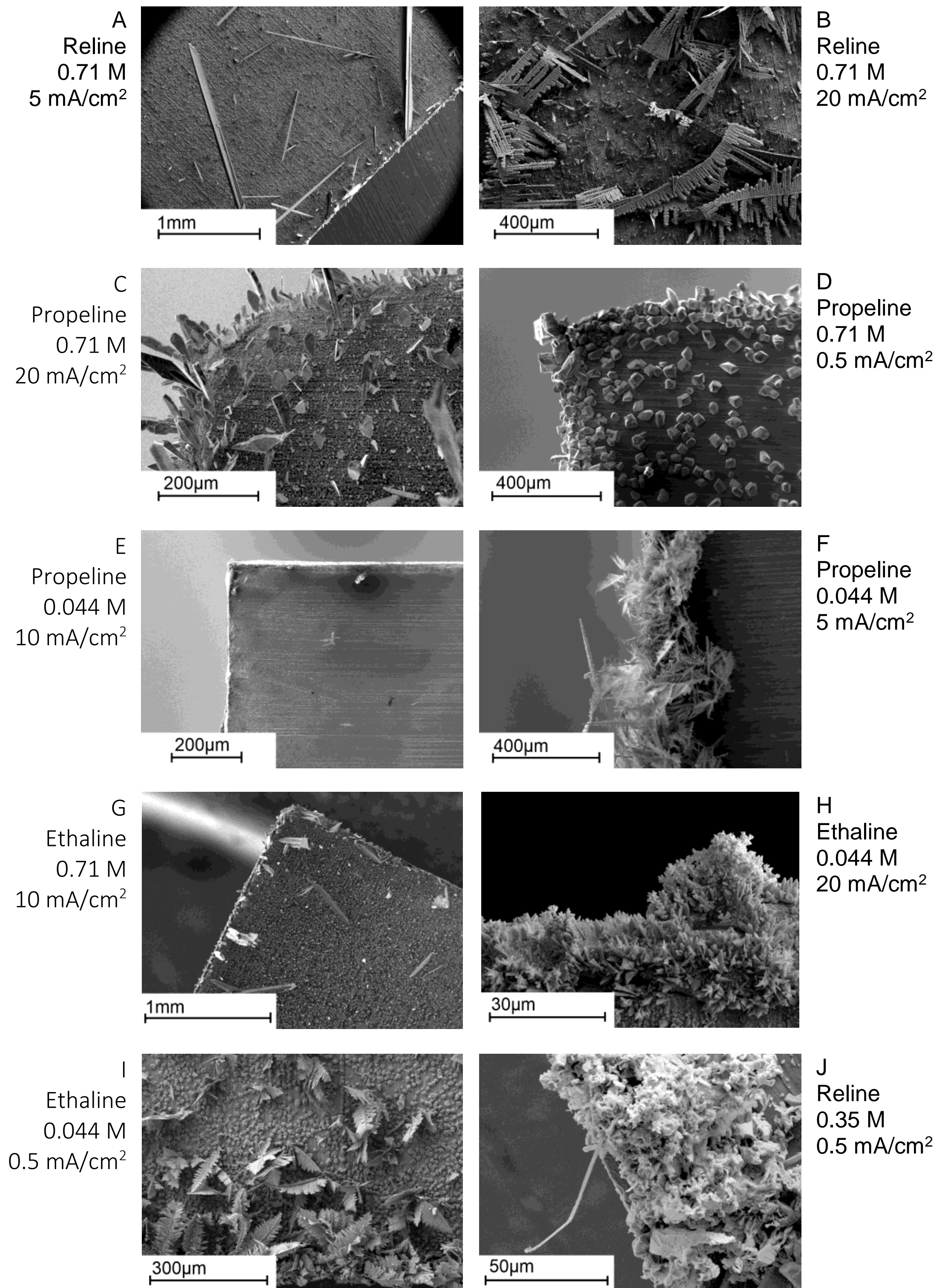


Figure 4.62: SEM micrographs showing tin electrodeposits at sample edges produced from Reline, Ethaline and Propeline with various $SnCl_2.2H_2O$ concentrations at several current densities

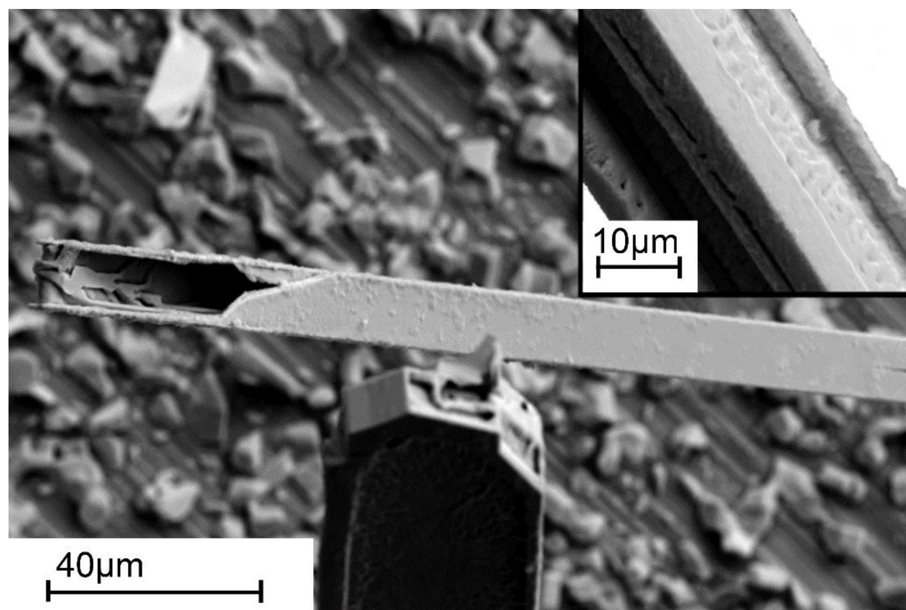


Figure 4.63: SEM micrographs of large hollow tin features electrodeposited at the sample edge areas from Reline containing 0.71 M of $SnCl_2.2H_2O$ at 5 mA/cm^2. Insert: surface of feature at high magnification

Although the majority of electrodeposits in Figure 4.62 formed non-uniform features; greatly reducing deposit quality, a few displayed uniform edge regions as shown in Figure 4.62.E; the smoothest edge regions seen in this study.

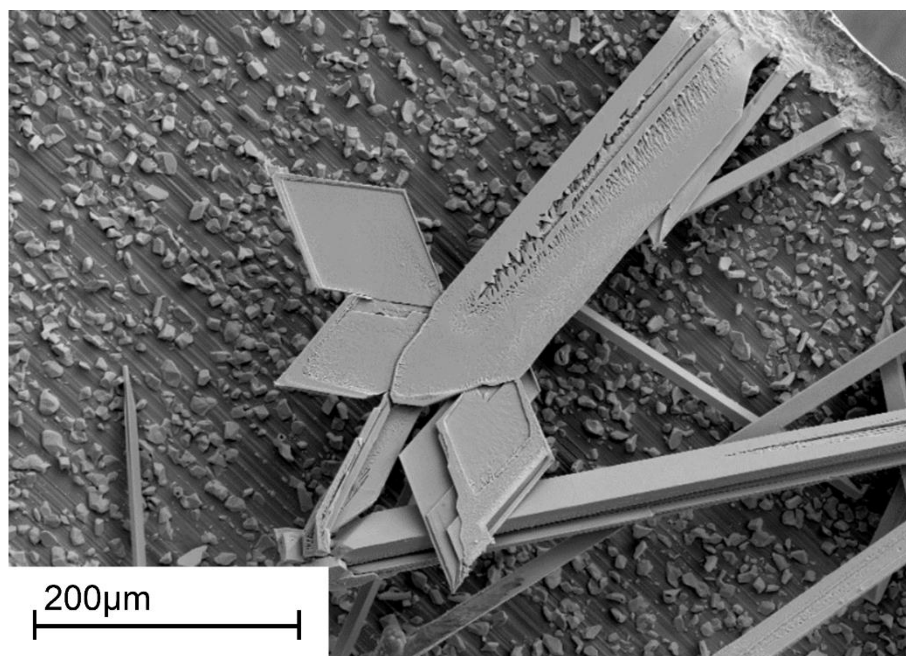


Figure 4.64: SEM micrographs of large flat features electrodeposited at the sample edge from Reline containing 0.71 M of $SnCl_2.2H_2O$ at 5 mA/cm^2

In addition, central regions were classified with example structures shown in Figure 4.65, categorised into groups: uniform (Figures 4.65.A, B, C and D), patchy (Figures 4.65.E and F),

containing large features (Figures 4.65.G, H, I, J and K) and localised particles (Figures 4.65.L, M, N). Furthermore, to summarise the large amount of sample evaluation, Table 4.4 has been produced classing the features dependent upon the similarities to previous Figures for edge and central regions.

By the removal of Propeline with 0.044 M $SnCl_2.2H_2O$ at 20 mA/cm^2 due to its resemblance to Propeline with 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 this evaluation produced three sets of variables deemed to be promising and worthy of further investigation. These variables were: Ethaline, 0.044 M $SnCl_2.2H_2O$, 10 mA/cm^2, selected due to its high uniformity; Propeline, 0.044 M $SnCl_2.2H_2O$, 10 mA/cm^2, also selected due to its high uniformity; Propeline, 0.71 M $SnCl_2.2H_2O$, 0.5 mA/cm^2 selected due to its high uniformity and bright finish.

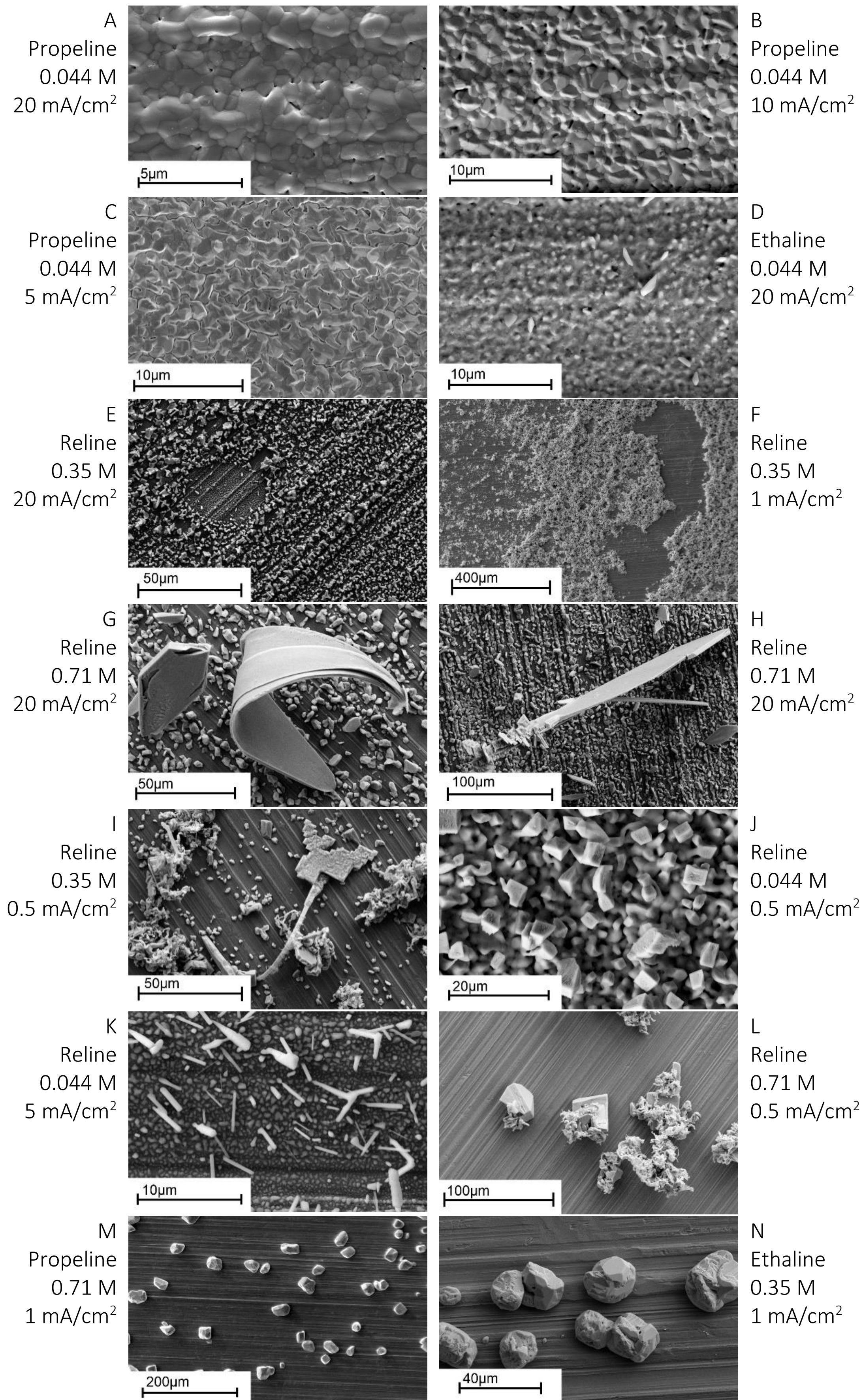


Figure 4.65: SEM micrographs showing central regions of tin electrodeposits from Reline, Ethaline and Propeline with various $SnCl_2.2H_2O$ concentrations at several current densities

Table 4.4: Evaluation of electrodeposit quality with respect to deep eutectic solvent, $SnCl_2.2H_2O$ concentration and current density. Green – good, yellow – improved, but still unacceptable, red- poor.

DES	$SnCl_2$ Conc. (M)	Current density (mA/cm^2)	Figure similarity: Edge region	Figure similarity: Central region	Visual feature	Overall quality
Reline	0.044	0.5	Figure 4.62.G	Figure 4.65.J	Matte	Poor
		1	Figure 4.62.G	Figure 4.65.J	Matte	Poor
		5	Figure 4.62.G	Figure 4.65.K	Matte	Poor
		10	Figure 4.62.E	Figure 4.65.K	Matte	Poor
		20	Figure 4.62.E	Figure 4.65.L	Matte	Poor
	0.35	0.5	Figure 4.62.J	Figure 4.65.I	Matte	Poor
		1	Figure 4.62.E	Figure 4.65.F	Matte, patchy	Poor
		5	Figure 4.62.E	Figure 4.65.F	Matte, patchy	Poor
		10	Figure 4.62.J	Figure 4.65.H	Matte, dendritic	Poor
		20	Figure 4.62.H	Figure 4.65.E	Matte	Poor
	0.71	0.5	Figure 4.62.D	Figure 4.65.L	Matte	Poor
		1	Figure 4.62.E	Figure 4.65.M	Matte, patchy	Poor
		5	Figure 4.62.A	Figure 4.65.M	Matte	Poor
		10	Figure 4.62.E	Figure 4.65.H	Matte	Poor
		20	Figure 4.62.B	Figure 4.65.G	Matte	Poor
Ethaline	0.044	0.5	Figure 4.62.I	Figure 4.65.N	Matte	Poor
		1	Figure 4.62.E	Figure 4.65.N	Satin, bright specks	Poor
		5	Figure 4.62.I	Figure 4.65.H	Satin	Poor
		10	Figure 4.62.H	Figure 4.65.D	Matte	Med.
		20	Figure 4.62.H	Figure 4.65.D	Matte	Poor
	0.35	0.5	Figure 4.62.A	Figure 4.65.N	Satin, bright specks	Poor
		1	Figure 4.62.D	Figure 4.65.N	Satin, bright specks	Poor
		5	Figure 4.62.D	Figure 4.65.M	Satin, bright dendrites	Poor
		10	Figure 4.62.A	Figure 4.65.H	Satin, bright dendrites	Poor
		20	Figure 4.62.C	Figure 4.65.M	Satin, bright dendrites	Poor
	0.71	0.5	Figure 4.62.D	Figure 4.65.M	Satin, bright specks	Med.
		1	Figure 4.62.D	Figure 4.65.M	Thin, bright specks	Poor
		5	Figure 4.62.A	Figure 4.65.M	Thin, bright specks	Poor
		10	Figure 4.62.A	Figure 4.65.G	Thin, bright specks	Poor
		20	Figure 4.62.C	Figure 4.65.H	Thin, bright specks	Poor
Propeline	0.044	0.5	Figure 4.62.G	Figure 4.65.C	Satin, bright specks	Poor
		1	Figure 4.62.C	Figure 4.65.C	Satin, bright dendrites	Poor
		5	Figure 4.62.H	Figure 4.65.C	Matte, dark dendrites	Poor
		10	Figure 4.62.E	Figure 4.65.C	Matte	Good
		20	Figure 4.62.E	Figure 4.65.A	Matte	Good
	0.35	0.5	Figure 4.62.G	Figure 4.65.H	Satin, bright dendrites	Poor
		1	Figure 4.62.G	Figure 4.65.H	Satin, bright dendrites	Poor
		5	Figure 4.62.C	Figure 4.65.H	Satin, bright dendrites	Poor
		10	Figure 4.62.C	Figure 4.65.H	Satin, bright dendrites	Poor
		20	Figure 4.62.C	Figure 4.65.H	Satin, bright dendrites	Poor
	0.71	0.5	Figure 4.62.D	Figure 4.65.M	Bright, bright specks	Med.
		1	Figure 4.62.G	Figure 4.65.M	Satin, bright specks	Poor
		5	Figure 4.62.C	Figure 4.65.M	Satin, bright dendrites	Poor
		10	Figure 4.62.A	Figure 4.65.H	Satin, bright dendrites	Poor
		20	Figure 4.62.A	Figure 4.65.H	Matte	Poor

4.4.2.2 *Preparation of electrodeposits with increased thickness*

Ethaline, 0.044 M $SnCl_2.2H_2O$, 10 mA/cm^2.

Uniform deposits produced using Ethaline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 were difficult to reproduce without dendritic growth, with an example shown in Figure 4.66. Furthermore, during electrodeposition, samples shows unusual intriguing potential-time graphs, as shown in Figure 4.67.

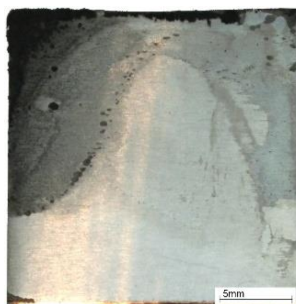


Figure 4.66: micrograph of electrodeposit created from Ethaline with 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 showing darkened areas due to dendritic growth

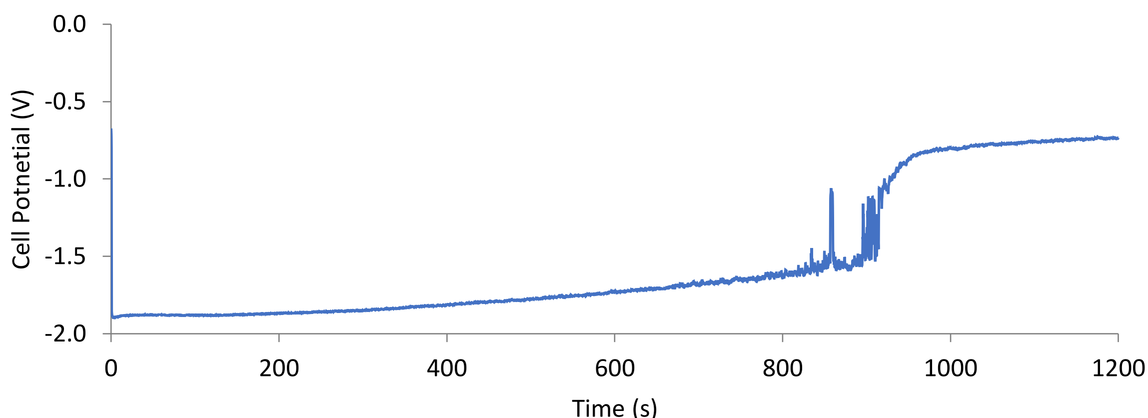


Figure 4.67: Potential-time curve for the galvanostatic electrodeposition of a deposit produced from Ethaline with 0.044 M of $SnCl_2.2H_2O$ at 10 mA/cm^2 showing a reduction in cell potential

The sudden and large ~1 V potential increase shown in Figure 4.67 at 900 s is thought to coincide with the formation of dendritic growth on the sample surface. This may to be due to, reduction taking place primarily upon uniform areas prior to the increase, whereas following it reduction is occurring onto rapidly growing dendritic protrusions occurring at a large number of sites across the surface of the deposit. It is thought that deposition occurring solely at dendritic tips would result in a reduction of potential due to rapid growth into the bulk electrolyte experiencing only a minor diffusion layer.

Additionally, these dendritic growths are thought to be the form the deposit prefers to grow above a specific value of thickness. As such, these dendritic growths initiate due to the deposit forming into its preferred form, these protrusions then penetrate the electrolyte diffusion layer thus providing a low potential pathway for growth. Therefore, as such a large number of sites across the whole of the deposit surface experience this effect the cell potential increases due to an increased ease of galvanostatic deposition. This is much the same as traditional dendritic growth theory, but in this case their initiation is encouraged by a preferred form of deposit topography.

Propeline, 0.71 M $SnCl_2.2H_2O$, 10 mA/cm^2

Deposits prepared using Propeline with 0.71 M $SnCl_2.2H_2O$ at 0.5 mA/cm^2 produced a bright surface deposit, as seen in Figure 4.68, at high cathodic efficiencies (~85%). By SEM analysis these deposits exhibited a uniform layer of tin and sporadic large faceted particles, as seen in Figure 4.69, observed visually as highly reflective specks on a bright deposit.

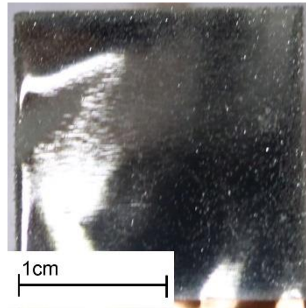


Figure 4.68: Photograph of electrodeposit created from Propeline with 0.71 M $SnCl_2.2H_2O$ at 0.5 mA/cm^2

EBSD analysis of these large particles revealed that they were polycrystalline suggesting their growth was caused by multiple nucleation sites. Furthermore, mechanical cross-sectioning of these deposits, seen in Figure 4.70, revealed that the majority of tin deposition taking place was in the form of large particles rather than uniformly. Moreover, by FIB cross-sectioning it was possible to discern a ~20 nm thick tin coating between the large particles.

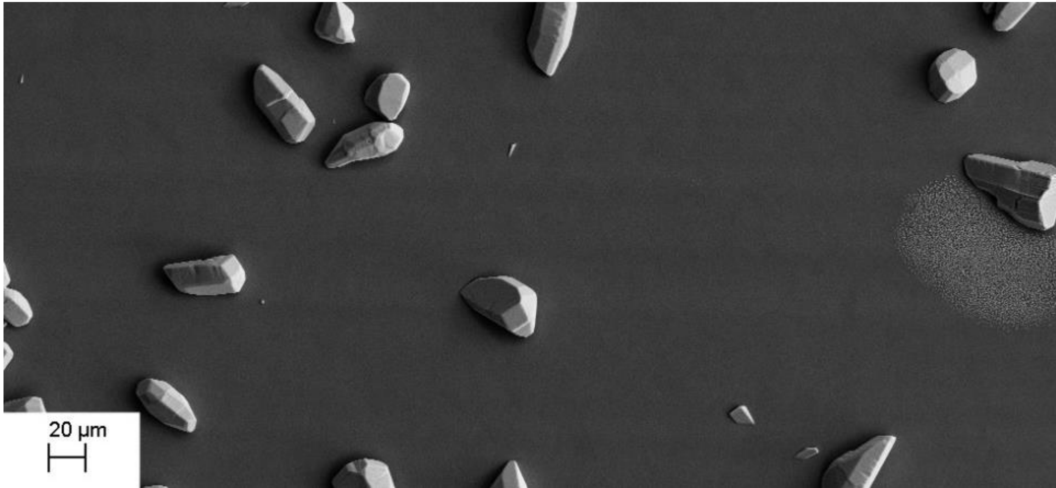


Figure 4.69: SEM micrograph of electrodeposit produced from Propeline containing 0.71 M $SnCl_2.2H_2O$ at 0.5 mA/cm^2

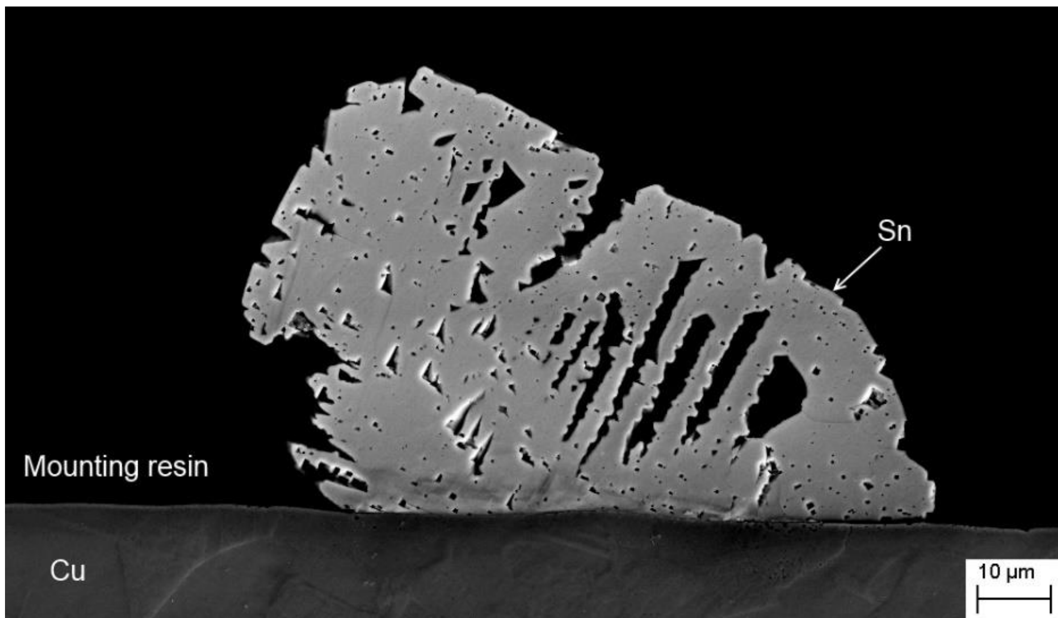


Figure 4.70: SEM micrograph of cross-sectioned electrodeposit produced from Propeline containing 0.71 M $SnCl_2.2H_2O$ at 0.5 mA/cm^2 showing a large tin particle

Although these parameters create a bright surface finish with high cathode current efficiencies, the thickness of the uniform deposit is too thin for this study. Therefore, although this deposit may be useful in another area of industry, such as nanowire coating [262] and RF transparent coatings [263], this deposit will not be studied further here.

Propeline with 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2

Deposits prepared using Propeline with 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 at 15% cathode current efficiency produced changes in visual and topographical appearance with increasing thickness, as shown in Figure 4.71.

Additionally, at 18 C/cm^2 darkening around coupon edges can be observed. Using SEM analysis these darkened areas were shown to contain a high density of ring formations, with an example in Figure 4.72. The formation of these rings is attributed to electrolyte reduction at their centres creating gas evolution, thought to be hydrogen bubbles [31]. It is this gas evolution which then increases localised agitation, decreasing polarisation and accelerating localised deposition rates. Moreover, additional dendrite-like features were seen to grow from uniform areas as shown in Figure 4.73.

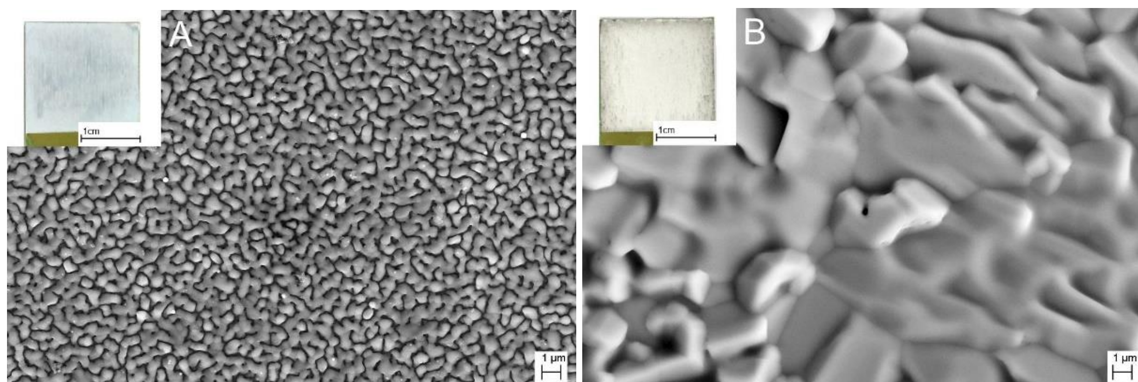


Figure 4.71: SEM micrographs and corresponding photographs of electrodeposits produced from Propeline containing 0.044 M $SnCl_2.2H_2O$ at 10mA/cm^2. A: produced after passing 1.2 C/cm^2 and B: produced after passing 18 C/cm^2

Furthermore, a deposit produced by passing 18 C/cm^2 was cross-sectioned using FIB to attain deposit thickness measurements ranging from 1 to 2 µm along its surface, as seen in Figure 4.74.

Therefore, as the presence of dendrite-like growth increased with deposit thickness it was apparent that deposits below 2 µm featured sufficient uniform deposition; providing a maximum practical thickness for these deposits of 2 µm.

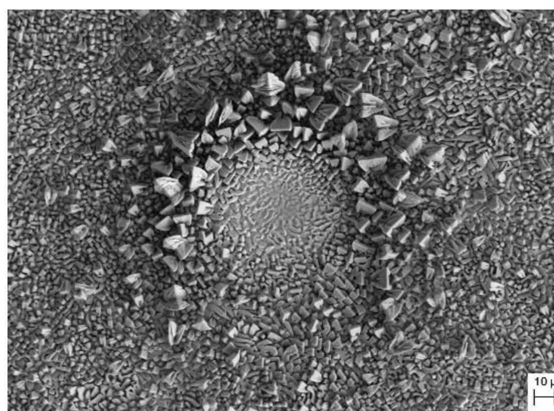


Figure 4.72: SEM micrograph of electrodeposit produced from Propeline containing 0.044 M $SnCl_2.2H_2O$ at 10mA/cm^2 after passing 18 C/cm^2 showing a ring formation

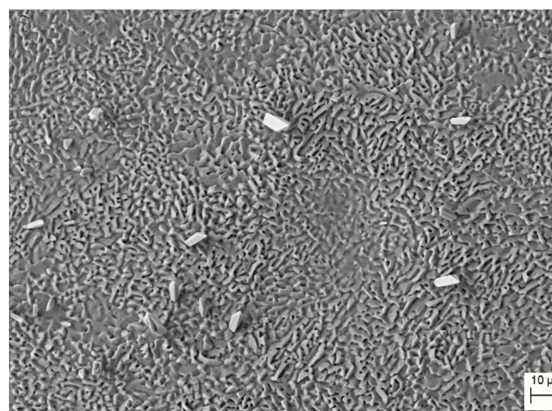


Figure 4.73: SEM micrograph of electrodeposit produced from Propeline containing 0.044 M $SnCl_2.2H_2O$ at 10mA/cm^2 after passing 12 C/cm^2

Figure 4.74: SEM micrograph of a FIB cross-sectioned electrodeposit produced from Propeline with 0.044 M $SnCl_2.2H_2O$ at 10mA/cm^2 after passing 18 C/cm^2

Many of the deposits formed in this study produced large dendrite-like structures. It was primarily due to the formation of these large protruding features that the vast majority of deposits were not studied in more detail. Although this was the case, a few deposits at low thicknesses were uniform. Moreover, one deposit even appeared bright in appearance, a feature commonly only achieved with additional additive electrolyte content.

Additionally, many of the electroplating variables investigated in this study produced coatings which, although uniform at low thicknesses, became increasingly prone to the formation of large dendritic-type structures. It has been shown that for thicknesses, of ~ 1 µm, the deposits are undergoing uniform growth, whereas at thicknesses larger than ~ 2 µm tin no longer deposits uniformly. Therefore, one means of increasing the uniformity of these deposits is to maintain coating thicknesses below ~ 2 µm; which is also known to encourage whisker growth [50], [228].

4.5 Comparison of electrodeposits from aqueous and non-aqueous electrolytes

To form 1 and 2 µm thick deposits produced from Propeline (with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2) and Tinmac (proprietary aqueous electrolyte), each electrodeposition method required characterisation to define deposition times. Therefore, a range of deposits produced throughout this study for various experiments (using standard experimental procedures with a range of deposition times) were characterised with respect to mass gain (independent to uniformity), and time of deposition, these results are shown in **Error! Reference source not found.**.

The Tinmac and Propeline processes deposit at cathode current efficiencies of 98 % and 15 % respectively; which represents a significant difference. Therefore, deposition times for 1 and 2 µm thick deposits for each bath can be calculated by using the deposition rate of 1 µm/min for Tinmac and 0.081 µm/min for Propeline.

Visually, the electrodeposits produced from these 1 and 2 µm coatings exhibited matte and satin surface finishes for the Propeline and Tinmac respectively, as seen in Figure 4.75.

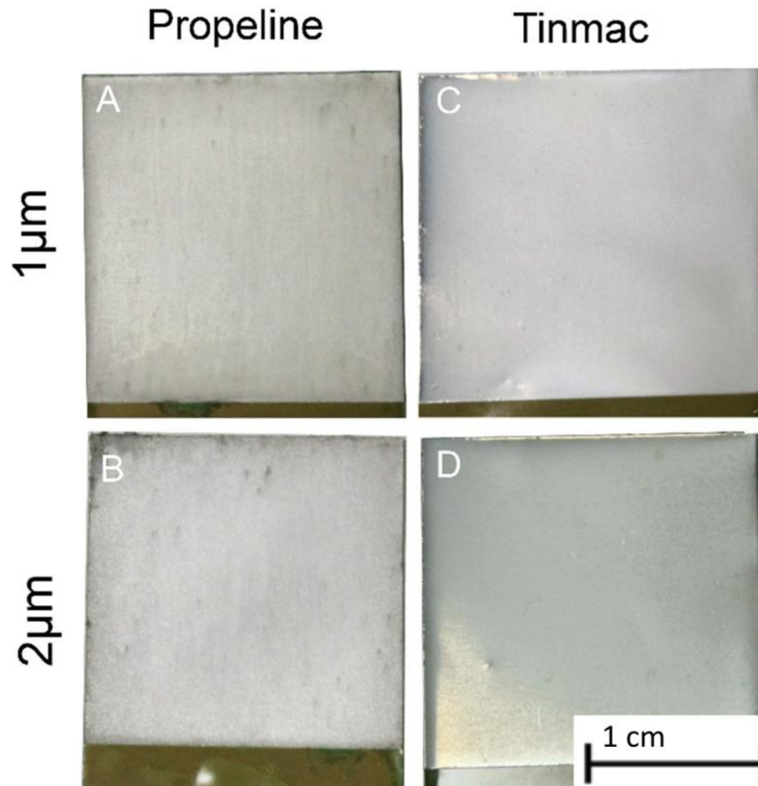


Figure 4.75: Micrograph of tin electrodeposits produced from Propeline (A and B) with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 and Tinmac proprietary aqueous bright tin bath (C and D) at 20 mA/cm^2 for thicknesses of 1 µm (A and C) and 2 µm (B and D)

To confirm the predicted thicknesses the electrodeposits were cross-sectioned using FIB and mechanical methods. The resulting thickness cross-sections are shown in Figure 4.76.

Although an approximate doubling of thickness has occurred for the Tinmac deposits, the thickness increase seen for the deposits produced from the Propeline electrolyte is less than double. This would suggest that, compared with the 1 µm thick deposit, the 2 µm thick Propeline deposit is being deposited less uniformly.

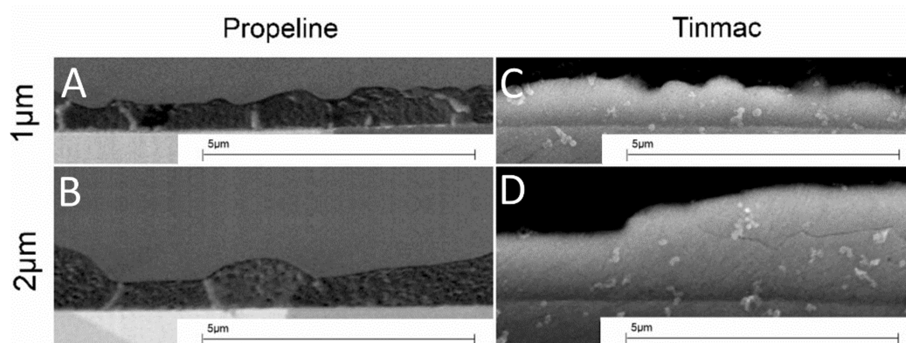


Figure 4.76: SEM micrograph of FIB (A and B) and mechanically (C and D) cross-sectioned tin electrodeposits from Figure 4.75

Additionally, Figure 4.76 shows the deposits produced from Propeline exhibiting a grain size of approximately 1-2 µm in diameter. This value of grain size is well within expected values [78]. Moreover, this grain size is far smaller than the substrate grain size, as seen in Figure 3.2.

4.5.1 *Electrodeposit topography comparison*

The electrodeposits were also analysed by SEM with respect to their surface topography as seen in Figures 4.77 and 4.78. It was seen that roughness variations of the tin surfaces were of comparable scale to those of the underlying copper substrate grain structure in Figure 3.2. In addition, deposits produced from Propeline exhibited increased dendritic growth with increasing thickness.

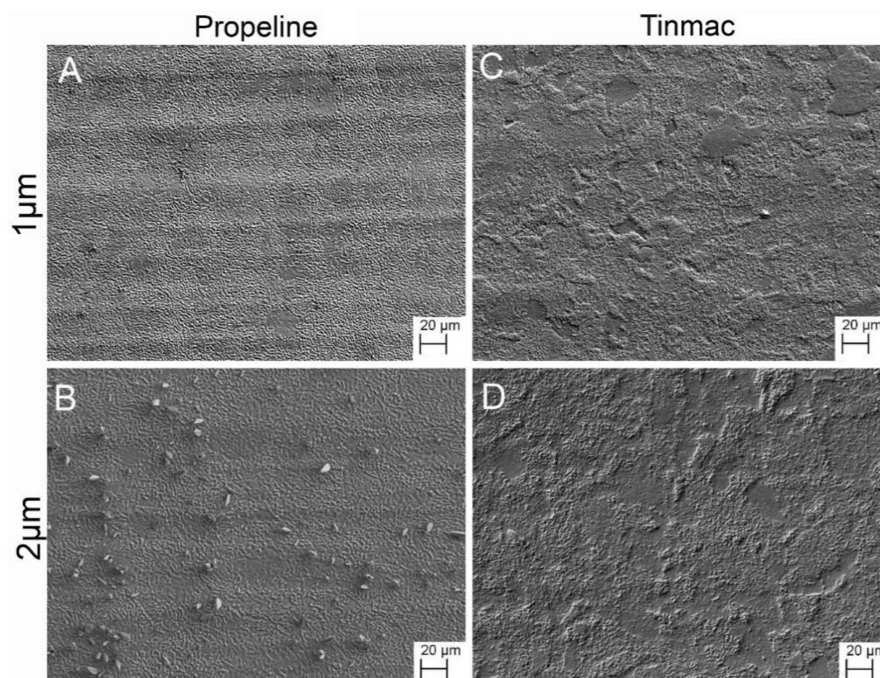


Figure 4.77: SEM micrographs of tin electrodeposits from Figure 4.75 at low magnification

At higher magnifications, as seen in Figure 4.78, deposits produced from the aqueous electrolyte showed a reduction in surface roughness with increased deposition thickness, resulting in the satin finish.

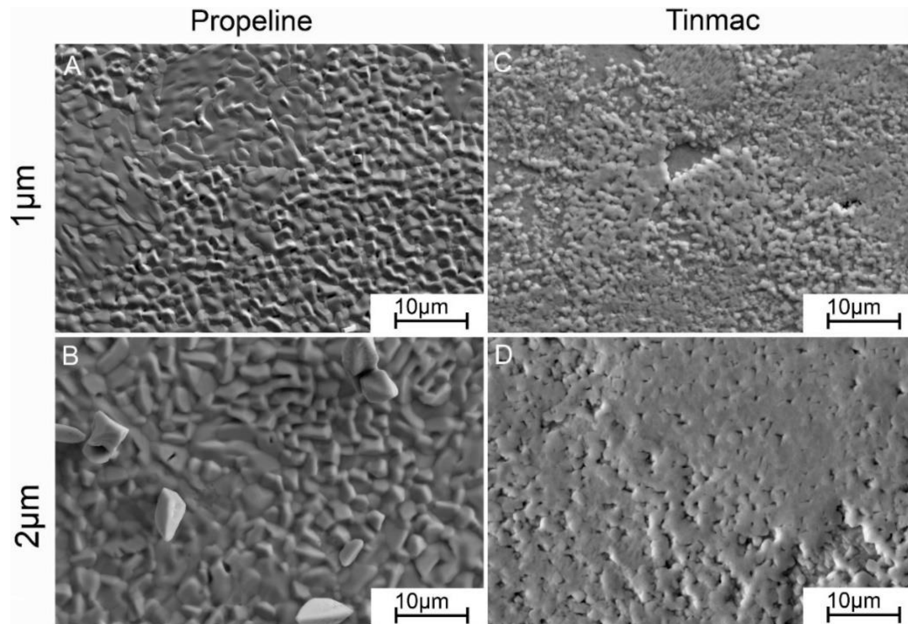


Figure 4.78: SEM micrographs at high magnification of tin electrodeposits shown in Figure 4.75

4.5.2 *Electrodeposit crystallographic comparison*

EBSD pole figure maps for deposits produced from both the electrolytes are shown in Figure 4.79. The deposit with similar orientation, containing many grains, ignoring mis-indexed points, for Figures 4.79.A and C resemble those of the copper substrates in Figure 3.2 suggesting both electrolytes produce deposits which are affected by the underlying copper at 1 µm in thickness. Furthermore, the similarity of the deposit grain structure to the underlying copper grain structure decrease as the deposit becomes thicker, as seen between Figures 4.79.A and B.

Additionally, Figure 4.79 suggest the variation in deposit surface topography in Figure 4.77 to be caused by grain growth of different grain orientations as both vary on similar scales. Moreover, these EBSD maps shows a difference in preferred crystallographic orientation for deposits produced from the different electrolytes depicted by colour changes within the maps. This orientation change is best shown with the use of folded pole figures in Figure 4.80.

Figure 4.80.C and D show that as the Tinmac deposit becomes thicker the 001 texture becomes more pronounced. Furthermore, EBSD maps and pole figures show that the crystallographic orientation of the tin deposited from the Propeline and Tinmac electrolytes is different. This therefore shows that deposit crystallographic orientation is not solely dependent upon deposit thickness but also on the composition of the electrolyte, as frequently seen for other electrodeposits [57], [77]–[80].

These results show that, although the deposits produced from Tinmac undergo an orientation change as their thickness is increased, the deposit produced from Propeline shows little difference in orientation between Figures 4.80.A and B. This might reflect the limited increase in deposit thickness that occurs as the plating time is doubled i.e. although deposition time has doubled the uniform thickness of the deposit, which can be measured, has not.

The grain sizes of the Propeline and Tinmac electrolytes appear similar to those of the underlying copper grains, seen in Figure 3.2 with grain sizes between 10 and 50 µm, which suggests that these deposits, at these thicknesses, are undergoing epitaxial growth. Furthermore, the change in grain size and preferred texture for the Tinmac deposits with increasing thickness suggests that this epitaxial effect is reducing. Therefore, the crystallographic orientation of the tin deposits at these thicknesses can be shown to be additionally dependent upon underlying copper substrate orientation. Whereas, in the literature electrodeposits epitaxy in is said to be rare [264] and normally seen for thin films produced from other materials, frequently oxides [265].

Additionally, if the reduction in substrate effects, seen with the Tinmac deposits, is also reducing for the deposits produced from Propeline at similar thicknesses, then the non-uniform growth seen on the deposits produced from Propeline may be a result a reduced epitaxial affect. Moreover, this would suggest the uniformity of the 1 µm deposits produced from Propeline is caused by substrate effects. On the other hand, the non-uniform growths may initiate due to preferred growth directions accelerated by the chemicals within the electrolyte or hydrogen evolution encouraging dendritic growth [266].

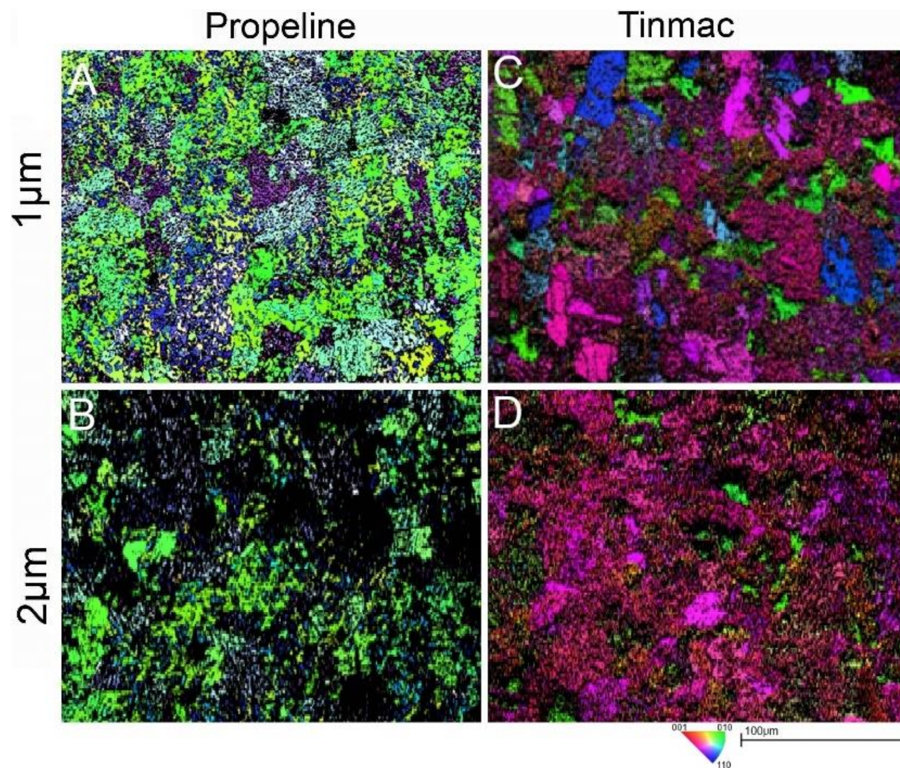


Figure 4.79: EBSD Z pole figure maps for tin electrodeposits shown in Figure 4.75

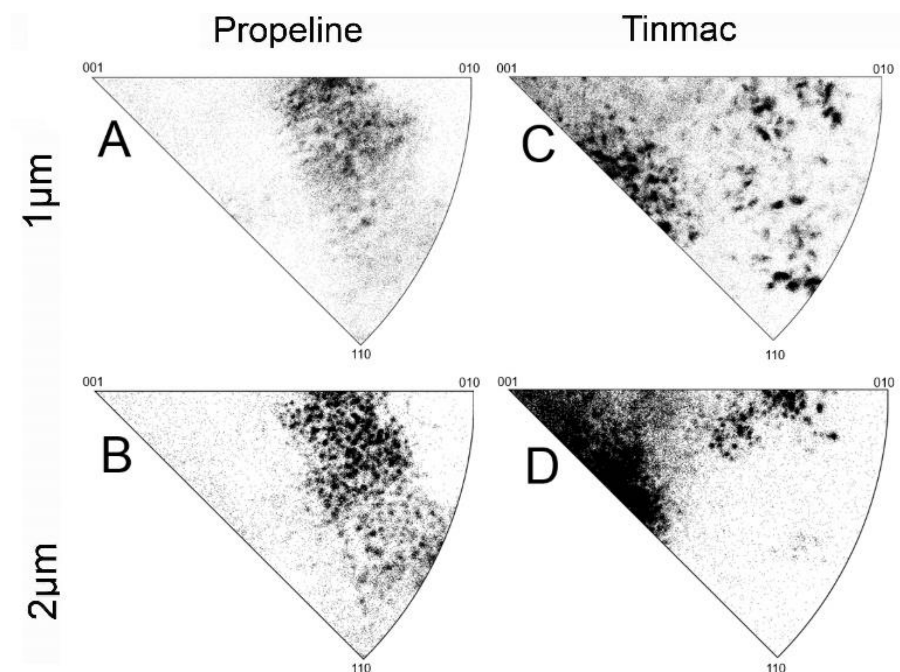


Figure 4.80: EBSD Z folded pole figures for of tin electrodeposits shown in Figure 4.75

4.5.3 *Electrodeposit contamination comparison*

During electrodeposition it is possible to contaminate the deposit with species from the electrolyte. XPS sputter depth profiling carried out to assess any deposit contamination by the

electrolyte for both 1 µm thick electrodeposits produced from Tinmac and Propeline electrolytes. These tests were performed on the deposit surface and subsurface, observed after surface sputtering, with data shown in Figure 4.81.

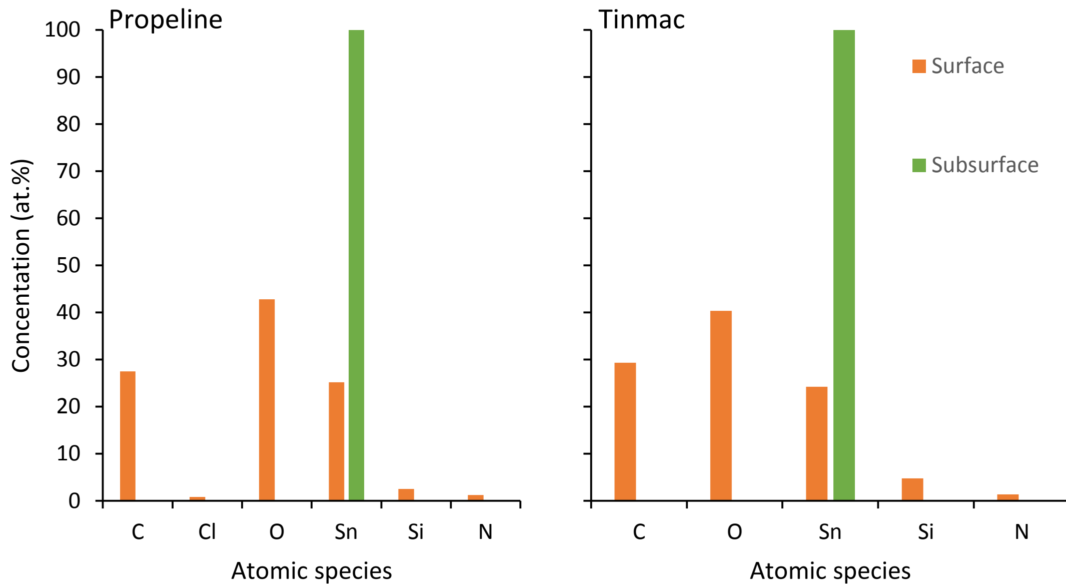


Figure 4.81: XPS survey data of surface and subsurface for 1 µm thick electrodeposits produced from Propeline and Tinmac

Both sets of data, in Figure 4.81, show surface contamination in the form of carbon, silicon and nitrogen, most likely caused during sample storage. However, the presence of chlorine on the surface of the non-aqueous deposit suggests imperfect electrolyte removal during post-deposition rinsing. This surface contamination may result in additional surface oxidation and increase whisker propensity, but due to its low levels this was not of concern. Moreover, the results for both deposit's subsurface showed 100 at.% tin present indicating no measurable contamination or co-deposition.

4.5.4 *Electrodeposit IMC growth comparison*

Analysis of the IMC growth from 2 µm electrodeposited tin coatings formed from the aqueous and non-aqueous electrolytes was carried out after storage at room temperature for 7 and 177 days. IMC distributions are shown in Figure 4.82.

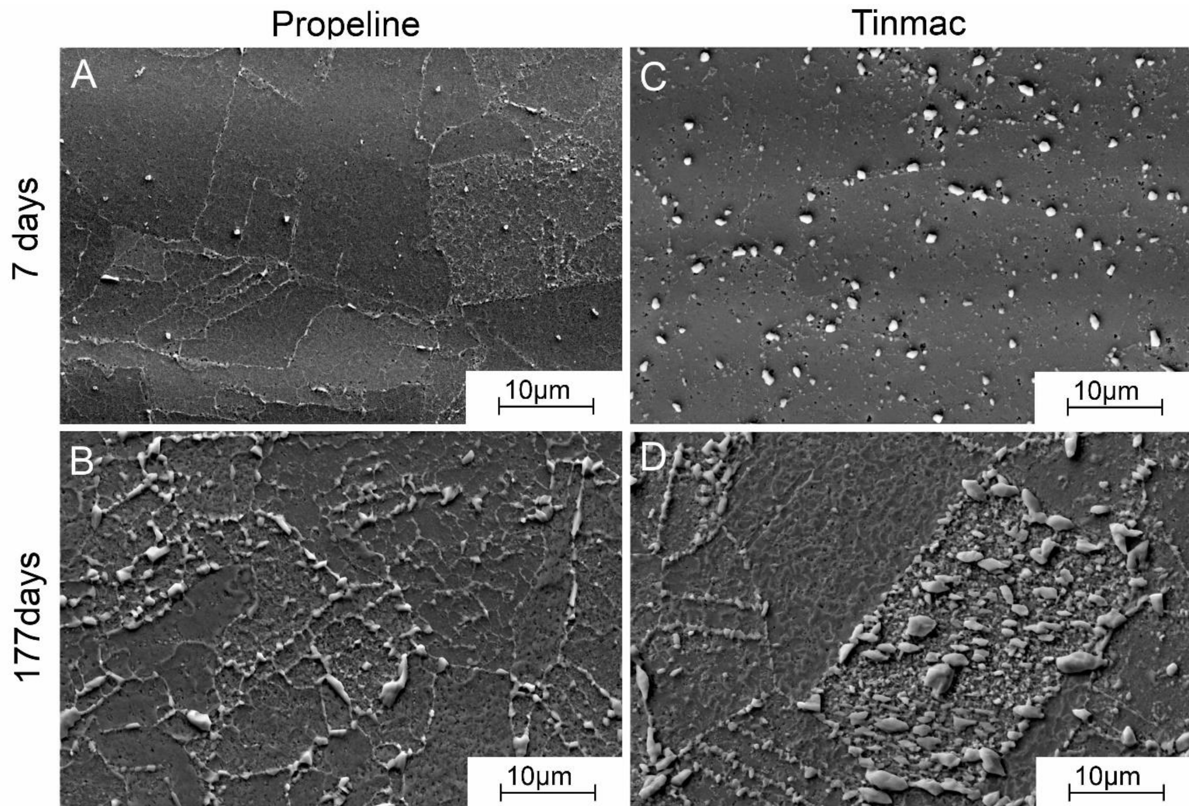


Figure 4.82: SEM micrograph of IMC growth from tin electrodeposits produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 (A and B) and Tinmac proprietary aqueous bright tin bath at 20 mA/cm^2 (C and D) at nominal thicknesses of 2 μm. A and C show 7 days ambient storage before tin stripping and B and D show 177 days ambient storage before tin stripping

The IMC growth for the deposits from the aqueous electrolyte after 7 days storage is made up of large discrete particles, whilst the non-aqueous electrolyte forms fewer large individual particles and more continuous growth along copper grain boundaries and upon specific grain orientations.

For samples stored for 177 days, Figures 4.82.B and D, large discrete particles are still prevalent for the deposit from the aqueous electrolyte, whereas the IMCs formed from the non-aqueous electrolyte are localised along copper grain boundaries. Additionally, the IMCs formed from the non-aqueous electrolyte shows a finer scale structures, thought to correspond to IMC growth at previous tin grain boundaries. This finer structure, shown more clearly in Figure 4.83, is more pronounced and more faceted for the non-aqueous tin deposit compared with that from the aqueous deposit. In addition, IMCs formed from the tin deposit produced from the non-aqueous electrolyte would appear to be wedge shaped, faceted with a single peak, much like the description commonly seen in literature describing IMC growth forming along tin grain boundaries [72].

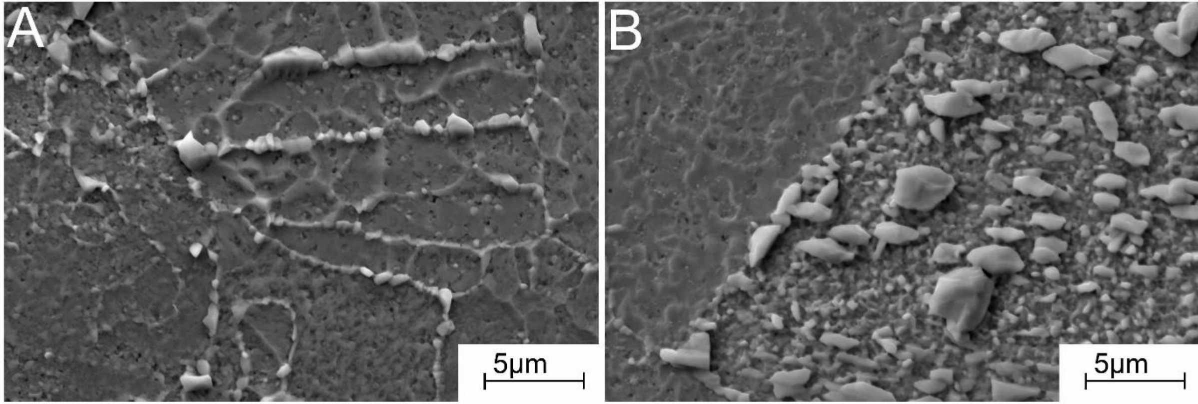


Figure 4.83: SEM micrograph of IMC growth for tin electrodeposits seen in Figures 4.82.B and D at high magnification showing fine IMC structure

4.5.5 *Electrodeposit whisker growth comparison*

Since whisker growth is dependent upon deposit thickness, only the 1 μm deposits were analysed as the 2 μm thick deposits formed from the non-aqueous electrolyte did not reach their desired thicknesses. Whisker growth was observed from deposits produced from Propeline with examples shown in Figure 4.84.

Figure 4.85 illustrates whiskers featuring cross-sections with significant reductions in thickness. The reason for this whisker thinning could be due to a reduction in compressive stress within the deposit at the whisker base as stress is relaxed during its growth. The reduction in stress may be caused by the thickness of the deposit reducing long range diffusion. Therefore, after close range stresses have relaxed the replenishment of stress is reduced leading to a decreasing stress and a smaller whisker cross-section.

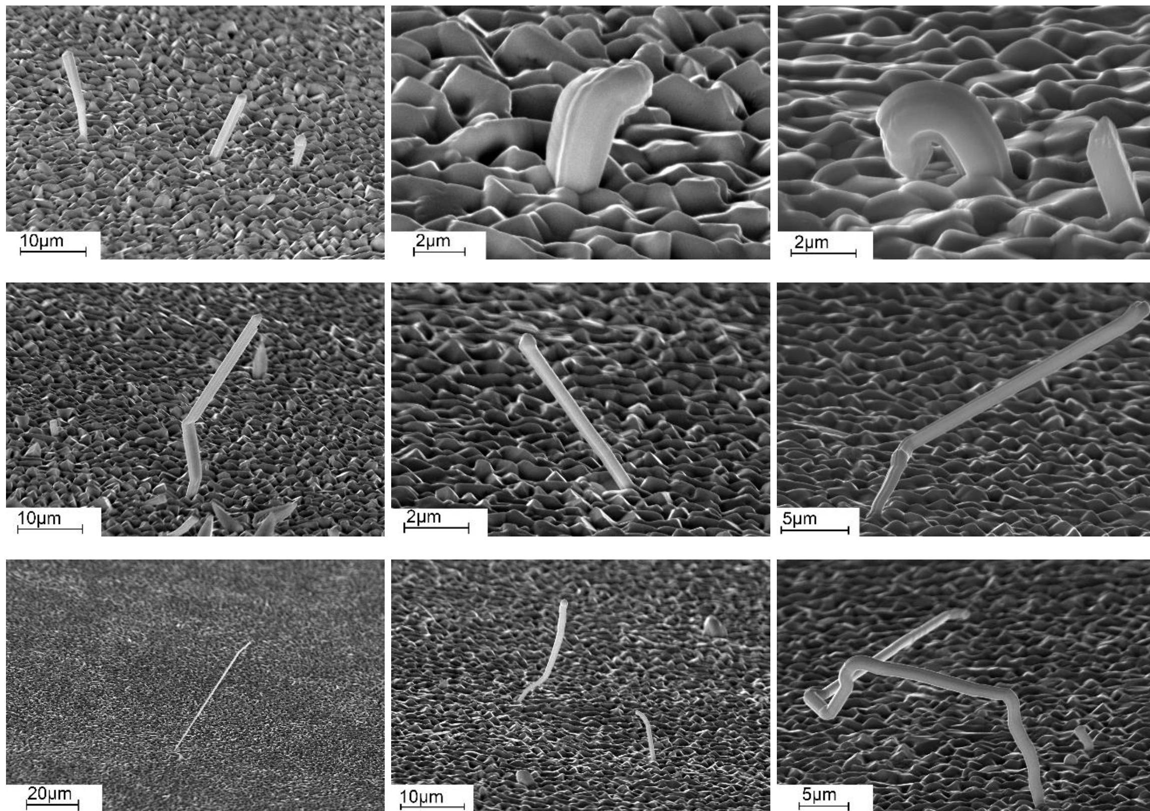


Figure 4.84: SEM micrographs, viewed at 70^0 tilt of whiskers grown on 1 µm tin electrodeposits produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 days ambient storage

Although this contradicts current whisker thermodynamics [77], as a smaller cross section requires a larger stress relaxation per unit volume, this observation suggests that as stresses increase resultant whisker diameter increases. The question which arises from this theory is: If reducing stresses causes whisker thinning, why is this not more commonly seen as whiskers reduce deposit stress? Therefore the reason why these whiskers have reduced cross-sections is still unknown.

Moreover, whisker thinning, evident in Figure 4.86, often produced striations along the electrodeposit from where the tip of the whisker was originally situated prior to growth. In the case of the whisker in Figure 4.86 the tip of the whisker features a smooth surface, absent of common whisker features, and as such the tip is considered to exhibit the surface finish from the electrodeposit.

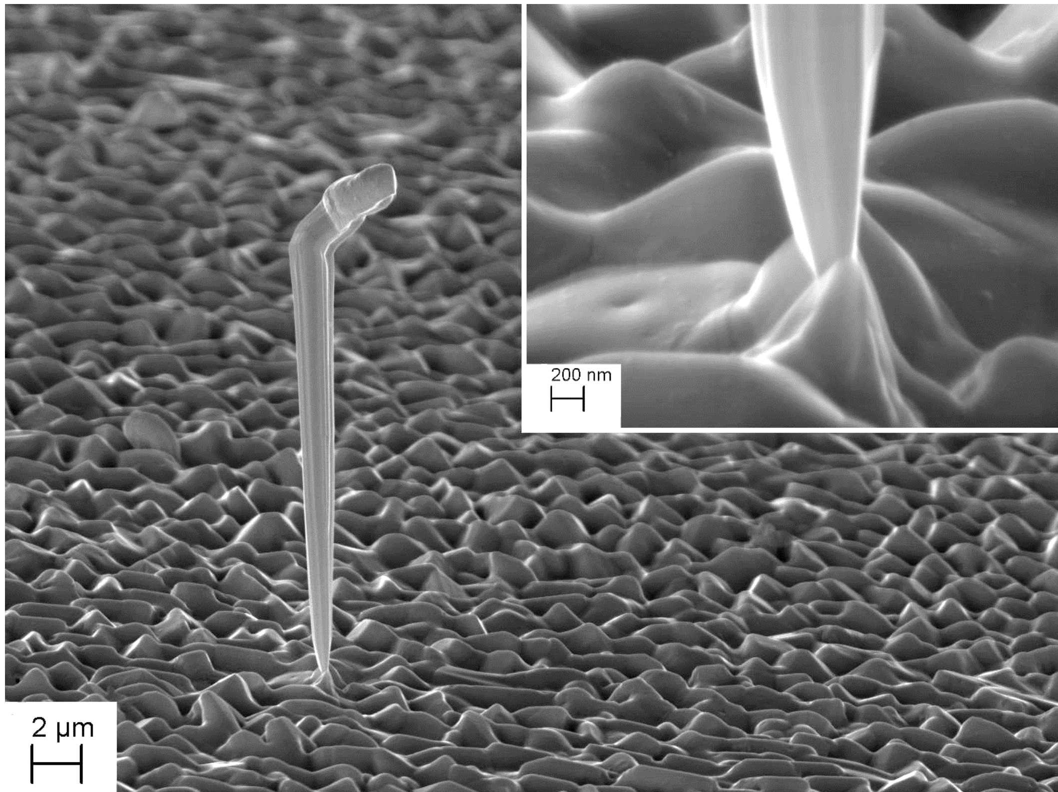


Figure 4.85: SEM micrograph viewed at 70^0 tilt of tin whisker grown from an electrodeposits produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 day storage showing reducing in cross section with high magnification inset

Furthermore, the shape of the tip in Figure 4.86 is additionally outlined at the base of the whisker with concentric striations where the whisker tip was once situated. These concentric striations on the electrodeposit at the whisker base are most likely caused by whisker thinning producing a reduction in cross-section. This thinning is then able to create striations by the formation of oxide during constant cross-section growth followed by cracking of the oxide at the deposit-whisker vertex during a reduction of cross-section.

In addition, whisker thinning as seen in Figures 4.85 and 4.86 does not take place in a single abrupt motion but rather occurs over a period of the whiskers growth. This is exhibited in both Figures with a smooth reduction in cross-section, although the rate and duration of this reduction in cross-section is seen to vary for each whisker. As the duration and rate of whisker thinning is unknown it may be possible for a whisker’s cross-section to reduce to such an extent that they detach themselves from the deposit, or cause the base to become fragile causing fracture due to vibrations or bending. Furthermore, this would additionally complicate whisker density measurements as a reduction in counts could be possible as whiskers are removed from the deposit.

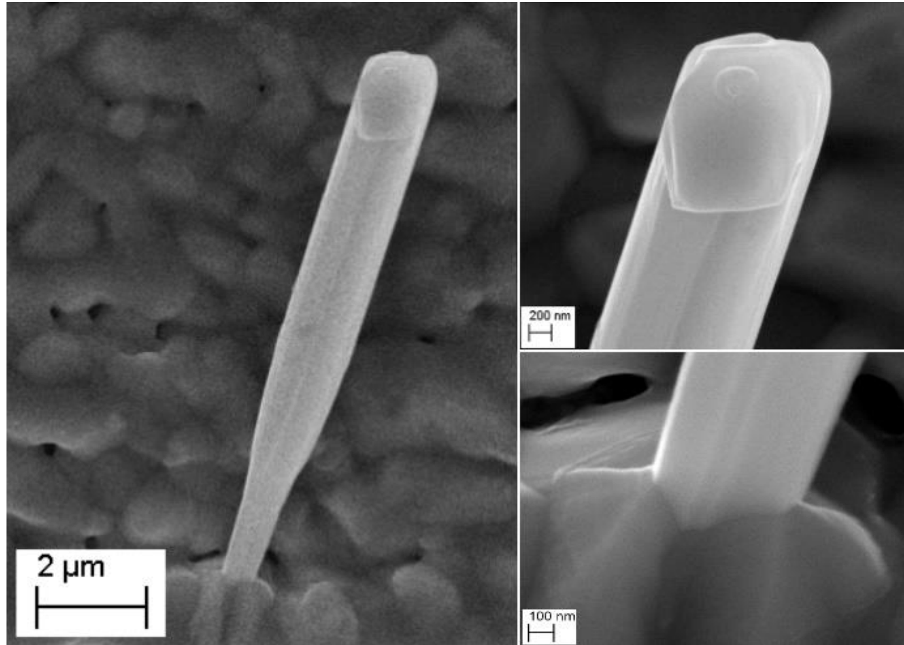


Figure 4.86: SEM micrograph viewed at 70^0 tilt of tin whisker grown from a 1 µm tin electrodeposit produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 day storage at ambient showing whisker tip and base at high magnification

In addition to whisker thinning, whisker thickening has also been exhibited subsequent to a short period of whisker thinning. Whiskers in Figures 4.87 and 4.88 both demonstrate stepped-striations around their circumference which decrease the whisker cross-section and are followed by an increase. Whiskers frequently show small striations around their circumference [201], but not normally to this degree. These stepped-striations may be caused by an oscillation in compressive stress within the deposit leading to an oscillation in whisker cross-section during growth. Moreover, these examples of whisker thickening only take place after whisker thinning. Therefore, this suggests that the original cross-section of the whisker is the maximum thickness which it is able to possess and stepped-striations are only possible after whisker thinning has taken place.

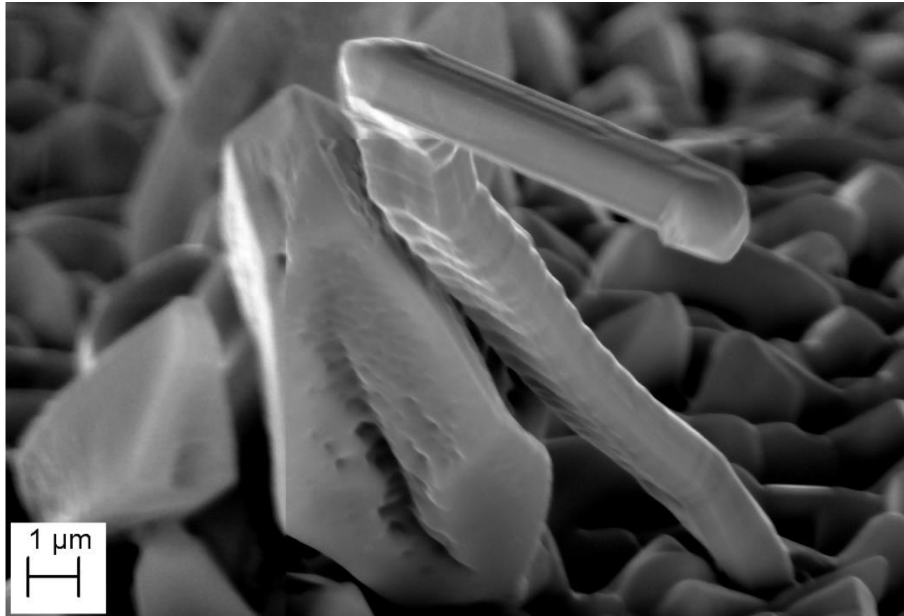


Figure 4.87: SEM micrograph viewed at 70^0 tilt of a tin whisker growing from a 1 μm tin electrodeposit produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 day storage at ambient

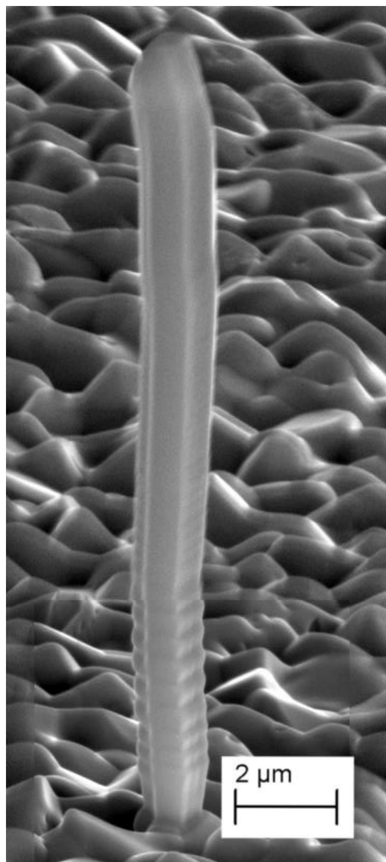


Figure 4.88: SEM micrograph viewed at 70^0 tilt of a tin whisker grown from a 1 μm tin electrodeposit produced from Propeline with additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 day storage at ambient. The whisker shows stepped-striations along its length

Whisker diameters within literature are frequently said to be 1-5 µm [8], [201] whereas whiskers grown from these electrodeposits featured 0.38-2.4 µm diameters with an average of 1.2 µm. Although, the maximum whisker diameter for these whiskers is within normal limits the minimum is less common and possibly caused by the thickness of the deposit, its small grain sizes or its uneven topography.

Additionally, the high whisker density present in these deposits is illustrated in Figure 4.89. The resulting whisker densities for the deposits produced from the non-aqueous electrolyte after 250 days storage are shown in Figure 4.90. Three identical electrodeposits were produced from the Propeline electrolyte to increase surface areas thus showing reproducibility and increasing statistical reliability. Of the three coupons produced, two produced whiskers, of these whiskering electrodeposits there were variations in the number of whiskers counted during each assessment showing a variation in whisker density across the surface of each coupon and between coupons. The average of these whisker densities for these samples for the non-aqueous electrolyte was 0.64 whiskers/mm^2.

Figure 4.89: SEM micrograph viewed at 70 ^{0} tilt of tin whisker grown from an electrodeposit produced from Propeline with an additions of 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2 after 250 day storage showing a high density of long whiskers

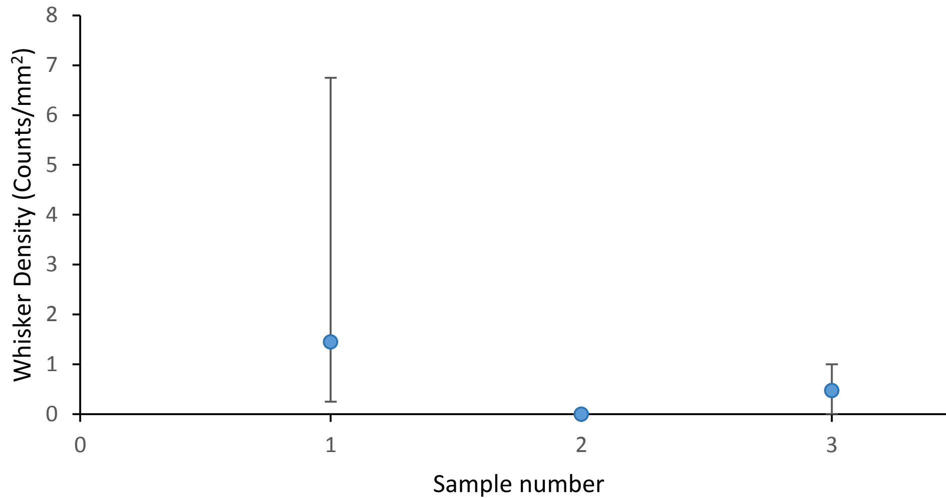


Figure 4.90: Average density values for three identical 1 μm thick tin deposits produced from Propeline with 0.044 M $SnCl_2.2H_2O$ at 10 mA/cm^2. Error bars representing maximum and minimum of collected data

Analysis of 1 μm thick tin deposits formed from the Tinmac aqueous bright tin bath at 20 mA/cm^2 resulted in a whisker count of zero for each sample. Thus, for the deposition parameters investigated it is clear that the deposits formed from the non-aqueous electrolyte showed an increased propensity for whisker growth compared with deposits prepared from a specific aqueous electrolyte (Tinmac).

4.6 Creation, development and testing of 3Dview for whisker length measurement

Most frequently used methods of length measurement by SEM are undertaken directly above the substrate. This allows for accurate measurements of objects which lie parallel to the substrate's surface, but for lengths which this is not the case can appear shorter, even to the point of disappearing altogether. This therefore requires three dimensional measurements to be acquired to minimise error. These calculations can be undertaken manually but due to the large amount of data collection required for whisker growth measurement this is best undertaken in an automated system. On the other hand, issues arise from fully automated systems attempting to map three dimensional surfaces due to the complex shapes of whiskers, therefore, the most effective means of whisker measurement is with the use of a semi-automated system which logs user actions, calculates Cartesian coordinates and records data. It is this semi-automated system which is featured within this thesis under the name of '3DView'.

From acquiring SEM micrographs at two different tilts it was possible to measure whisker lengths in three dimensions using a bespoke software package named '3DView'. This programme was developed for this project and was created using Visual C++ in Microsoft Visual Studio.

The operation of the programme proceeds through several mathematical and logic based processes to produce Cartesian coordinates in three dimensions. The processes which the user and programme undertake are shown in the flow diagram in Figure 4.91. In addition, the abbreviations used in this section are shown in Table 4.5

Table 4.5: Abbreviations used during programme development.

Abbreviation	Full text	Description
POV	Point of view	The location of observer i.e. Electron source for SEMs.
POI	Point of interest	A point of particular interest at a particular location that is seen in both images
LOS	Line of sight	The line which connects the POV and the POI.
Ref point	Reference point	A POI which defines the coordinate origin.
WD	Working distance	Distance from POV to POI

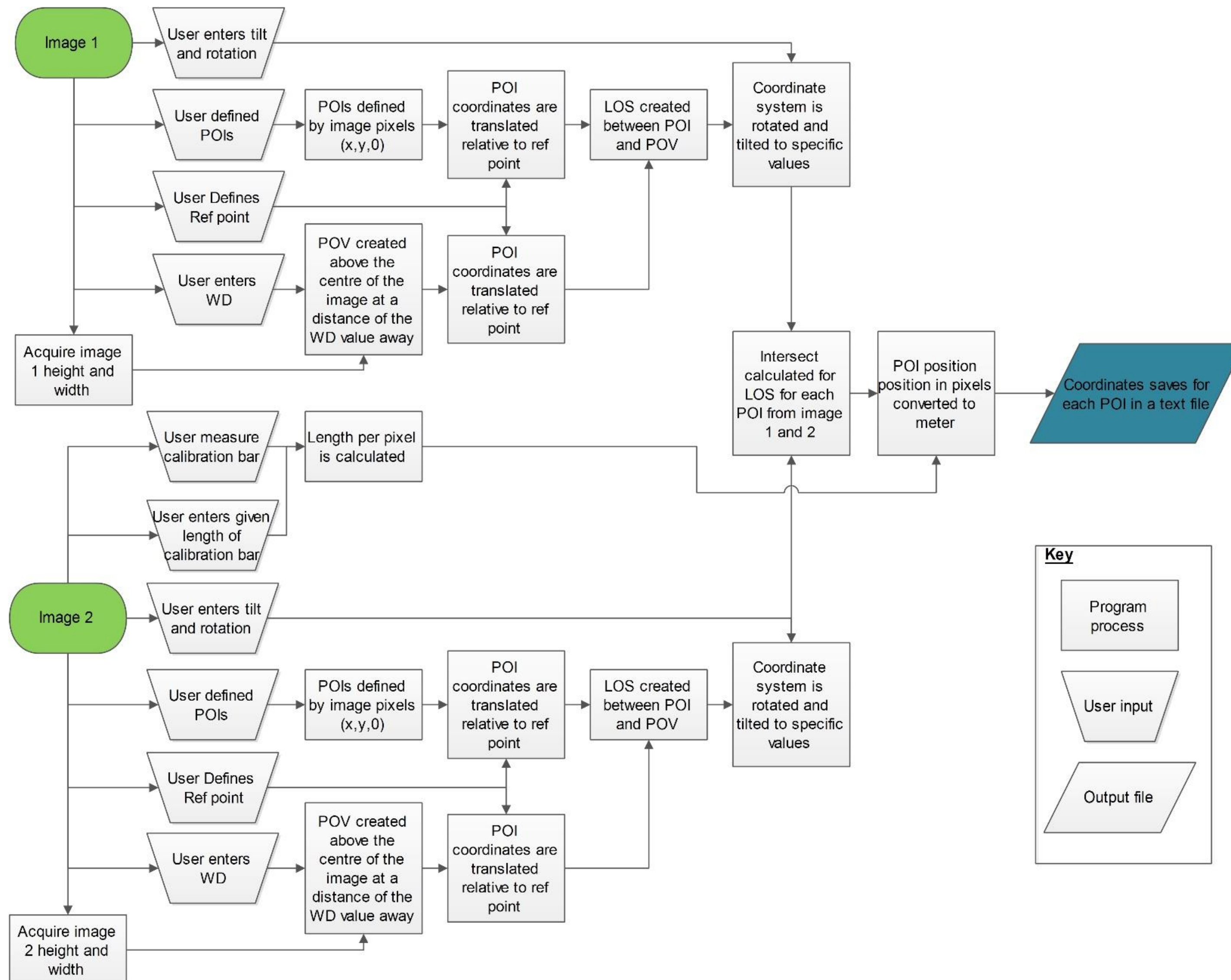


Figure 4.91: Flow diagram for 3Dview of user inputs, programme logic and outputs files creating three dimensional coordinates from two images at different tilts

The majority of the coding processes seen in Figure 4.91 are common place e.g. basic logic, data manipulation and visual input/output controls. Although this is the case, there are specific mathematical operations that are undertaken which make this programme unique; the translation of a three dimensional coordinate system, the rotation of a three dimensional coordinate system about a point and the calculation of the coincident point for two POVs.

4.6.1 *Translation of a three dimensional coordinate system*

The translation of a coordinate system moves all its points by a specific vector. This is then able to allow the origin to be moved to a specific location. In the case of the process for this programme, it is creating a common origin for image 1 and 2 at a known POI (defined as the reference point). To achieve this the transformation shown in Equation 4.2 is undertaken. Basically, this process is used to bring both images to overlap at a given POI.

$$X^R = X^I - X^r \qquad \text{Equation 4.1}$$

$$\rightarrow (x_R, y_R) = (x_I - x_r, y_I - y_r) \qquad \text{Equation 4.2}$$

Where: X^R is the coordinate set (x_R,y_R), (x_R,y_R) are the coordinates relative to the reference point, X^I is the coordinate set (x_I,y_I), and (x_I,y_I) are the coordinate relative to the image origin.

4.6.2 *Rotation of a three dimensional coordinate system*

The other complicated mathematical operation that occurs in the core programming is the rotation of each image's coordinate systems about the reference point. This rotation is a mathematical transformation of a coordinate system as shown in Equation 4.3.

$$X' = R(axis, \emptyset)X \qquad \text{Equation 4.3}$$

Where; X' is the rotated axis coordinates, R is the second order tensor that operates by a rotation of Ø about a defined axis and X is the original coordinate set

The tensor R can be seen in three different forms when using Cartesian coordinates. It can be used to rotate about all three axes and can be seen in Equations 4.4, 4.5 and 4.6.

$$R(x, \emptyset) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\emptyset & -\sin\emptyset \\ 0 & \sin\emptyset & \cos\emptyset \end{bmatrix}$$ Equation 4.4

$$R(y, \emptyset) = \begin{bmatrix} cos\,\emptyset & 0 & sin\,\emptyset \\ 0 & 1 & 0 \\ -sin\,\emptyset & 0 & cos\,\emptyset \end{bmatrix}$$ Equation 4.5

$$R(z, \emptyset) = \begin{bmatrix} cos\,\emptyset & -sin\,\emptyset & 0 \\ sin\,\emptyset & cos\,\emptyset & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ Equation 4.6

Equations 4.4, 4.5 and 4.6 show all the possible rotations about each Cartesian axis. In the case of the process seen in the programme we firstly need to rotate the reference system for that image by the tilt value about the y axis and then the rotation about the x axis. The transformation that takes place in the programme can be seen in Equation 4.7.

$$X' = R(x, \beta)R(y, \alpha)X$$ Equation 4.7

Where; β is the angle due to the rotation, α is the angle due to the tilt and x and y are the original coordinate system unit vectors.

By using both these processes; translation and rotation of a coordinate system, two images can be overlapped and rotated allowing for features to be measured. This is then able to take two images, with lines of sight defined on each one for specific POIs, as seen in Figure 4.92, to be combined, as seen in Figure 4.93, allowing LOS for each POI in each image to become coincident producing the POI's coordinates in three dimensions. For a visual representation of the coincident points of the IOS, Figure 4.93 features whiskers, represented by cylinders, in a virtual model created from the whiskers present in the two images shown in Figure 4.92.

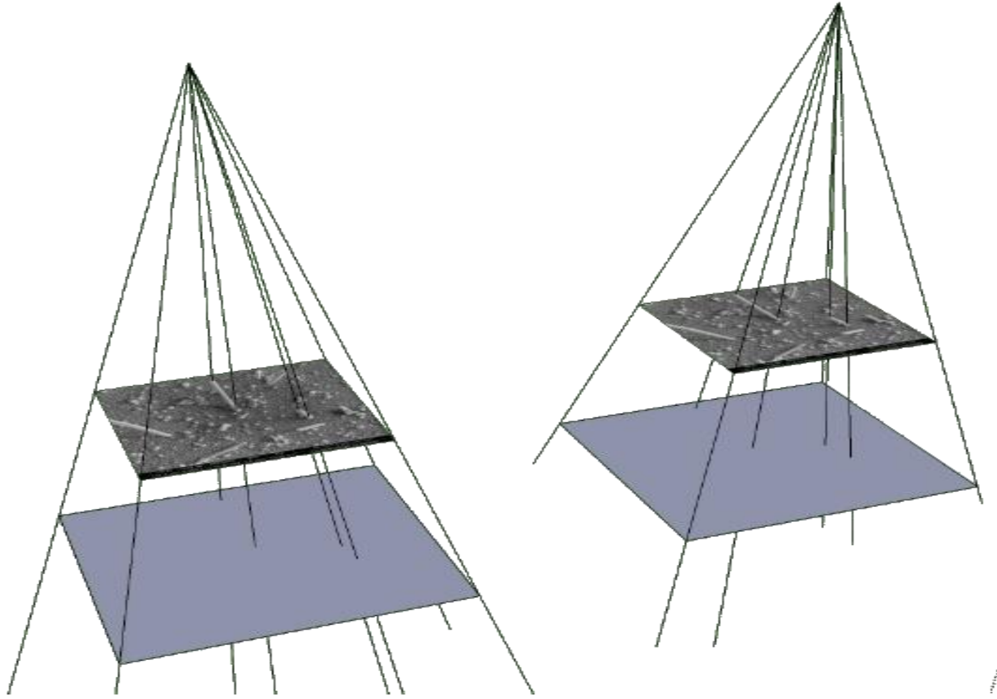


Figure 4.92: Virtual wireframe model of LOS for various POIs in two images at differing tilts (Separated).

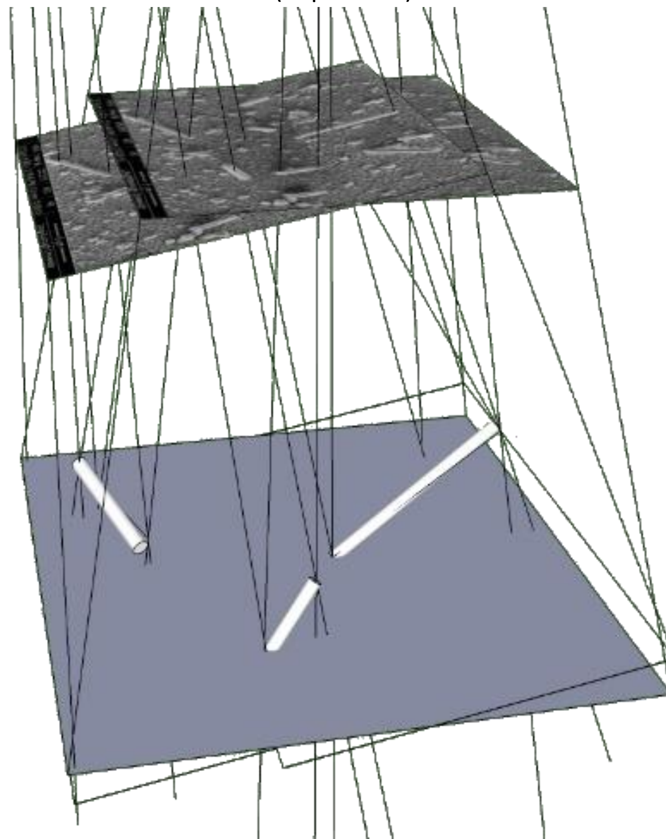


Figure 4.93: Virtual model of the overlap of wire frames shown in

Figure 4.92 creating coincident points in LOS allowing a model to be created featuring whiskers.

4.6.3 *Calculation of the coincident point for two POVs*

The mathematical concept of calculating the intercept of two lines, which in this case are the two LOS, is simple. The problem is, due to POIs being defined by pixel references, which are integer values, along with human error, the two lines which should cross rarely do. Due to these coincident points being vital for the programme's operation a solution was created. A line was created to pass between either LOS for each image. Then, when this line is at its shortest, which is simple minimisation of a differential equation, or by defining the additional line to be perpendicular to both the LOS, its midpoint is defined as the location of the POI in three dimensions. By undertaking this midpoint calculation it firstly solves the issue of the lines not crossing but in addition, reduces the human and systematic error when selecting the POIs in each image.

Mathematically this process is undertaken by defining the two LOS as shown in Equations 4.8 and 4.9, respectively.

$$P_a = P_1 + mu_a(P_2 - P_1) \qquad \text{Equation 4.8}$$

$$P_b = P_3 + mu_b(P_4 - P_3) \qquad \text{Equation 4.9}$$

Where: P_a and P_b are any point along line 1 and line 2 respectively, P_1 and P_2 are two known points along line 1, P_4 and P_3 are two points on line 2, m is a number between negative to positive infinity, and U_a and U_b are the gradients of lines 1 and 2 respectively

Then by defining the length of the line which lies between each LOS that goes from points P_b and P_a as shown in Equation 4.10.

$$S = \|P_b - P_a\| \qquad \text{Equation 4.10}$$

Where S is the length between points P_a and P_b.

Then by substituting Equations 4.8 and 4.9 into Equation 4.10 gives Equation 4.11.

$$S = \|P_3 + mu_b(P_4 - P_3) - P_1 - mu_a(P_2 - P_1)\| \qquad \text{Equation 4.11}$$

At this point there are then two ways of solving this; either by minimising the length of length S by calculus or by defining the line S as perpendicular to both LOSs.

4.6.3.1 *Calculus minimisation of length S*

The solution to Equation 4.12 will give two points, one on each LOS which are closest to each other. This then has Equation 4.11 substituted into it to give Equations 4.13 and 4.14.

$$\frac{\partial S^2}{\partial U_a} = 0 \text{ And } \frac{\partial S^2}{\partial U_b} = 0 \qquad \text{Equation 4.12}$$

$$\rightarrow \frac{\partial S^2}{\partial U_a} = \frac{\partial}{\partial U_a} \|P_3 + mu_b(P_4 - P_3) - P_1 - mu_a(P_2 - P_1)\|^2 \qquad \text{Equation 4.13}$$

$$\rightarrow \frac{\partial S^2}{\partial U_b} = \frac{\partial}{\partial U_b} \|P_3 + mu_b(P_4 - P_3) - P_1 - mu_a(P_2 - P_1)\|^2 \qquad \text{Equation 4.14}$$

It is at this point it is clear that the workings involved in this calculation are lengthy and basic. For these reasons the workings are omitted from this section.

4.6.3.2 *S is defined to be perpendicular to both LOSs*

The other means by which the location of the shortest line between each LOS is calculated, is to say that this line is perpendicular to each LOS. These points, one on each LOS seen to be the closest to each other (creating the shorted line between each LOS), can be found by solving Equations 4.15 and 4.16.

$$(\mathrm{P_a} - \mathrm{P_b}).(\mathrm{P_2} - \mathrm{P_1}) = 0 \qquad \text{Equation 4.15}$$

$$(\mathrm{P_a} - \mathrm{P_b}).(\mathrm{P_4} - \mathrm{P_3}) = 0 \qquad \text{Equation 4.16}$$

Again, it is at this point that it is clear that the workings involved in this calculation are lengthy and basic. For these reasons the workings are omitted from this section.

The solution to both these methods produces two points each along a LOS which is closest to another. These points are simply averaged giving the final result; the position of the POI in three dimensions.

4.6.4 *Testing of 3Dview*

For obvious reasons a bespoke programme should be tested to verify its ability to create reproducible and accurate results. For this, task three processes were used; a whisker-like virtual model from the programme Google SketchUp™, the measurement of a scratch on a flat copper substrate viewed from various non-normal angles and the measurement of whiskers from a bismuth sputter coating at varying tilts.

4.6.4.1 *Virtual agreement with ‘Google SketchUp™’*

The programme ‘Google SketchUp™’ was used to create a virtual model of cylinders protruding from a flat surface. An image of the model is shown in Figure 4.94.

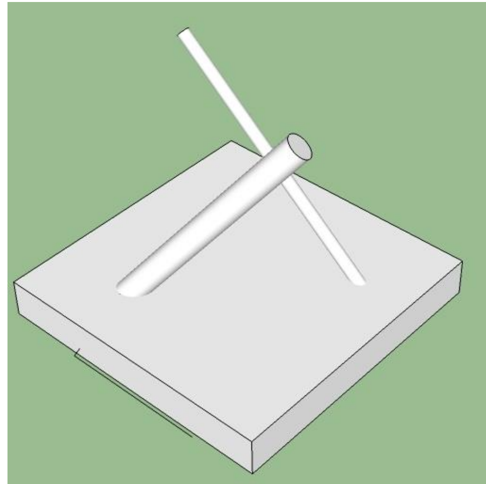


Figure 4.94: Google SketchUp™ model of cylindrical protrusions from a flat surface representing whisker growth from a deposit

By using a model like the one shown in Figure 4.94, two images were collected at differing known angles. Although both images shortened the lengths of the cylinders 3Dview was able to calculate their lengths. These values of calculated length were then compared to the known length of the cylinders from Google SketchUp™. The results of this comparison are shown in Table 4.6.

Table 4.6; Known length compared to the result using 3Dview from with two images at different tilts of a virtual model created in Google SketchUp™.

	Known length	Programme measurement	Accuracy (Programme result / Google SketchUp™)
Cylinder 1	4.36µm	4.35µm	99.8%
Cylinder 2	3.27µm	3.27µm	100%
Cylinder 3	4.19µm	4.17µm	99.5%

The high values of accuracy in Table 4.6 validates the programme results to be in agreement with another, far more elaborate programme. But to be fully validated, results from real samples must be shown to be accurate.

4.6.4.2 *Copper scratches*

Another test of 3Dview was to create a scratch in a flat copper sheet followed by measurement from the vertical, to give its actual length, followed by measurement from two non-vertical tilts. These two non-vertical tilts would depict the scratch as shortened allowing for 3Dview to create a calculated length for comparison compared to the actual length. The two tilted micrographs were

taken from 15 and 35 degrees to the vertical causing the scratch to measure 1.37mm and 1.20mm respectively as shown in Figure 4.95.

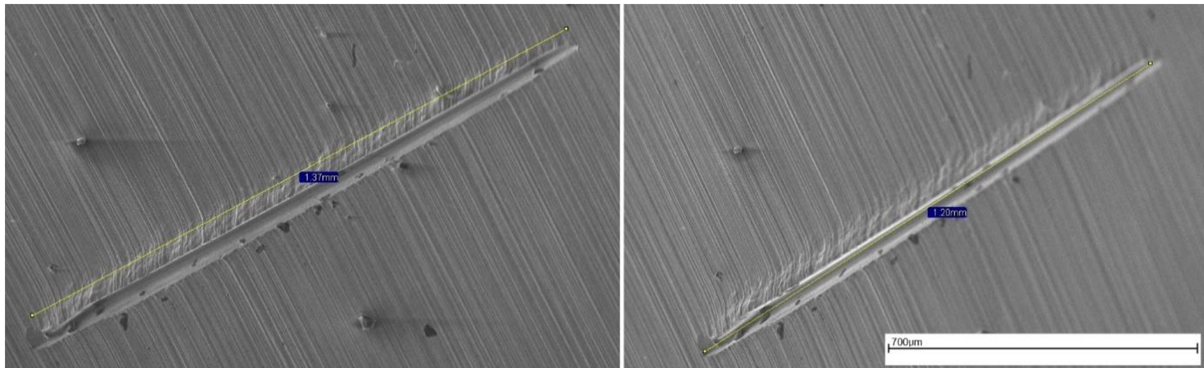


Figure 4.95; SEM micrographs of a scratch made in copper captured at 15 degrees from the vertical (left) and 35 degrees (right). Featuring scratch lengths measured at the given tilts of each image.

The micrographs shown in Figure 4.95 were then used in 3Dview and produced the results shown in Table 4.7 compared to the actual length measured from the vertical.

Table 4.7; Measured length of scratch in copper compared to calculated length from two non-vertical tilts in 3Dview.

Source	length
0 tilt micrograph	1.44mm
3Dview result	1.48mm

This result only contains 3% error. This form of testing may seem misplaced, but it demonstrates that a feature's length can be calculated from two images that are taken at non-perpendicular positions; which is fundamental to the measurement of whiskers.

4.6.4.3 *Reproducing lengths with real whiskers*

To validate the created programme, four images taken at different tilts of bismuth whiskers were used [139], as these whiskers are not kinked and simple to measure. This created six sets of results from the 6 possible pairings of the four images. An example of one of the images that was captured and used in 3Dview is shown in Figure 4.96 which led to producing the results in Figure 4.97.

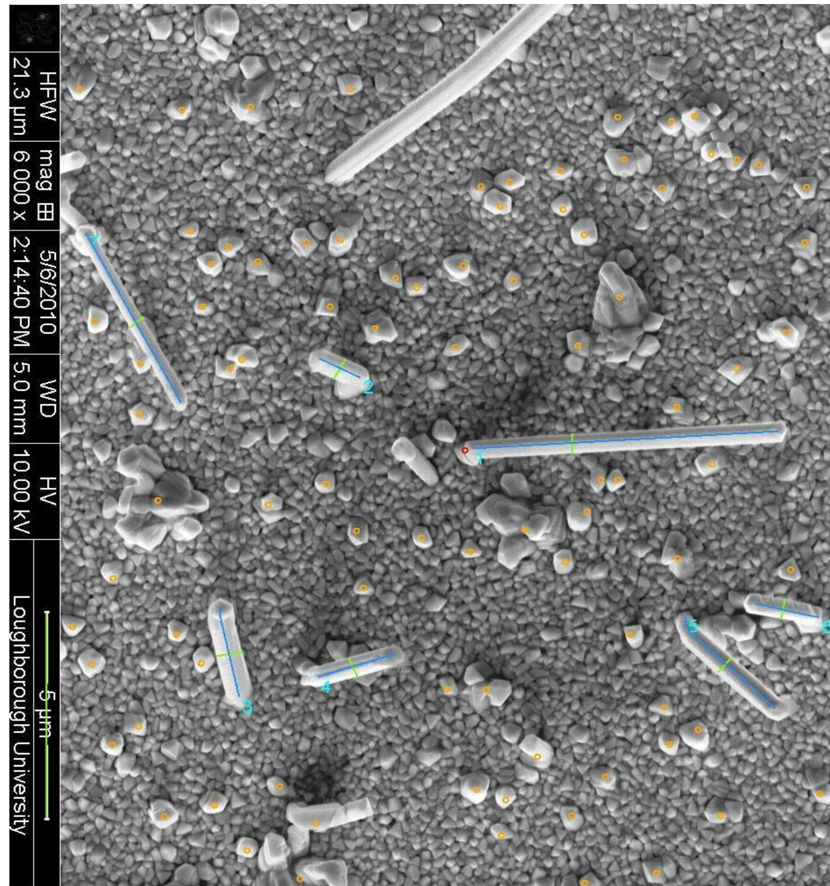


Figure 4.96: SEM micrograph of bismuth whisker captured after being used in 3Dview. Whiskers 1-5 are measured in blue, reference point in red, diameter and calibration in green and hillocks counting in orange [139]

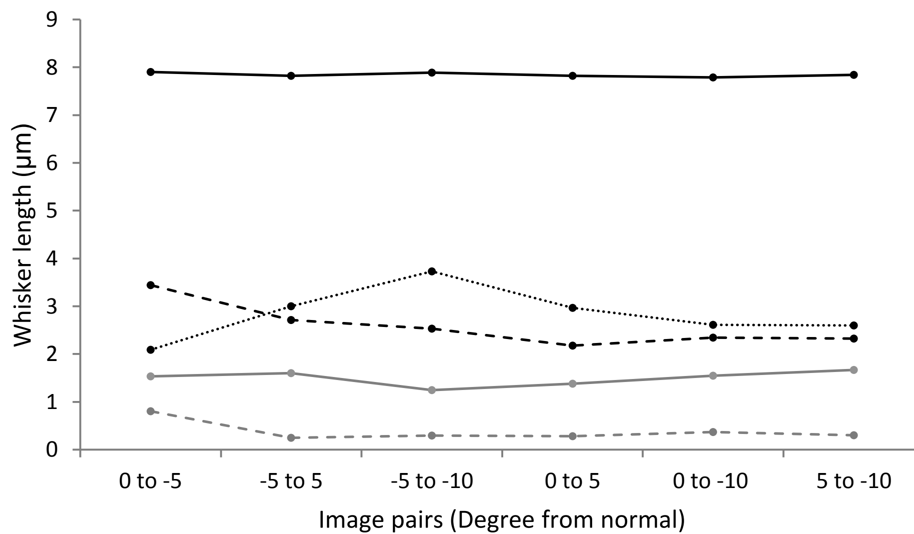


Figure 4.97: Lengths of whiskers in Figure 4.96 measured from different tiled images pairs used in 3Dview. Black solid line – Whisker 1, black dotted line – whisker 2, black dashed line – whisker 4 part 1, grey solid line – whisker 3 and grey dashed line – whisker 4 part 2.

Table 4.8: Average lengths and standard deviations of whiskers seen in Figure 4.96 using 6 image pairs in 3Dview.

	Average length (µm)	standard deviation (µm)
Whisker 4 part 2	0.38	0.19
Whisker 3	1.49	0.14
Whisker 4 part 1	2.59	0.42
Whisker 2	2.83	0.50
Whisker 1	7.84	0.04

The results of the whiskers measured in Figure 4.96 as shown in Figure 4.97 and Table 4.8 show the results to be of reasonable lengths and also reproducible, shown by low values of standard deviation.

4.6.4.4 *Closing remarks on 3Dview testing*

Overall, in each testing method used i.e. virtual modelling, single feature measurement, and measuring the length of multiple real whiskers featuring kinks, has shown that 3Dview has been capable of measuring features and whiskers accurately. Moreover, the results that have been produced by the programme have been considered to be reasonable and reproducible. In addition, the error produced by this process has been shown to be acceptable during all means of testing.

Chapter 5 Synoptic Discussion

The proprietary bright tin acid sulphate electroplating electrolyte was shown to produce 10μm thick bright tin deposits over a range of current densities with and without agitation. At the tested limits of current density it was shown that deposits featured a satin finish due to a faceted topography, at low current density, and vertical streaks due to hydrogen bubble formation at high current densities. Furthermore, the current density range over which uniform, homogeneous bright deposits were produced increased with the application of electrolyte agitation. In addition, grain shape and size were shown to vary with respect to current density and agitation. The reasons for these variations in grain size, shape and orientation with respect to the examined variables may be due to conventional electrolyte dynamics but, a more probable reason, these effects are resultant from the variation of applied over potential interacting with the additive formulation within the electrolyte causing deposit variation.

Voltammetry of DESs with additions of $SnCl_2.2H_2O$ in this thesis reported larger potential windows for tin reduction than those known for aqueous electrolytes, providing a significant range of cathode current efficiencies and resulting electrodeposition rates. As seen within the literature review these electrolytes are said to be favourable for industrial applications, with many highly beneficial qualities, they are, however, subject to a number of serious drawbacks such as their innate high viscosities leading to low mass transport and limited deposition rates. One approach to partly overcome these limitations was found to be to increase the $SnCl_2.2H_2O$ concentration. This resulted in some of the electrolyte formulations being able to produce deposition rates and cathode current efficiencies equivalent to those for aqueous formulations. Moreover, these formulations exhibited minimal reductions in cathode current efficiencies, even at the highest current densities examined, suggesting that even higher deposition rates can be achieved with these liquids.

Overall, although DESs containing high concentrations $SnCl_2.2H_2O$ are shown to be effective electrolytes, with respect to deposition rate and cathode current efficiency, they are more expensive, and most importantly, produce far less uniform deposits. Therefore to make tin containing DESs industry relevant the deposits produced must exhibit a highly desirable characteristic; such as a low whisker propensity. From the deposits produced within this project, higher levels of whisker growth have been observed compared with an aqueous alternative. However, the aqueous electrolyte utilised in this study has been shown to produce deposits which exhibit significantly low levels of whisker growth [59]. As such, this may be an overly negative representation of this electrodeposit as it is being compared to a very low whiskering counterpart.

Although many of the combinations of current density and $SnCl_2.2H_2O$ concentrations investigated produced low cathode current efficiencies and low deposition rates, all deposits were characterised with respect to uniformity and homogeneity. The majority of the deposits created during trials to optimise deposition variables were seen to be highly non-uniform. Therefore, by utilising deposits which showed the highest uniformity it was found that an electrodeposit could be produced that was between 1 and 2 µm in thickness. This uniformity was attributed to substrate effects upon the deposits occurring at low thicknesses, i.e. epitaxial-like growth, whereas at increasing thicknesses the influence of the substrate was reduced leading to the deposition of a non-uniform form. On the other hand, the increase in the non-uniformity of the deposit, and the reduced influence of the substrate could occur at similar thicknesses but actually be independent effects. In the author's opinion, the deposit uniformity observed at low thicknesses is due to the influence of the substrate on the initial stages of tin deposition.

The non-uniformity observed for many of the electrodeposits could be attributed to a common chemical used in all of the electrolytes examined within this study. Two such chemicals used within all electroplating baths were: $SnCl_2.2H_2O$; and choline chloride. If the uniformity of these deposits was caused by the use of a specific common chemical the use of alternative eutectic formulations, those not containing choline chloride, or different tin containing compounds, such as tin sulphate or tin fluoride, may result in deposits that are more uniform.

Within this thesis, work was undertaken to create a computer program which would calculate whisker lengths in three dimensions from two SEM micrographs at different tilts. The mathematical functions, testing, and code is contained within Chapter 10 Appendix, 4.6 Creation, development and testing of 3Dview for whisker length measurement, page 132 and can be used for future projects.

The electrochemical techniques developed and used in this work for the anodic stripping of tin were successful, however, the galvanostatic and potentiostatic methods each had advantages and limitations. The galvanostatic method was unable to remove the tin completely in its final stages, whilst the potentiostatic method was able to remove all the metallic tin. However, during the final stages of stripping sulphate formation was sometimes noted due to the uncontrolled high current densities that were developed during the removal of the bulk of the tin coating. The optimised method utilised an initial galvanostatic step followed by a potentiostatic step, which provided a quick and effective means of studying the IMC formed at the tin-copper interface.

Moreover, the tin stripping techniques developed within this project require only low concentrations of H_2SO_4, and the use of a potentiostat. As such, compared with alternative chemical

stripping techniques, this method uses lower acidity electrolytes, produces low financial and environmental cost waste for disposal. Furthermore, this technique allows for the removal of tin coatings quickly, using common laboratory equipment and acids, while, and most importantly, revealing clear and undamaged IMC formations present on the copper substrates leaving minimal remaining tin.

Deposits produced from the non-aqueous and aqueous electrolyte are shown to have different IMC growth, crystallographic structure, topography, and whisker propensity. Furthermore, the deposits produced from the non-aqueous electrolyte formed 0.64 whiskers/mm^2, whilst deposits prepared from the aqueous showed no whisker growth over the same time period. Although the whisker propensity of the non-aqueous electrolyte appears significant, compared with the result from the aqueous alternative, comparison of whisker propensities within literature of equivalent thickness deposits, featuring whisker densities of 63 whiskers/mm^2 after only 30 days [50], reveals the non-aqueous coating to be superior to others, but not to all. This increased whisker propensity between the deposits produced from the aqueous and non-aqueous electrolytes may be due to several deposit characteristics, such as: IMC growth distribution and morphology; deposit crystallographic orientation; tin grain size and grain shape. Although it is tempting to relate the differences in IMC growth to the different propensity of the aqueous and non-aqueous deposits to form whiskers, attempting to relate individual deposit characteristics to whisker growth is problematic since many other characteristics also change.

The IMC distribution observed for deposits produced from the non-aqueous Propeline electrolyte seen within this project were similar to that reported in the literature with wedge shaped IMCs being evident [191]. In comparison, the coatings produced from Tinmac showed different IMC morphologies and distributions, and therefore were studied in greater detail. IMC growth was shown to be rapid and extensive coverage was observed within 55 days. But as the Tinmac electrodeposit demonstrates a low whisker propensity at the thicknesses studied for IMC growth [59]. This reinforces that it is not just the amount of IMCs that is important, but also the distribution and shape of the IMC and the tin coating’s grain structure. Although the coating’s structure with respect to thickness was not studied here, the overall IMC distribution has been seen to be akin to that of other tin coatings on roughened substrates which are reported to show decreased whisker growth [82]. Thus, the IMC distribution from the Tinmac coating is deemed to be beneficial and may be at least partly responsible for slow tin whisker growth. Furthermore, if this is the case, this coating forms an IMC distribution at room temperature that demonstrates a reduced propensity to induce whisker growth even when electrodeposited upon electropolished copper (a very smooth

substrate surface), which is deemed to be a problematic substrate state in terms of IMC distribution [82].

1 and 2 µm deposits created from the aqueous electrolyte showed differences in, surface finish, surface topography and grain structure. These differences within the deposits were accentuated at 10 µm (thicknesses used for electrodeposit characterisation using standard thicknesses) resulting in highly uniform bright deposits, with a crystallographic orientation preferentially located towards <001> with respect to the surface normal. Furthermore, the low whisker propensities of these deposits may be due to this uniform crystallographic orientation, or many other combined deposit characteristics.

Although when using Tinmac we were unable to create bright deposits at low deposit thicknesses those produced from Propeline with 0.71 M $SnCl_2.2H_2O$ at 0.5 mA/cm^2 were bright. This process was able to create a bright coating with thicknesses measured in tens of nanometres, and although these thicknesses may be of little use in the electronics industry with respect to solder joints it may be of use in other industries, such as nanowire coating [262] and RF transparent coatings [263], which can utilise a nanoscale tin coating with high uniformity, bright surface finish and high substrate coverage.

Electrodeposition from DESs is still in its infancy, with only a small number of metals being examined, and even fewer of those electrodeposits forming uniformly. The research within this thesis has shown that by selecting appropriate deposition variables and electrolyte formulations it is possible to achieve a uniform tin electrodeposit. Moreover, the importance of this development is emphasised when put into context of the eventual creation of novel tin alloys using DESs. As such, this current project is part of the initial steps towards novel tin alloy electrodeposition in the hope to eventually find a highly whisker mitigating deposit.

Chapter 6 Conclusions

6.1 Electrodeposition from an aqueous electrolyte

Tin coatings electrodeposited from an aqueous proprietary bath (Tinmac) showed that bright deposits were created from a range of current densities; between 10 mA/cm^2 to 50 mA/cm^2 with agitation and between 10 mA/cm^2 to 30 mA/cm^2 without agitation. Moreover, electrodeposits features crystallographic orientations towards the <001 > direction, with respect to the surface normal), regardless of current density or electrolyte agitation. Furthermore, grain size and shape varied dependent upon current density and electrolyte agitation with electrodeposits produced at low current densities containing grain with faceted grain boundaries possibly increasing whisker growth experienced in other studies.

6.2 IMC growth for electrodeposit produced from an aqueous electrolyte

Electrodeposited bright tin coatings were produced on electropolished copper substrate material using a proprietary aqueous electrolyte (Tinmac). Electrochemical stripping of the tin allowed examination of copper-tin IMCs formed after a variety of storage periods subsequent to electroplating. The following findings were made:

- IMC growth after 24 h of storage was dependent upon specific copper grain orientations, grain boundaries and to a lesser degree tin electrodeposit grain boundaries, highlighting the importance of substrate texture as opposed to coating structure.
- IMC growth after 55 days of storage resulted in almost complete interface coverage. It is thought that the growth of a uniform layer of IMCs is, at least in part, responsible for the low level of whisker growth observed for this tin coating system.
- IMC growth due to a heat treatment at 150°C for 30 mins accelerated interface coverage with a seven day period and affected IMC particle morphology. Comparatively, without the heat treatment IMC growth was low but with the addition of a heat treatment the IMC completely covered the interface.

6.3 Anodic electrochemical tin stripping

Electrodeposited bright tin coatings were produced on electropolished copper substrate material. Electrochemical stripping of the tin allowed examination of copper-tin IMCs formed after a variety of storage periods subsequent to electroplating. Complete anodic electrochemical removal of tin electrodeposits using 5 % H_2SO_4 was sufficiently successful leaving the copper substrates and IMCs unaffected allowing detailed analysis by SEM.

Initially this technique utilised solely, either a potentiostatic oxidation which caused sulphate formation inhibiting analysis, or galvanostatic stripping which did not remove all remaining tin. Therefore, this technique was further developed by the removal of the tin coatings by galvanostatic followed by potentiostatic oxidation. This later improvement provided minimal sulphate build up, undamaged IMC, a rapid and regulated means of tin stripping, yet still providing high quality results.

6.4 Non-aqueous electrolyte DC voltammetry for tin electrodeposition

By undertaking cyclic linear polarisation of Reline, Ethaline and Propeline at 70 ^oC with additions of $SnCl_2$ it has been shown that:

- Increasing additions of $SnCl_2$ increased currents at given overpotentials and additionally decreased the reduction of the electrolyte.
- Anodic potentials above that of the onset of tin oxidation produced sufficient current to suggest tin to be a suitable anode material during electrodeposition to minimise unwanted oxidation products, such as the degradation of the electrolyte producing hydrogen gas.

6.5 Comparison of electrodeposits from aqueous and non-aqueous electrolytes

By comparing electrodeposits created from Propeline (0.044 M $SnCl_2$ at 10 mA/cm^2) with those produced from a commercial aqueous bright tin electroplating bath it was seen that:

- Deposits created from Propeline, at thicknesses above 1 µm produced needle-like dendrites; reducing deposit uniformity.

- Uniform 1 µm deposits from both electrolytes demonstrated topographical and crystallographic variations dependent upon underlying substrate grain orientation and deposit thickness.
- Deposit IMC growth and whisker propensity were dependent upon electrolyte formulation, with the deposits formed from the non-aqueous electrolyte producing 0.64 whiskers/mm^2, whilst deposits prepared from the aqueous bath showed no whisker growth over the same time period.

6.6 Whisker measurement programme

The Programme (3DView) created during this project is able to, through a user's interactions with two differing tilted micrographs, create measurements in Cartesian coordinates. This programme has been tested showing itself to create reproducible and accurate measurements of a virtual model, measurements of flat features viewed from non-vertical orientations, and measurement of real bismuth whiskers. Moreover, this programme is able to measure multiple locations simultaneously, thus allowing it to measure multiple features within paired micrographs. This therefore allows multiple whiskers, or locations along the length of a whisker, to be measured simultaneously.

Chapter 7 Future work

Future work could be taken in many directions from this point, e.g. the development of novel deep eutectic solvents or the electrochemical characterisation of pre-existing ones. In the author's opinion, many of the promising pre-existing characterised DESs require experimental application to prove their practical value. Moreover, from this thesis there is a natural progression towards electrodepositing novel tin alloys. Therefore, two recommendations would be given for future work concerning electrodeposition using DESs: Firstly, testing the DESs used in this project, Ethaline, Reline and Propeline, with similar tin(II) concentrations and additional metal compounds. These additions may significantly alter electrodeposit formation and may form thick, uniform tin alloyed deposits. Secondly, alternative DESs could be tested to create a tin alloy deposit. Again, this has the possibility of producing thick uniform tin alloyed electrodeposits, but as deposit structure is unpredictable the knowledge concerning the already studied DESs makes the study of them preferable.

Additionally, future work concerning the standardisation of accelerated whisker growth and measurement of propensities for additional electrodeposits would be highly beneficial for the tin electroplating industry. The standards currently used for assessing whisker propensity utilise elevated or cycled temperature and humidity acceleration, shown to be limited in its ability. On the other hand, substrate bending within clamps has been shown affective in accelerating whisker growth but its accuracy or reproducibility is uncertain. Therefore recommendation to evaluate whisker acceleration by mechanical bending would be made. Furthermore, if induced stresses by substrate bending were shown to be a reproducible and accurate means of accelerated whisker growth, then this method may become a new standardised means of acceleration. Furthermore, by producing a reliable standardised and trusted method of whisker acceleration it would be hoped that more tin deposits would be eventually evaluated, leading to a greater informed tin electrodeposition market and reduced numbers of whisker failures.

Finally, future work concerning three dimensional measurement of whisker growth using the programme created within this project would be to release the code into the public domain to allow three-dimensional measurements within many areas of research.

Chapter 8 References

[1] L. I. Tushinsky, I. Kovensky, A. Plokov, V. Sindeyev, P. Reshedko, *Coated Metal: Structure and Properties of Metal-Coatings Compositions,* Springer, 2002.

[2] J. Levy, *Tin (Understanding the Elements of the Periodic Table),* Rosen Central, 2009.

[3] B. W. Darvell, *Materials Science for Dentistry*, Woodhead Publishing, 2009

[4] J. W. S. M. Datta, Tetsuya Osaka, *Microelectronic Packaging*. CRC Press, 2004.

[5] C. Hornbostel, *Construction Materials: Types, Uses and Applications*. John Wiley & Sons, 1991.

[6] A. C. Tan, *Tin and Solder Plating in the Semiconductor Industry*. Chapman and Hall, 1992.

[7] N. D. Burns, "A Tin Pest Failure," *J. Fail. Anal. Prev.*, vol. 9, no. 5, pp. 461–465, 2009.

[8] G. T. Galyon, "Annotated tin whisker bibliography and anthology," *IEEE Trans. Electron.Packag. Manuf.*, vol. 28, no. 1, pp. 94–122, 2005.

[9] R. C. Lasky, "Tin Pest: a Forgotten Issue in Lead-Free Soldering?," in *SMTA International*, 2004.

[10] D. H. Buckley, *Surface Effects in Adhesion, Friction, Wear and Lubrication*. Elsevier Science Ltd, 1981.

[11] J. P. Glusker, M. Lewis, and M. Rossi, *Crystal Structure Analysis for Chemists and Biologists*. Wiley-Blackwell, 1994.

[12] B. H. Chudnovsky, *Electrical Power Transmission and Distribution: Aging and Life Extension Techniques*. CRC Press Inc, 2012.

[13] K. G. Compton, A. Mendizza, and S. M. Arnold, "Filamentary Growths On Metal Surfaces – 'Whiskers,'" *Corrosion*, vol. 7, no. 10, pp. 327–334, 1951.

[14] E. R. Crandall, *Factors Governing Tin Whisker Growth,* Springer International Publishing, 2013.

[15] European Parliament and the Council of the European Union, "Directive 2012/19/EU of the European Parliment and of the Council on Waste Electrical and Electronic Equipment (WEEE)," *Off. J. Eur. Union*, vol. 55, pp. L197/38–L197/70, 2012.

[16] European Parliament and the Council of the European Union, "Directive 2002/95/EC: Restriction of the use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS)," *Off. J. Eur. Union*, vol. 46, pp. L37/19–L37/23, 2003.

[17] D. Shangguan, *Lead-Free Solder Interconnect Reliability*, ASM International, 2005.

[18] G. Langbein and W. T. Brannt, *A Complete Treatise on the Electro-Deposition of Metals*. Hodder & Stoughton, Limited, 1913.

[19] S. P. Thompson, *Dynamo-Electricity Machinery: A Manual for Students of Electrotechnics*. Cambridge University Press, 2011.

[20] C. H. Pitt and R. G. Henning, "Pressure-Induced Growth of Metal Whiskers," *J. Appl. Phys.*, vol. 35, no. 2, p. 459, 1964.

[21] California Department of Public Health, "Medical Guidelines for the Lead-Exposed Worker," 2009.

[22] D. J. Hoffman, B. A. Rattner, G. A. Burton, and John Cairns, *Handbook of Ecotoxicology, Second Edition*. CRC Press, 2002.

[23] J. F. Shackelford and R. H. Doremus, *Ceramic and Glass Materials: Structure, Properties and Processing*. Springer Science & Business Media, 2008.

[24] Murrietta Circuits, "Re-tinning Services." [Online]. Available: http://www.murrietta.com/retinning-services.html. [Accessed: 07-Jul-2014].

[25] ASTM International, "Designation: B545–97 Standard Specification for Electrodeposited Coatings of Tin," 2004.

[26] National Research Council (U.S.). Committee on Technical Aspects of Critical and Strategic Material. Panel on Tin, "Trends in the Use of Tin: Report," 1970.

[27] K. J. Puttlitz and K. A. Stalter, *Handbook of Lead-Free Solder Technology for Microelectronic Assemblies*. CRC Press, 2004.

[28] C. Y. Heong, A.S.M.A.Haseeb, G. Yingxin, and L. S. Fang, "Effects of Sn concentration and current density on Sn-Bi electrodeposition in additive free plating bath," in *4th Asia Symposium on Quality Electronic Design*, 2012.

[29] J. Duan and X. Kou, "Effect of Current Density on the Microstructure and Magnetic Properties of Electrodeposited Co_2FeSn Heusler Alloy," *J. Electrochem. Soc.*, vol. 160, no. 10, pp. D471–D475, 2013.

[30] A. M. Rashidi and A. Amadeh, "The effect of current density on the grain size of electrodeposited nanocrystalline nickel coatings," *Surf. Coatings Technol.*, vol. 202, no. 16, pp. 3772–3776, 2008.

[31] K. Haerens, E. Matthijs, K. Binnemans, and B. Van der Bruggen, "Electrochemical decomposition of choline chloride based ionic liquid analogues," *Green Chem.*, vol. 11, no. 9, pp. 1357–1365, 2009.

[32] D. R. Gabe, "The role of hydrogen in metal electrodeposition processes," *J. Appl. Electrochem.*, vol. 27, no. 8, pp. 908–915, 1997.

[33] D. Pletcher, *A First Course in Electrode Processes*, Second Edi. Royal Society of Chemistry, 2009.

[34] W. Plieth, *Electrochemistry for Materials Science*, First Edi., Elsevier Science, 2008.

[35] R. K. Pandey, S. N. Sahu, and S. Chandra, *Handbook of Semiconductor Electrodeposition*. CRC Press Inc, 1996.

[36] P. M. Shirage, D. D. Shivagan, R. S. Kalubarme, V. Ganesan, and S. H. Pawar, "The nucleation and growth mechanism of the electrodeposition of $Tl_2Ba_2Ca_2Cu_3O_{10}$ superconducting thin films on Al-substrate," *Supercond. Sci. Technol.*, vol. 21, no. 6, pp. 065009.1–065009.6, Jun. 2008.

[37] D. Grujicic and B. Pesic, "Electrodeposition of copper: the nucleation mechanisms," *Electrochim. Acta*, vol. 47, no. 18, pp. 2901–2912, 2002.

[38] R. Krumm, B. Guel, C. Schmitz, and G. Staikov, "Nucleation and growth in electrodeposition of metals on n-Si(111)," *Electrochim. Acta*, vol. 45, no. 20, pp. 3255–3262, 2000.

[39] K. Kondo, *Morphological Evolution of Electrodeposits and Electrochemical Processing in ULSI Fabrication and Electrodeposition of and on Semiconductors IV: Proceedings of the International Symposia*. The Electrochemical Society, 2005.

[40] J. Jovicevic and A. Bewick, "A study of the initial stages of the electrochemical deposition of metals on foreign substrates: Lead and thallium on copper and silver surfaces: General discussion," *Facta Univ. - Ser. Physics, Chem. Technol.*, vol. 3, no. 2, pp. 183–203, 2005.

[41] D. Wolf and S. Yip, *Materials Interfaces: Atomic-level Structure and Properties*. Springer Science & Business Media, 1992.

[42] Y. D. Gamburg and G. Zangari, *Theory and Practice of Metal Electrodeposition*. New York, NY: Springer New York, 2011.

[43] H. Wu, G. Zhao, J. Mu, X. Li, and Y. He, "Effects of ultrasonic dispersion on structure of electrodeposited Ni coating on AZ91D magnesium alloy," *Trans. Nonferrous Met. Soc. China*, vol. 20, pp. s703–s707, 2010.

[44] J. Torrent-Burgués and E. Guaus, "Electrodeposition of Tin from Tartrate Solutions," *Port. Electrochim. Acta*, vol. 23, no. 4, pp. 471–479, 2005.

[45] K.-W. Moon, S.-K. Kim, M. E. Williams, W. J. Boettinger, and G. R. Stafford, "Effect of current density and electrolyte concentration on hillock growth from pure bright Sn electrodeposits," *J. Appl. Electrochem.*, vol. 40, no. 9, pp. 1671–1681, 2010.

[46] M. A. Ashworth, G. D. Wilcox, R. L. Higginson, R. J. Heath, and C. Liu, "Effect of direct current and pulse plating parameters on tin whisker growth from tin electrodeposits on copper and brass substrates," *Trans. Inst. Met. Finish.*, vol. 91, no. 5, pp. 260–268, 2013.

[47] U. Sahaym, S. L. Miller, and M. G. Norton, "Effect of plating temperature on Sn surface morphology," *Mater. Lett.*, vol. 64, no. 14, pp. 1547–1550, 2010.

[48] W. Zhang, J. Guebey, and M. Toben, “A novel electrolyte for the high speed electrodeposition of bright pure tin at elevated temperatures,” *Met. Finish.*, vol. 109, no. 1–2, pp. 13–19, 2011.

[49] A. Sharma, S. Bhattacharya, R. Sen, B. S. B. Reddy, H.-J. Fecht, K. Das, and S. Das, “Influence of current density on microstructure of pulse electrodeposited tin coatings,” *Mater. Charact.*, vol. 68, pp. 22–32, 2012.

[50] B.-Z. Lee and D. N. Lee, “Spontaneous growth mechanism of tin whiskers,” *Acta Mater.*, vol. 46, no. 10, pp. 3701–3714, 1998.

[51] G. Trejo, H. Ruiz, R. O. Borges, and Y. Meas, “Influence of polyethoxylated additives on zinc electrodeposition from acidic solutions,” *J. Appl. Electrochem.*, vol. 31, no. 6, pp. 685–692, 2001.

[52] S. Shanmugasigamani and M. Pushpavanam, “Voltammetric Studies on the Role of Additives in Bright Zinc Electrodeposition from an Alkaline Non-Cyanide Bath,” *Port. Electrochim. Acta*, vol. 27, no. 6, pp. 725–735, 2009.

[53] J. L. Ortiz-Aparicio, Y. Meas, G. Trejo, R. Ortega, T. W. Chapman, and E. Chainet, “Effects of organic additives on zinc electrodeposition from alkaline electrolytes,” *J. Appl. Electrochem.*, vol. 43, no. 3, pp. 289–300, 2012.

[54] N. M. Pereira, S. Salomé, C. M. Pereira, and A. Fernando Silva, “Zn–Sn electrodeposition from deep eutectic solvents containing EDTA, HEDTA, and Idranal VII,” *J. Appl. Electrochem.*, vol. 42, no. 8, pp. 561–571, 2012.

[55] F. Xiao, X. Shen, F. Ren, and A. A. Volinsky, “Additive effects on tin electrodepositing in acid sulfate electrolytes,” *Int. J. Miner. Metall. Mater.*, vol. 20, no. 5, pp. 472–478, 2013.

[56] L. Oniciu and L. Mureşan, “Some fundamental aspects of levelling and brightening in metal electrodeposition,” *J. Appl. Electrochem.*, vol. 21, no. 7, pp. 565–574, 1991.

[57] K. Tsuji, “Roll of Grain-boundary Free Energy & Surface Free Energy for Tin Whisker Growth,” in *IPC/JEDEC 10th Int. Conf. Lead Free Electron. Compon. Assemblies*, 2005.

[58] F. Zhao, Q. Wang, and T. Lee, “Characteristics of Tin Whisker Growth on Matte Tin Surface Finishes by Acceleration Conditions,” in *8th International Conference on Electronic Packaging Technology*, 2007.

[59] M. A. Ashworth, “WHISKERMIT presentation,” *Unpubl. Manuscr.*, 2011.

[60] R. L. Broggi, G. M. de Oliveira, L. L. Barbosa, E. M. J. A. Pallone, and I. A. Carlos, “Study of an alkaline bath for tin deposition in the presence of sorbitol and physical and morphological characterization of tin film,” *J. Appl. Electrochem.*, vol. 36, no. 4, pp. 403–409, 2005.

[61] C. Han, Q. Liu, and D. G. Ivey, “Nucleation of Sn and Sn–Cu alloys on Pt during electrodeposition from Sn–citrate and Sn–Cu–citrate solutions,” *Electrochim. Acta*, vol. 54, no. 12, pp. 3419–3427, 2009.

[62] C. T. J. Low and F. C. Walsh, "The influence of a perfluorinated cationic surfactant on the electrodeposition of tin from a methanesulfonic acid bath," *J. Electroanal. Chem.*, vol. 615, no. 2, pp. 91–102, 2008.

[63] N. M. Martyak and R. Seefeldt, "Additive-effects during plating in acid tin methanesulfonate electrolytes," *Electrochim. Acta*, vol. 49, no. 25, pp. 4303–4311, 2004.

[64] J. Wu, G. Zhai, Q. Chen, J. Wang, and G. Ren, "The interfacial structure of plated copper alloy resistance spot welded joint," *Appl. Surf. Sci.*, vol. 254, no. 22, pp. 7227–7231, 2008.

[65] M. García-Gabaldón, V. Pérez-Herranz, E. Sánchez, and S. Mestre, "Effect of tin concentration on the electrical properties of ceramic membranes used as separators in electrochemical reactors," *J. Memb. Sci.*, vol. 323, no. 1, pp. 213–220, 2008.

[66] M.-S. Suh, C.-J. Park, and H.-S. Kwon, "Effects of plating parameters on alloy composition and microstructure of Sn–Bi electrodeposits from methane sulphonate bath," *Surf. Coatings Technol.*, vol. 200, no. 11, pp. 3527–3532, 2006.

[67] O. Roussak and H. D. Gesser, *Applied Chemistry: A Textbook for Engineers and Technologists*. Springer Science & Business Media, 2012.

[68] E. Gammon, *General Chemistry, Ninth Edition*. Houghton Mifflin Company, 2007.

[69] G. S. Tzeng, S. H. Lin, Y. Y. Wang, and C. C. Wan, "Effects of additives on the electrodeposition of tin from an acidic Sn(II) bath," *J. Appl. Electrochem.*, vol. 26, no. 4, pp. 419–423, 1996.

[70] I. A. Carlos, E. D. Bidoia, E. M. J. A. Pallone, M. R. H. Almeida, and C. A. C. Souza, "Effect of tartrate content on aging and deposition condition of copper–tin electrodeposits from a non-cyanide acid bath," *Surf. Coatings Technol.*, vol. 157, no. 1, pp. 14–18, 2002.

[71] G. P. Rockwell and J. R. Dahn, "Sn-based roughness gradients for high-throughput screening," *Thin Solid Films*, vol. 516, no. 21, pp. 7361–7365, 2008.

[72] B. Horváth, "Influence of copper diffusion on the shape of whiskers grown on bright tin layers," *Microelectron. Reliab.*, vol. 53, no. 7, pp. 1009–1020, 2013.

[73] M. Charrouf, S. Bakkali, M. Cherkaoui, and M. Amrani, "Influence of decyl glucoside on the electrodeposition of tin," *J. Serbian Chem. Soc.*, vol. 71, no. 6, pp. 661–668, 2006.

[74] Y. Wang, J. He, W. Wang, M. Naotoshi, and Z. Chen, "Sustained Immersion Tin Deposition on Copper from Choline Chloride Based Aqueous Solution without Reducing Agent," *J. Electrochem. Soc.*, vol. 160, no. 8, pp. D295–D299, 2013.

[75] Y. Zhang and J. A. Abys, "A unique electroplating tin chemistry," *Circuit World*, vol. 25, no. 1, pp. 30–37, 1999.

[76] B. Jiang and A.-P. Xian, "Whisker growth on tin finishes of different electrolytes," *Microelectron. Reliab.*, vol. 48, no. 1, pp. 105–110, 2008.

[77] T. Shibutani, "Effect of Grain Size on Pressure-Induced Tin Whisker Formation," *IEEE Trans. Electron. Packag. Manuf.*, vol. 33, no. 3, pp. 177–182, 2010.

[78] T. Kakeshita, K. Shimizu, R. Kawanaka, and T. Hasegawa, "Grain size effect of electro-plated tin coatings on whisker growth," *Journal of Materials Science*, vol. 17, no. 9, pp. 2560–2566, 1982.

[79] N. M. Pereira, C. M. Pereira, and A. F. Silva, "The Effect of Complex Agents on the Electrodeposition of Tin from Deep Eutectic Solvents," *ECS Electrochem. Lett.*, vol. 1, no. 2, pp. D5–D7, 2012.

[80] C.-F. Yu, C.-M. Chan, and K.-C. Hsieh, "The effect of tin grain structure on whisker growth," *Microelectron. Reliab.*, vol. 50, no. 8, pp. 1146–1151, 2010.

[81] J. Smetana, "Theory of Tin Whisker Growth: 'The End Game,'" *IEEE Trans. Electron. Packag. Manuf.*, vol. 30, no. 1, pp. 11–22, 2007.

[82] G. Milad, "The elimination of whiskers from electroplated tin," *Circuit World*, vol. 37, no. 4, pp. 10–15, 2011.

[83] W. Zhang and F. Schwager, "Effects of Lead on Tin Whisker Elimination," *J. Electrochem. Soc.*, vol. 153, no. 5, pp. C337–C343, 2006.

[84] G. Grossmann and C. Zardini, *The ELFNET Book on Failure Mechanisms, Testing Methods, and Quality Issues of Lead-Free Solder Interconnects*. Springer London, 2011.

[85] A. Dimitrovska and R. Kovacevic, "The Effect of Micro-Alloying of Sn Plating on Mitigation of Sn Whisker Growth," *J. Electron. Mater.*, vol. 38, no. 12, pp. 2726–2734, 2009.

[86] M. Takamizawa, T. Naka, M. Hino, K. Murakami, Y. Mitooka, and K. Nakai, "Effect of Lead Co-Deposition on the Whisker Growth on Electrodeposited Tin Film," *J. Japan Inst. Met.*, vol. 72, no. 3, pp. 229–235, 2008.

[87] J. D. Watkins, C. E. Hotchen, J. M. Mitchels, and F. Marken, "Decamethylferrocene Redox Chemistry and Gold Nanowire Electrodeposition at Salt Crystal|Electrode|Nonpolar Organic Solvent Contacts," *Organometallics*, vol. 31, no. 7, pp. 2616–2620, 2012.

[88] R. Lines, V. D. Parker, J. Sandström, B.-L. Chang, F. S. El-Feraly, J.-E. Berg, and A.-M. Pilotti, "Voltammetry in Benzene and Chlorobenzene. The Behaviour of Ions of Aromatic Compounds in Nonpolar Media," *Acta Chem. Scand. B*, vol. 31, pp. 369–374, 1977.

[89] W. H. Kruesi and D. J. Fray, "The electrowinning of lithium from chloride-carbonate melts," *Metall. Trans. B*, vol. 24, no. 4, pp. 605–615, 1993.

[90] G. Chen, D. Fray, and T. Farthing, "Direct electrochemical reduction of titanium dioxide to titanium in molten calcium chloride," *Nature*, vol. 407, no. 6802, pp. 361–614, 2000.

[91] F. Endres, A. P. Abbott, and D. R. MacFarlane, Eds., *Electrodeposition from Ionic Liquids*. WILEY-VCH Verlag GmbH & Co. KGaA, 2008.

[92] M. Salanne, L. J. A. Siqueira, A. P. Seitsonen, P. A. Madden, and B. Kirchner, “From molten salts to room temperature ionic liquids: Simulation studies on chloroaluminate systems,” *Faraday Discuss.*, vol. 154, p. 171, 2012.

[93] J. P. Hallett and T. Welton, “Room-temperature ionic liquids: solvents for synthesis and catalysis,” *Chem. Rev.*, vol. 111, no. 5, pp. 2071–2083, 2011.

[94] A. P. M. Tavares, O. Rodríguez, and E. A. Macedo, “Ionic Liquids - New Aspects for the Future - New Generations of Ionic Liquids Applied to Enzymatic Biocatalysis,” J. Kadokawa, Ed. InTech, 2013.

[95] P. Walden, “Molecular weights and electrical conductivity of several fused salts,” *Bull. Russ. Acad. Sci.*, vol. 1800, pp. 405–422, 1914.

[96] S. P. M. Ventura, C. S. Marques, A. a Rosatella, C. a M. Afonso, F. Gonçalves, and J. a P. Coutinho, “Toxicity assessment of various ionic liquid families towards Vibrio fischeri marine bacteria.,” *Ecotoxicol. Environ. Saf.*, vol. 76, no. 2, pp. 162–168, 2012.

[97] F. Endres and S. Zein El Abedin, “Air and water stable ionic liquids in physical chemistry.,” *Phys. Chem. Chem. Phys.*, vol. 8, no. 18, pp. 2101–16, 2006.

[98] J. Gorke, F. Srienc, and R. Kazlauskas, “Toward advanced ionic liquids. Polar, enzyme-friendly solvents for biocatalysis,” *Biotechnol. Bioprocess Eng.*, vol. 15, no. 1, pp. 40–53, 2010.

[99] A. P. Abbott, J. C. Barron, K. S. Ryder, and D. Wilson, “Eutectic-Based Ionic Liquids with Metal-Containing Anions and Cations.,” *Chemistry (Weinheim an der Bergstrasse, Germany)*, vol. 13, no. 22, pp. 6495–501, 2007.

[100] Q. Zhang, K. De Oliveira Vigier, S. Royer, and F. Jérôme, “Deep eutectic solvents: syntheses, properties and applications.,” *Chem. Soc. Rev.*, vol. 41, no. 21, pp. 7108–46, 2012.

[101] Y. Dai, J. van Spronsen, G.-J. Witkamp, R. Verpoorte, and Y. H. Choi, “Natural deep eutectic solvents as new potential media for green technology.,” *Anal. Chim. Acta*, vol. 766, pp. 61–8, 2013.

[102] Y. Zhang, X. H. Lu, X. Feng, Y. J. Shi, and X. Y. Ji, “Properties and Applications of Choline-Based Deep Eutectic Solvents,” *Prog. Chem.*, vol. 25, no. 6, pp. 881–892, 2013.

[103] R. D. Rogers, K. R. Seddon, and S. Volkov, *Green Industrial Applications of Ionic Liquids*. Springer Science & Business Media, 2003.

[104] A. Kitada, Y. Kang, Y. Uchimoto, and K. Murase, “Room-Temperature Electrodeposition of Mg Metal from Amide Salts Dissolved in Glyme-Ionic Liquid Mixture,” *J. Electrochem. Soc.*, vol. 161, no. 3, pp. D102–D106, 2013.

[105] T. Jiang, M. J. Chollier Brym, G. Dubé, A. Lasia, and G. M. Brisard, “Electrodeposition of aluminium from ionic liquids: Part II - studies on the electrodeposition of aluminum from aluminum chloride ($AlCl_3$) - trimethylphenylammonium chloride (TMPAC) ionic liquids,” *Surf. Coatings Technol.*, vol. 201, no. 1–2, pp. 10–18, 2006.

[106] T. Jiang, M. J. Chollier Brym, G. Dubé, A. Lasia, and G. M. Brisard, “Electrodeposition of aluminium from ionic liquids: Part I—electrodeposition and surface morphology of aluminium from aluminium chloride ($AlCl_3$)–1-ethyl-3-methylimidazolium chloride ([EMIm]Cl) ionic liquids,” *Surf. Coatings Technol.*, vol. 201, no. 1–2, pp. 1–9, 2006.

[107] L. Bahadori, M. H. Chakrabarti, F. S. Mjalli, I. M. Al-Nashef, N. S. A. Manan, and M. A. Hashim, “Physicochemical properties of ammonium-based deep eutectic solvents and their electrochemical evaluation using organometallic reference redox systems,” *Electrochim. Acta*, vol. 113, pp. 205–211, 2013.

[108] D. L. Boxall, J. J. O’Dea, and R. A. Osteryoung, “Apparent Anomaly during Rotating Disk Voltammetry in Ionic Liquids,” *J. Electrochem. Soc.*, vol. 149, no. 11, pp. E468–E471, 2002.

[109] P.-Y. Chen and C. L. Hussey, “Electrodeposition of cesium at mercury electrodes in the tri-1-butylmethylammonium bis((trifluoromethyl)sulfonyl)imide room-temperature ionic liquid,” *Electrochim. Acta*, vol. 49, no. 28, pp. 5125–5138, 2004.

[110] M. Yamagata, Y. Katayama, and T. Miura, “Electrochemical Behavior of Samarium, Europium, and Ytterbium in Hydrophobic Room-Temperature Molten Salt Systems,” *J. Electrochem. Soc.*, vol. 153, no. 1, p. E5, 2006.

[111] Scionix, “Ionic Liquid Technology from Scionix” [Online]. Available: www.scionix.co.uk [Accessed: 14-Mar-2015].

[112] Thermo Fisher Scientific, “Water, extra pure, deionised – Acros Organics” [Online]. Available: www.fisher.co.uk [Accessed: 14-Mar-2015].

[113] N. M. Pereira, P. M. V. Fernandes, C. M. Pereira, and A. Fernando Silva, “Electrodeposition of Zinc from Choline Chloride-Ethylene Glycol Deep Eutectic Solvent: Effect of the Tartrate Ion,” *J. Electrochem. Soc.*, vol. 159, no. 9, pp. D501–D506, 2012.

[114] H. Lou, Yinlun Huang, and H. H. Lou, “Encyclopedia of Chemical Processing - Electroplating,” Taylor & Francis, 2007.

[115] A. Abbott, “Deep Eutectic Solvents,” in *Leuven Summer School on Ionic Liquids*, 2010.

[116] X.-H. Xu, “The Electrochemistry of Tin in the Aluminum Chloride-1-methyl-3-ethylimidazolium Chloride Molten Salt,” *J. Electrochem. Soc.*, vol. 140, no. 3, pp. 618–626, 1993.

[117] J.-F. Huang and I.-W. Sun, “Electrochemical Studies of Tin in Zinc Chloride-1-ethyl-3-methylimidazolium Chloride Ionic Liquids,” *J. Electrochem. Soc.*, vol. 150, no. 6, pp. E299–E306, 2003.

[118] M. Morimitsu, Y. Nakahara, Y. Iwaki, and M. Matsunaga, “Electrodeposition of tin from EMI· BF_4· Cl room temperature molten salts,” *J. Min. Metall. Sect. B Metall.*, vol. 39, no. 1–2, pp. 59–67, 2003.

[119] N. Tachikawa, N. Serizawa, Y. Katayama, and T. Miura, “Electrochemistry of Sn(II)/Sn in a hydrophobic room-temperature ionic liquid,” *Electrochim. Acta*, vol. 53, no. 22, pp. 6530–6534, 2008.

[120] T.-I. Leong, Y.-T. Hsieh, and I.-W. Sun, "Electrochemistry of tin in the 1-ethyl-3-methylimidazolium dicyanamide room temperature ionic liquid," *Electrochim. Acta*, vol. 56, no. 11, pp. 3941–3946, 2011.

[121] M.-J. Deng, J.-K. Chang, T.-I. Leong, S.-W. Fang, P.-Y. Chen, and I.-W. Sun, "Electrodeposition of Nanostructured Sn in 1-ethyl-3-methylimidazolium Dicyanamide Room Temperature Ionic Liquid," *Electrochemistry*, vol. 77, no. 8, pp. 588–590, 2009.

[122] H. M. Maltanava, T. N. Vorobyova, and O. N. Vrublevskaya, "Electrodeposition of tin coatings from ethylene glycol and propylene glycol electrolytes," *Surf. Coatings Technol.*, vol. 254, pp. 388–397, 2014.

[123] D. Zhu, H. Sun, and Y. Fung, "Electrodeposition of Tin from 1-Methyl-3-ethylimidazolium Chloride/$AlCl_3$/$SnCl_2$ Room Temperature Molten Salts," in *ECS Transactions*, 2007.

[124] Y. K. Kee, W. J. Basirun, and K. H. Lee, "Electrodeposition of Tin Using Tin(II) Methanesulfonate from a Mixture of Ionic Liquid and Methane Sulfonic Acid," *Adv. Mater. Res.*, vol. 264–265, pp. 1462–1467, 2011.

[125] W. Yang, H. Cang, Y. Tang, J. Wang, and Y. Shi, "Electrodeposition of tin and antimony in 1-ethyl- 3-methylimidazolium tetrafluoroborate ionic liquid," *J. Appl. Electrochem.*, vol. 38, no. 4, pp. 537–542, 2007.

[126] M.-J. Deng, T.-I. Leong, I.-W. Sun, P.-Y. Chen, J.-K. Chang, and W.-T. Tsai, "Fabrication of Porous Tin by Template-Free Electrodeposition of Tin Nanowires from an Ionic Liquid," *Electrochem. Solid-State Lett.*, vol. 11, no. 11, pp. D85–D88, 2008.

[127] A. P. Abbott, G. Capper, K. J. McKenzie, and K. S. Ryder, "Electrodeposition of Zinc–Tin Alloys from Deep Eutectic Solvents Based on Choline Chloride," *Journal of Electroanalytical Chemistry*, vol. 599, no. 2, pp. 288–294, 2007.

[128] L. Anicai, A. Petica, S. Costovici, P. Prioteasa, and T. Visan, "Electrodeposition of Sn and NiSn alloys coatings using choline chloride based ionic liquids—Evaluation of corrosion behavior," *Electrochim. Acta*, vol. 114, pp. 868–877, 2013.

[129] S. Salomé, N. M. Pereira, E. S. Ferreira, C. M. Pereira, and A. F. Silva, "Tin electrodeposition from choline chloride based solvent: Influence of the hydrogen bond donors," *J. Electroanal. Chem.*, vol. 703, pp. 80–87, 2013.

[130] C. D. Gu, Y. J. Mai, J. P. Zhou, Y. H. You, and J. P. Tu, "Non-aqueous electrodeposition of porous tin-based film as an anode for lithium-ion battery," *J. Power Sources*, vol. 214, pp. 200–207, 2012.

[131] C. D'Agostino, R. C. Harris, A. P. Abbott, L. F. Gladden, and M. D. Mantle, "Molecular motion and ion diffusion in choline chloride based deep eutectic solvents studied by 1H pulsed field gradient NMR spectroscopy.," *Phys. Chem. Chem. Phys.*, vol. 13, no. 48, pp. 21383–91, 2011.

[132] R. D. Hilty and N. Corman, "Tin Whisker Reliability Assessment by Monte Carlo Simulation," in *IPC/JEDEC Lead-Free Symposium*, 2005.

[133] T. R. Bieler, L. P. Lehman, T. Kirkpatrick, E. J. Cotts, and B. Nandagopal, “Influence of Sn Grain Size and Orientation on the Thermomechanical Response and Reliability of Pb-free Solder Joints,” *IEEE Trans. Components Packag. Technol.*, vol. 31, no. 2, pp. 370–381, 2008.

[134] S. Han, “Assessment of an Electrical Shorting and Metal Vapor Arcing Potential of Tin Whiskers,” Deparment of Mechanical Engineering, University of Maryland, 2012.

[135] K. J. Courey, S. S. Asfour, A. Onar, J. A. Bayliss, L. L. Ludwig, and M. C. Wright, “Tin Whisker Electrical Short Circuit Characteristics—Part II,” *IEEE Trans. Electron. Packag. Manuf.*, vol. 32, no. 1, pp. 41–48, 2009.

[136] M. Osterman, “Arcing of Metal Whiskers,” in *7th International Symposium on Tin Whiskers*, 2013.

[137] C.-C. Hu, Y.-D. Tsai, C.-C. Lin, G.-L. Lee, S.-W. Chen, T.-C. Lee, and T.-C. Wen, “Anomalous growth of whisker-like bismuth–tin extrusions from tin-enriched tin–Bi deposits,” *J. Alloys Compd.*, vol. 472, no. 1–2, pp. 121–126, 2009.

[138] F. M. Reicha and P. B. Barna, “Growth of hillocks and whiskers in aluminium films,” *Thin Solid Films*, vol. 85, no. 3–4, p. 317, 1981.

[139] S. A. Stanley, C. Stuttle, A. J. Caruana, M. D. Cropper, and A. S. O. Walton, “An investigation of the growth of bismuth whiskers and nanowires during physical vapour deposition,” *J. Phys. D. Appl. Phys.*, vol. 45, no. 43, pp. 1–12, 2012.

[140] J. Kitterman and D. Winder, “Growth of cadmium whiskers in various atmospheres,” *J. Cryst. Growth*, vol. 13–14, pp. 198–201, 1972.

[141] J. Westwater, D. P. Gosain, K. Yamauchi, and S. Usui, “Nanoscale silicon whiskers formed by silane / gold reaction at 335C,” *Mater. Lett.*, vol. 24, pp. 109–112, 1995.

[142] S. S. Brenner, “Strength of Gold Whiskers,” *J. Appl. Phys.*, vol. 30, no. 2, p. 266, 1959.

[143] A. G. Starliper and H. Kenworthy, “Tungsten whiskers by vapor-phase growth,” *Electrodepos. Surf. Treat.*, vol. 2, no. 4, pp. 249–262, May 1974.

[144] R. V. Coleman and G. W. Sears, “Growth of zinc whiskers,” *Acta Metall.*, vol. 5, no. 3, pp. 131–136, 1957.

[145] J. Brusse and M. Sampson, “Zinc whiskers: hidden cause of equipment failure,” *IT Prof.*, vol. 6, no. 6, pp. 43–47, Nov. 2004.

[146] Wikipedia, “Whisker (metallurgy).” [Online]. Available: http://en.wikipedia.org/wiki/Whisker_(metallurgy). [Accessed: 14-Jul-2014].

[147] C. V. Krishnan, J. Chen, C. Burger, and B. Chu, “Polymer-assisted growth of molybdenum oxide whiskers via a sonochemical process,” *J. Phys. Chem. B*, vol. 110, no. 41, pp. 20182–20188, 2006.

[148] J. Drott, “Growth of silver sulphide whiskers,” *Acta Metall.*, vol. 8, pp. 19–22, 1960.

[149] S. Simov, N. Koparanova, G. Metchenov, and D. Genchev, “Growth and morphology of cadmium oxide whiskers,” *Journal of Materials Science*, vol. 18, no. 2, pp. 623–633, 1983.

[150] E. G. Bylander, “Vapor Deposition of Sodium Chloride Whiskers,” *J. Appl. Phys.*, vol. 35, no. 6, p. 1988, 1964.

[151] G. W. Sears, “Twist in Lithium Fluoride Whiskers,” *J. Chem. Phys.*, vol. 31, no. 1, p. 53, Aug. 1959.

[152] B. D. Dunn, “Mechanical and Electrical Characteristics of Tin Whiskers with Special Reference to Spacecraft Systems,” *ESA J.*, vol. 12, pp. 1–17, 1988.

[153] L. Panashchenko and M. Osterman, “Examination of nickel underlayer as a tin whisker mitigator,” in *59th Electronic Components and Technology Conference*, 2009.

[154] G. T. Galyon, “A History of Tin Whisker Theory: 1946 to 2004,” in *SMTA International*, 2004.

[155] Y. Fukuda, M. Osterman, and M. Pecht, “The impact of electrical current, mechanical bending, and thermal annealing on tin whisker growth,” *Microelectron. Reliab.*, vol. 47, no. 1, pp. 88–92, 2007.

[156] M. Sampson and H. Leidecker, “NASA Whisker Failures.” [Online]. Available: http://nepp.nasa.gov/whisker/failures/.

[157] Dept of Health Education and Welfare Public Health Service Food and Drug Administration, “Inspection Technical Guides - Tin Wiskers Problems, Causes, and Solutions.” [Online]. Available: www.fda.gov/ICECI/Inspections/InspectionGuides/InspectionTechnicalGuides/ucm072921.html. [Accessed: 10-Jul-2014].

[158] Satellite News Digest, “Satellite Outages and Failures.” [Online]. Available: http://www.sat-index.co.uk/failures/. [Accessed: 10-Jul-2014].

[159] M. Takeuchi, K. Kamiyama, and K. Suganuma, “Suppression of tin whisker formation on fine pitch connectors by surface roughening,” *J. Electron. Mater.*, vol. 35, no. 11, pp. 1918–1925, 2006.

[160] G. Davy, “Relay Failure Caused by Tin Whiskers,” *Northrop Grumman Electronic Systems*, 2002. [Online]. Available: http://nepp.nasa.gov/whisker/reference/tech_papers/davy2002-relay-failure-caused-by-tin-whiskers.pdf. [Accessed: 10-Jul-2014].

[161] Westinghouse Electric Company, “Technical Bulletin - Baslew power supplies.” [Online]. Available: http://nepp.nasa.gov/whisker/reference/tech_papers/2002-westinghouse-tb02005.pdf. [Accessed: 10-Jul-2014].

[162] Westinghouse Electrical Company, “Technical Bulletin - Potential Tin Whiskers on Printed Circuit Board Components.” [Online]. Available: http://nepp.nasa.gov/whisker/reference/tech_papers/2005-westinghouse-tb05004.pdf. [Accessed: 10-Jul-2014].

[163] P. Daddona, “Reactor Shutdown: Dominion Learns Big Lesson From A Tiny ‘tin Whisker,” *TheDay.com*, 2005. [Online]. Available: http://nepp.nasa.gov/whisker/reference/tech_papers/2005-dadonna-nuclear-reactor-shutdown.pdf. [Accessed: 10-Jul-2014].

[164] National Highway Traffic Safety Administration (USA), “Technical Assessment of Toyota Electronic Throttle Control (ETC) Systems,” 2011.

[165] National Aeronautics and Space Administration (USA), “National Highway Traffic Safety Administration Toyota Unintended Acceleration Investigation Investigation,” 2011.

[166] JEDEC, *JEDEC STANDARD -JESD201 Environmental Acceptance Requirements for Tin Whisker Susceptibility of Tin and Tin Alloy Surface Finishes*. 2006.

[167] JEDEC/IPC, *JP0021: JEDEC/IPC Current Tin Whiskers Theory and Mitigation Practices Guideline*. 2006.

[168] JEDEC, *JEDEC STANDARD - JESD22A121 - Measuring Whisker Growth on Tin and Tin Alloy Surface Finishes*, no. May. 2005.

[169] J. Eshelby, “A Tentative Theory of Metallic Whisker Growth,” *Phys. Rev.*, vol. 91, no. 3, pp. 755–756, 1953.

[170] Y. Mizuguchi, Y. Murakami, S. Tomiya, T. Asai, T. Kiga, and K. Suganuma, “Effect of Crystal Orientation on Mechanically Induced Sn Whiskers on Sn-Cu Plating,” *J. Electron. Mater.*, vol. 41, no. 7, pp. 1859–1867, 2012.

[171] P. Sarobol, W.-H. Chen, A. E. Pedigo, P. Su, J. E. Blendell, and C. A. Handwerker, “Effects of local grain misorientation and β-Sn elastic anisotropy on whisker and hillock formation,” *J. Mater. Res.*, vol. 28, no. 05, pp. 747–756, 2013.

[172] P. Sarobol, J. E. Blendell, and C. A. Handwerker, “Whisker and hillock growth via coupled localized Coble creep, grain boundary sliding, and shear induced grain boundary migration,” *Acta Mater.*, vol. 61, no. 6, pp. 1991–2003, 2013.

[173] K. Tu, “Irreversible processes of spontaneous whisker growth in bimetallic Cu-Sn thin-film reactions,” *Phys. Rev. B*, vol. 49, no. 3, pp. 2030–2034, 1994.

[174] E. Chason, N. Jadhav, F. Pei, E. Buchovecky, and A. Bower, “Growth of whiskers from Sn surfaces: Driving forces and growth mechanisms,” *Prog. Surf. Sci.*, vol. 88, no. 2, pp. 103–131, 2013.

[175] J.-H. Zhao, P. Su, M. Ding, S. Chopin, and P. S. Ho, “Microstructure-Based Stress Modeling of Tin Whisker Growth,” *IEEE Trans. Electron. Packag. Manuf.*, vol. 29, no. 4, pp. 265–273, 2006.

[176] T. Kato, H. Akahoshi, M. Nakamura, T. Terasaki, T. Iwasaki, T. Hashimoto, and A. Nishimura, “Correlation Between Whisker Initiation and Compressive Stress in Electrodeposited Tin–Copper Coating on Copper Leadframes,” *IEEE Trans. Electron. Packag. Manuf.*, vol. 33, no. 3, pp. 165–176, 2010.

[177] W. J. Boettinger, C. E. Johnson, L. A. Bendersky, K.-W. Moon, M. E. Williams, and G. R. Stafford, “Whisker and Hillock formation on Sn, Sn–Cu and Sn–Pb electrodeposits,” *Acta Mater.*, vol. 53, no. 19, pp. 5033–5050, 2005.

[178] M. Barsoum, E. Hoffman, R. Doherty, S. Gupta, and A. Zavaliangos, “Driving Force and Mechanism for Spontaneous Metal Whisker Formation,” *Phys. Rev. Lett.*, vol. 93, no. 20, pp. 206104.1–206104.4, 2004.

[179] M. Dittes, P. Oberndorff, P. Crema, and V. Schroeder, “The Effect of Temperature Cycling on Tin Whisker Formation,” in *Proc. on IPC / JEDEC 4th International Conference on Leadfree Electronics Assemblies and Components*, 2005.

[180] G. T. Galyon, C. Xu, S. Lai, B. Notohardjono, and L. Palmer, “The integrated theory of whisker formation - a stress analysis,” in *Proceedings Electronic Components and Technology*, 2005.

[181] P. Oberndorff, M. Dittes, P. Crema, P. Su, and E. Yu, “Humidity Effects on Sn Whisker Formation,” *IEEE Trans. Electron. Packag. Manuf.*, vol. 29, no. 4, pp. 239–245, 2006.

[182] R. M. Fisher, L. S. Darken, and K. G. Carroll, “Accelerated growth of tin whiskers,” *Acta Metall.*, vol. 2, no. 3, pp. 368–373, 1954.

[183] Y. Zhang, C. Fan, C. Xu, O. Khaselev, and J. A. Abys, “Tin whisker growth: substrate effect understanding CTE mismatch and IMC formation,” *CircuiTree*, 2004.

[184] W. J. Choi, T. Y. Lee, K. N. Tu, N. Tamura, R. S. Celestre, A. A. MacDowell, Y. Y. Bong, and L. Nguyen, “Tin whiskers studied by synchrotron radiation scanning X-ray micro-diffraction,” *Acta Mater.*, vol. 51, no. 20, pp. 6253–6261, 2003.

[185] C. L. Rodekohr, R. N. Dean, G. T. Flowers, R. L. Jackson, M. J. Bozack, and P. Lall, “Influence of Quantifiable Extrinsic Stresses on Tin Whisker Growth,” in *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2009.

[186] K. S. Kumar, L. Reinbold, A. F. Bower, and E. Chason, “Plastic deformation processes in Cu/Sn bimetallic films,” *J. Mater. Res.*, vol. 23, no. 11, pp. 2916–2934, 2011.

[187] C. Coston and N. H. Nachtrieb, “Self-Diffusion in Tin at High Pressure,” *J. Phys. Chem.*, vol. 68, no. 8, pp. 2219–2229, 1964.

[188] H. P. Howard, J. Cheng, P. T. Vianco, and J. C. M. Li, “Interface flow mechanism for tin whisker growth,” *Acta Mater.*, vol. 59, no. 5, pp. 1957–1963, 2011.

[189] K. N. Tu, “Interdiffusion and reaction in bimetallic Cu-Sn thin films,” *Acta Metall.*, vol. 21, no. 4, pp. 347–354, 1973.

[190] K. N. Tu and R. D. Thompson, “Kinetics of interfacial reaction in bimetallic Cu—Sn thin films,” *Acta Metall.*, vol. 30, pp. 947–952, 1982.

[191] A. Egli, F. Schwager, and N. Brown, "Investigation of Sn-Cu intermetallic compounds by AFM: new aspects of the role of intermetallic compounds in whisker formation," *IEEE Trans. Electron. Packag. Manuf.*, vol. 28, no. 1, pp. 85–93, 2005.

[192] G. T. Galyon and L. Palmer, "An integrated theory of whisker formation: the physical metallurgy of whisker formation and the role of internal stresses," *IEEE Trans. Electron. Packag. Manuf.*, vol. 28, no. 1, pp. 17–30, 2005.

[193] A. Baated, K.-S. Kim, and K. Suganuma, "Effect of intermetallic growth rate on spontaneous whisker growth from a tin coating on copper," *J. Mater. Sci. Mater. Electron.*, vol. 22, no. 11, pp. 1685–1693, 2011.

[194] A. Paul, "The Kirkendall Effect in Solid State Diffusion," Technische Universiteit Eindhoven, 2004.

[195] L. Sauter, A. Seekamp, Y. Shibata, Y. Kanameda, and H. Yamashita, "Whisker mitigation measures for Sn-plated Cu for different stress tests," *Microelectron. Reliab.*, vol. 50, no. 9–11, pp. 1631–1635, 2010.

[196] C. Kerr, "Sustainable technologies for the regeneration of acidic tin stripping solutions used in PCB fabrication," *Circuit World*, vol. 30, no. 3, pp. 51–58, 2004.

[197] W. Anders and J. Jokinen, "Method for the treatment and reuse of a stripper solution," WO2008125728 A223-Oct-2008.

[198] L. Reinbold, N. Jadhav, E. Chason, and K. Sharvan Kumar, "Relation of Sn whisker formation to intermetallic growth: Results from a novel Sn–Cu 'bimetal ledge specimen,'" *J. Mater. Res.*, vol. 24, no. 12, pp. 3583–3589, 2011.

[199] Y. Fukuda, M. Osterman, and M. Pecht, "The Effect of Annealing on Tin Whisker Growth," *IEEE Trans. Electron. Packag. Manuf.*, vol. 29, no. 4, pp. 252–258, 2006.

[200] B. Illes, T. Shinohara, B. Horvath, and G. Harsanyi, "Tin whisker growth from Sn-Cu (0–5 wt%) surface finishes," in *IEEE 17th International Symposium for Design and Technology in Electronic Packaging*, 2011.

[201] J. B. LeBret and M. G. Norton, "Electron microscopy study of tin whisker growth," *J. Mater. Res.*, vol. 18, no. 3, pp. 585–593, 2011.

[202] K. Whitlaw and J. Crosby, "An empirical study into whisker-growth of tin and tin alloy electrodeposits," in *2002 AESF SUR/FIN Conf.*, 2002.

[203] L. Panashchenko, "Evaluation of Environmental Tests for Tin Whisker Assessment," Mechanical Engineering, University of Maryland, 2009.

[204] Y. Fukuda, M. Osterman, and M. Pecht, "Length Distribution Analysis for Tin Whisker Growth," *IEEE Trans. Electron. Packag. Manuf.*, vol. 30, no. 1, pp. 36–40, 2007.

[205] E. Limpert, W. A. Stahel, and M. Abbt, "Log-normal Distributions across the Sciences : Keys and Clues," vol. 51, no. 5, pp. 341–352, 2001.

[206] S. Mathew, W. Wang, M. Osterman, and M. Pecht, "Assessment of Solder-Dipping as a Tin Whisker Mitigation Strategy," *IEEE Trans. Components, Packag. Manuf. Technol.*, vol. 1, no. 6, pp. 957–963, 2011.

[207] M. A. Ashworth, G. D. Wilcox, R. L. Higginson, R. J. Heath, and C. Liu, "An Investigation into Zinc Diffusion and Tin Whisker Growth for Electroplated Tin Deposits on Brass," *J. Electron. Mater.*, vol. 43, no. 4, pp. 1005–1016, 2014.

[208] C.-K. Lin and T.-H. Lin, "Effects of continuously applied stress on tin whisker growth," *Microelectron. Reliab.*, vol. 48, no. 10, pp. 1737–1740, 2008.

[209] E. Chason, N. Jadhav, W. L. Chan, L. Reinbold, and K. S. Kumar, "Whisker formation in Sn and Pb–Sn coatings: Role of intermetallic growth, stress evolution, and plastic deformation processes," *Appl. Phys. Lett.*, vol. 92, no. 17, p. 171901, 2008.

[210] D. Pinsky, M. Osterman, and S. Ganesan, "Tin Whiskering Risk Factors," *IEEE Trans. Components Packag. Technol.*, vol. 27, no. 2, pp. 427–431, 2004.

[211] T. Fang, M. Osterman, S. Mathew, and M. Pecht, "Tin whisker risk assessment," *Circuit World*, vol. 32, no. 3, pp. 25–29, 2006.

[212] M. Mason, G. Eng, M. Leung, G. Stupian, and T. Yeoh, "Microanalysis for tin whisker risk assessment," in *2011 International Reliability Physics Symposium*, 2011.

[213] T. Fang, S. Mathew, M. Osterman, and M. Pecht, "Assessment of risk resulting from unattached tin whisker bridging," *Circuit World*, vol. 33, no. 1, pp. 5–8, 2007.

[214] T. Fang, M. Osterman, S. Mathew, and M. Pecht, "Tin whisker risk assessment," *Circuit World*, vol. 32, no. 3, pp. 25–29, 2006.

[215] TechAmerica, "GEIA-STD-0005-2: Standard for Mitigating the Effects of Tin Whiskers in Aerospace and High Performance Electronic Systems," 2006.

[216] W. Fox and L. Woody, "Conformal coatings for tin whisker risk management," in *IPC APEX EXPO Technical Conference*, 2010.

[217] C. L. Rodekohr, G. T. Flowers, J. C. Suhling, and M. J. Bozack, "Influence of Substrate Surface Roughness on Tin Whisker Growth," in *Proceedings of the 54th IEEE Holm Conference on Electrical Contacts*, 2008.

[218] S. Canumalla and P. Viswanadham, *Portable Consumer Electronics: Packaging, Materials, and Reliability*. PennWell Books, 2010.

[219] W. Diyatmika, J. P. Chu, Y. W. Yen, W. Z. Chang, and C. H. Hsueh, "Thin film metallic glass as an underlayer for tin whisker mitigation: A room-temperature evaluation," *Thin Solid Films*, vol. 561, pp. 93–97, 2014.

[220] B. Illés, B. Horváth, and G. Harsányi, "Effect of strongly oxidizing environment on whisker growth form tin coating," *Surf. Coatings Technol.*, vol. 205, no. 7, pp. 2262–2266, 2010.

[221] European Parliament and the Council of the European Union, “Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH),” *Off. J. Eur. Union*, no. 7, pp. L136/3–L136/280, 2007.

[222] Y. Chervona, A. Arita, and M. Costa, “Carcinogenic metals and the epigenome: understanding the effect of nickel, arsenic, and chromium,” *Metallomics*, vol. 4, no. 7, pp. 619–627, 2012.

[223] H. M. El-Shafei, “Assessment of liver function among nickel-plating workers in Egypt.,” *Eastern Mediterranean health journal*, vol. 17, no. 6, pp. 490–494, 2011.

[224] K. K. Das, S. N. Das, and S. A. Dhundasi, “Nickel, its adverse health effects & oxidative stress,” *Indian J. Med. Res.*, vol. 128, no. 4, pp. 412–425, 2008.

[225] P. Apostoli and S. Catalani, “Metal ions affecting reproduction and development.,” *Met. Ions Life Sci.*, vol. 8, pp. 263–303, 2011.

[226] Z. Forgacs, P. Massányi, N. Lukac, and Z. Somosy, “Reproductive toxicology of nickel - review,” *J. Environ. Sci. Health. A. Tox. Hazard. Subst. Environ. Eng.*, vol. 47, no. 9, pp. 1249–1260, 2012.

[227] Public Health England, “Nickel - Toxicological Overview,” 2009.

[228] M. Dittes, P. Obemdorff, and L. Petit, “Tin Whisker formation - results, test methods and countermeasures,” in *53rd Electronic Components and Technology Conference Proceedings.*, 2003.

[229] K. Chen and G. Wilcox, “Observations of the Spontaneous Growth of Tin Whiskers on Tin-Manganese Alloy Electrodeposits,” *Phys. Rev. Lett.*, vol. 94, no. 6, pp. 066104/1–066104/4, 2005.

[230] M. Wickham, D. Lea, J. Nottay, and C. Hunt, “Processability of Lead-Free Component Termination Materials,” 2001.

[231] B. Jiang and A.-P. Xian, “Spontaneous growth of tin whiskers on tin−rare-earth alloys,” *Philos. Mag. Lett.*, vol. 87, no. 9, pp. 657–662, 2007.

[232] H. Hao, Y. Shi, Z. Xia, Y. Lei, and F. Guo, “Oxidization-Induced Tin Whisker Growth on the Surface of Sn-3.8Ag-0.7Cu-1.0Er Alloy,” *Metall. Mater. Trans. A*, vol. 40, no. 8, pp. 2016–2021, 2009.

[233] T. H. Chuang, H. J. Lin, and C. C. Chi, “Rapid growth of tin whiskers on the surface of Sn–6.6Lu alloy,” *Scr. Mater.*, vol. 56, no. 1, pp. 45–48, 2007.

[234] K. Whitlaw, A. Egli, and M. Toben, “Preventing whiskers in electrodeposited tin for semiconductor lead frame applications,” *Circuit World*, vol. 30, no. 2, pp. 20–24, 2004.

[235] W. Liu and M. Pecht, *IC Component Sockets*. John Wiley & Sons, 2004.

[236] J. Barthelmes, F. Lagorce-Broc, P. Kuhlkamp, S. Kok, and D. Neoh, “Different Storage Conditions have Different Whisker Growth Mechanisms-Can Bright Tin be an Alternative to

Matt Tin ?," in *Thirty-First IEEE/CPMT International Electronics Manufacturing Technology Symposium*, 2006.

[237] R. Schetty, "Whisker Growth Studies," in *IPC Annual meeting*, 2001.

[238] J. Wang, M. A. Ashworth, and G. D. Wilcox, "An Investigation Into the Role of Lead as a Suppressant for Tin Whisker Growth in Electronics," *IEEE Trans. Components, Packag. Manuf. Technol.*, vol. 4, no. 4, pp. 727–740, 2014.

[239] Uyemura international corporation, "Tin Whiskers Electrolytic plating for connectors." [Online]. Available: http://uyemura.com/electrolytic-processes.htm. [Accessed: 21-May-2014].

[240] Chestech, "Pure Tins - Chestech." [Online]. Available: http://www.chestech.co.uk/Products/Dow-Electronic-Materials/Electronic-Finishing/Pure-Tins.aspx. [Accessed: 21-May-2014].

[241] MacDermid industrial Solutions, "MacStan HSR 3.0 - Immersion Tin Final Finish," 2014.

[242] G. Milad, "Technically Speaking - The Elimiation of Whiskers from Electroplated Tin," *Metal Finishing*, pp. 1–6, 2012.

[243] A. T. Wu and Y. C. Ding, "The suppression of tin whisker growth by the coating of tin oxide nano particles and surface treatment," *Microelectron. Reliab.*, vol. 49, no. 3, pp. 318–322, 2009.

[244] S. Chopin, "Effects of reflow on the microstructure and whisker growth propensity of Sn finish," in *Proceedings Electronic Components and Technology*, 2005.

[245] M. G. P. Sanka Ganesan, *Lead-free Electronics*. John Wiley & Sons, 2006.

[246] D. Pinsky, "Tin Whisker Risk Mitigation at a Large Defense OEM – Past , Present and Future," in *7th International Symposium on Tin Whiskers*, 2013.

[247] B. D. Dunn, "Whisker Formation on Electronic Materials," *Circuit World*, vol. 2, no. 4, pp. 32–40, 1976.

[248] J. Cho, "Characterization of Nanoparticle Enhanced Conformal Coatings for Whisker Mitigation," *7th International Symposium on Tin Whiskers*. 2013.

[249] S. Han, M. Osterman, S. Meschter, and M. Pecht, "Evaluation of Effectiveness of Conformal Coatings as Tin Whisker Mitigation," *J. Electron. Mater.*, vol. 41, no. 9, pp. 2508–2518, 2012.

[250] D. Hillman and R. Collins, "Conformal Coating Materials & Application: State of the Industry Assessment," in *7th International Symposium on Tin Whiskers*, 2013.

[251] C. Hunt and M. Wickham, "Mitigation of tin whiskers with polymer coatings," in *2010 International Symposium on Advanced Packaging Materials: Microtech (APM)*, 2010.

[252] K. Mahan, Y. Sun, B. Han, S. Han, and M. Osterman, “Adhesion and Puncture Strength of Polyurethane Coating Used to Mitigate Tin Whisker Growth,” *J. Electron. Packag.*, vol. 136, no. 3, pp. 031004/1–031004/7, 2014.

[253] M. Wickham, “Improving Electronic Reliability – Through Product & Process Qualification,” in *NPL Webinar Series*, 2011.

[254] M. A. Ashworth, “Personal correspondace.” 2013.

[255] M. Lu and K. Hsieh, “Sn-Cu Intermetallic Grain Morphology Related to Sn Layer Thickness,” *J. Electron. Mater.*, vol. 36, no. 11, pp. 1448–1454, 2007.

[256] A. J. Bard, R. Parsons, and J. Jordan, *Standard Potentials in Aqueous Solution*. CRC Press Inc, 1985.

[257] MacDermid industrial Solutions, “Tinmac Stannolyte - Bright Acid Tin Solution,” 2002.

[258] S. S. Djokic, *Electrodeposition: Theory and Practice*. Springer Science & Business Media, 2010.

[259] T. Teshigawara, T. Nakata, K. Inoue, and T. Watanabe, “Microstructure of pure tin electrodeposited films,” vol. 44, pp. 2285–2289, 2001.

[260] M. A. Ashworth, “The Mitigation of Tin Whiskers by Optimization of Electroplating Process Methodologies,” in *6th International Symposium on Tin Whiskers*, 2012.

[261] A. Palacios-Padrós, F. Caballero-Briones, I. Díez-Pérez, and F. Sanz, “Tin passivation in alkaline media: Formation of SnO microcrystals as hydroxyl etching product,” *Electrochim. Acta*, vol. 111, pp. 837–845, 2013.

[262] A. Kohandehghan, K. Cui, M. Kupsta, E. Memarzadeh, P. Kalisvaart, and D. Mitlin, “Nanometer-scale Sn coatings improve the performance of silicon nanowire LIB anodes,” *J. Mater. Chem. A*, vol. 2, no. 29, pp. 11261–11279, 2014.

[263] T. Subramanyam, A. Borger, A. Matilainen, S. Torniainen, and K. Pischow, “RF transparent metallic looking coatings,” in *MIKKELI International Industrial Coating Seminar*, 2006.

[264] L. J. Durney, *Graham’s Electroplating Engineering Handbook*, vol. 1984. Springer, 1984.

[265] S. A. Chambers, “Epitaxial growth and properties of thin film oxides,” *Surf. Sci. Rep.*, vol. 39, no. 5–6, pp. 105–180, 2000.

[266] M. Yang, “Fern-shaped bismuth dendrites electrodeposited at hydrogen evolution potentials,” *J. Mater. Chem.*, vol. 21, no. 9, p. 3119, 2011.

[267] P. Singh and M. Ohring, “Tracer study of diffusion and electromigration in thin tin films,” *J. Appl. Phys.*, vol. 56, no. 4, p. 899, 1984.

[268] T. A. Woodrow and B. P. Works, “Tracer diffusion in whisker-prone tin platings,” in *Proc. SMTA Int. Conf*, 2006.

Chapter 9 Publications and presentations during the Ph.D.

Work undertaken for this Ph.D. thesis has been presented in journals and at various national and international conferences in the form of oral presentations and posters. Within this section are presented the journal papers, posters (A4 format), and oral presentations slides.

Conference/Publication	Location	Form	Date
Department of Chemistry, Research day	Loughborough	Poster	24/04/2012
Department of Materials, Research day	Loughborough	Poster	11/06/2012
Department of Materials, Research day, Micrograph competition winner	Loughborough/online	Micrograph	11/06/2012
Midlands Electrochemistry Group meeting	Birmingham University	Poster	27/06/2012
An investigation of the growth of bismuth whiskers and nanowires during physical vapour deposition	Journal of Physics D: Applied Physics Volume 45 Number 43	Paper	09/10/2012
6th International Symposium on Tin Whiskers	Loughborough	Poster	27/11/2012
Department of Materials, Research day	Loughborough	Oral Presentation	15/07/2013
Department of Materials, Research day, Micrograph competition winner	Loughborough/online	Micrograph	15/07/2013
Eurocorr	Lisbon, PT	Poster	03/09/2013
IeMRC Conference	Loughborough	Poster	25/09/2013
7th International Symposium on Tin Whiskers	Costa mesa, USA	Oral Presentation	12/11/2013
Molten Salt Discussion Group	Warwick	Oral Presentation	10/01/2014
Materials Exchange	Coventry	Poster	25/02/2014
Characterisation of tin–copper intermetallic growth in electrodeposited tin coatings using electrochemical oxidation techniques	Transactions of the IMF	Paper	06/05/2014
Eurocorr 2014 paper – The electrodeposition of Sn coatings from deep eutectic solvents and their subsequent whisker growth	Pisa, IT	Oral Presentation	08/09/2014

9.1 Papers

- *"Characterisation of tin–copper intermetallic growth in electrodeposited tin coatings using electrochemical oxidation techniques*", Transactions of the IMF, (2014).
- "*An investigation of the growth of bismuth whiskers and nanowires during physical vapour deposition*", Journal of Physics D, (2012)

Paper presented at Eurocorr 2014

- *"The electrodeposition of Sn coating from deep eutectic solvent and their subsequent whisker growth", Eurocorr, 2014*

9.2 Posters

Poster presentation location	Location	Date
Department of Chemistry, Research day	Loughborough	24/04/2012
Department of Materials, Research day	Loughborough	11/06/2012
Midlands Electrochemistry Group meeting	Birmingham University	27/06/2012

Tin whisker mitigation through the electrodeposition of tin and novel tin alloys from ionic liquids.

Researcher: Christopher Stuttle Supervisors: Geoffrey Wilcox and Roger Mortimer

What are tin whiskers?

Shapes - Whiskers are commonly single crystal filaments with constant cross-sectional diameters of around 3µm and lengths between a few microns and several hundred, but can also be found in many other shapes such as nodules and eruptions (Fig.1). With the long filament whiskers being of particular interest as they cause high failure rates from shorting in electrical equipment.

History – Whisker studies started in the 40s with cadmium whiskers followed by whiskers being found to grow from other metals such as zinc, silver, bismuth and tin. Tin is used widely in electronics as a soldering surface and since 1959 lead at 3-10 wt.% had been co-deposited to almost totally mitigate whisker growth. But since 2003 the 'Restriction of certain Hazardous Substances in electrical and electronic equipment' (**RoHS**) enforced the removal of lead from electronics, leaving tin whiskers to grown uninhibited.

Aim –To find an effective lead free mitigation method that will significantly reduce whisker growth.

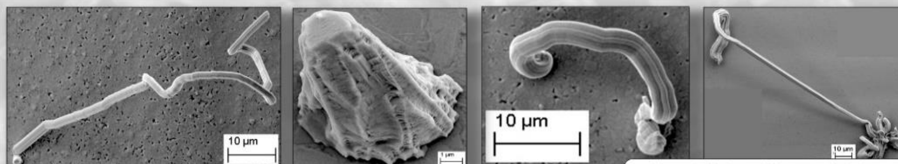


Figure 1; Various whisker morphologies

What causes whiskers ?

Stress – The mechanism by which whiskers grow is unclear but compressive stresses within the electrodeposited tin increase whisker growth. This means that there are a large number of coating characteristics that play a significant part in whisker growth due to their connection to compressive stresses such as:

- **Intermetallic compound (IMC) growth.**
- **Grain morphology.**
- **Electroplating bath chemistry.**

Graph 1; Galvanostatic stripping

Stresses from intermetallic compounds (IMC).

IMC growth can be analysed by galvanostatic stripping (Graph 1) and also by SEM. Graphs of galvanic stripping show the oxidation of tin and then two tin-copper IMCs that require ever increasing potentials allowing for compound specific quantitative analysis. SEM images of the interface after stripping then shows the morphology of the IMC to be seen (Fig.2) and highlights its high rate of growth.

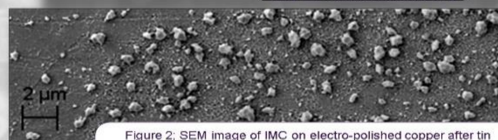


Figure 2; SEM image of IMC on electro-polished copper after tin stripping – 1 days growth.

Characterising Tin electrodeposits.

The grain morphology and topography of pure tin electrodeposits has been studied at current densities from 0.1A/dm^2 to 3.3A/dm^2 (Fig. 3) . The change in the current density clearly show a change in grain morphology from columnar to a more equiaxed structure.

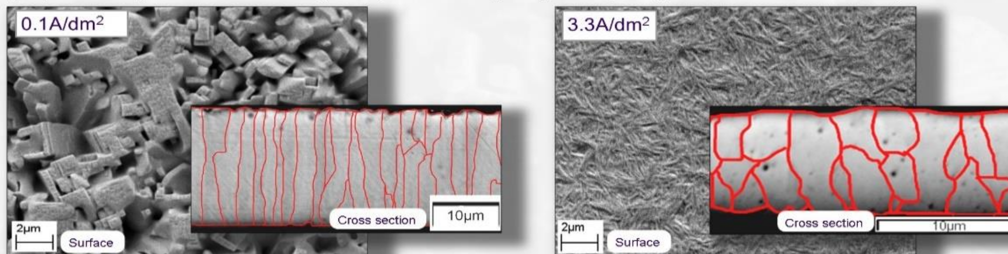


Figure 3; SEM image - grain morphology and topography - 0.1 A/dm^2 ,3.3 A/dm^2

Future work with ionic liquids.

The characteristics of the tin electrodeposits shown here as well as their propensity to whisker will be compared with electrodeposits created from a bath using an ionic electrolyte. Following this, alloys will be electroplated from an ionic liquid, characterised and whisker growth rates defined to find the optimum alloy and plating conditions to mitigate whisker growth.

Department of Materials June 2012 Loughborough University Centenary Studentship

Poster presentation location	Location	Date
6th International Symposium on Tin Whiskers	Loughborough	27/11/2012
Eurocorr	Lisbon, PT	03/09/2013
IeMRC Conference	Loughborough	25/09/2013
Materials Exchange	Coventry	25/02/2014

Characterisation of Sn-Cu Intermetallic Growth in Electrodeposited Tin Films on Copper Using Electrochemical Anodic Oxidation Techniques

C.J Stuttle[1], Dr M.A. Ashworth[1], Dr G.D. Wilcox[1] and Prof R.J. Mortimer[2]

[1] Department of Materials, [2] Department of Chemistry
Loughborough University, Loughborough, Leicestershire, UK LE11 3TU

Introduction

The electrodeposition of Sn on Cu circuitry is frequently utilised due to its highly beneficial properties such as corrosion and oxidation resistance and solderability. However, Sn electrodeposits have been reported, at times, to result in whisker formation, see Figure 1. A major contributor to their growth is thought to be the formation of Cu_6Sn_5 intermetallic compound at the coating-substrate interface. In the present investigations electrochemical oxidation allowed for the complete removal of Sn from electroplated Cu substrates permitting SEM to be employed to assess intermetallic compound distribution and morphology for a low whiskering bright Sn coating.

Tin Whiskers

One drawback of Sn usage is the propensity of such coatings to produce whiskers. These are protrusions that grow due to compressive stresses from the coating post–electrodeposition with diameters of a few µm and lengths that can range from just a few µm to several mm.

Since the 1950s Sn had been alloyed with Pb at 3–10 %wt. to successfully militate against Sn whiskers.[5] With the enforcement in 2006 of the 'Restriction of the use of certain Hazardous Substances in electrical and electronic equipment' (RoHS) the application of Sn–Pb was restricted due to the toxicity of lead, thus leaving Sn electrodeposits without an adequate whisker mitigation strategy.[5]

Without this primary means of mitigation, Sn whiskers may freely grow causing far greater numbers of incidences of malfunctions such as signal distortions, intermittent shorts within circuits and even, with sufficient power, extensive electrical failures.[6]

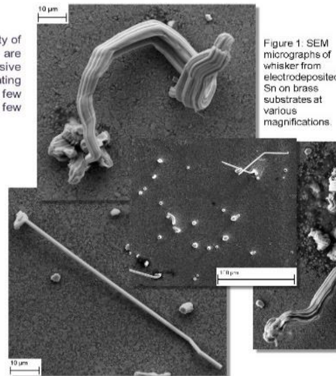


Figure 1: SEM micrographs of whisker from electrodeposited Sn on brass substrates at various magnifications.

Interfacial Intermetallic Compound Growth

Compressive stresses within Sn coatings are a major contributing factor to whisker growth.[1] these stresses originate from several possible sources; residual electroplating stress, mechanical stress, surface oxidation and coating and substrate–coating interfacial reactions promoting intermetallic compound (IMC) formation. [1]

At the interface, diffusion of Cu from the substrate into the Sn coating takes place resulting in the formation of Cu_6Sn_5 (η′–phase IMC) as shown in Figure 2 and 3.[2] The growth of η′–phase IMCs has been suggested to lead to a localised expansion causing compressive stresses and thus whisker growth.[3]

Studies have revealed that as the IMC distribution becomes more uniform, with less large, localised growth there is also a reduction in whisker growth.[4] This, therefore emphasises that the morphology and distribution of IMCs are significant characteristics to record rather than just the thickness of IMC growth.

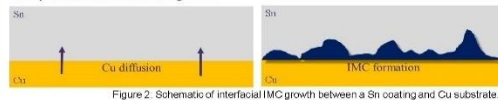


Figure 2: Schematic of interfacial IMC growth between a Sn coating and Cu substrate.

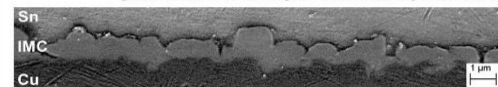
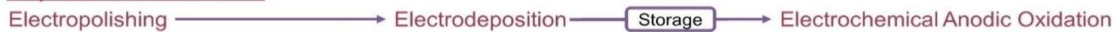


Figure 3: SEM micrograph of IMC growth between a Sn coating and Cu substrate.

Experimental Method



Electropolishing

100 µm thick Cu substrates were cut to 4 cm x 2 cm dimensions. To remove the as rolled surface topography electropolishing was carried out potentiostatically in 50 vol.% phosphoric acid using a potentiostat (Solartron ECI 1286) with counter (CE), working (WE), and reference (RE) electrodes of Cu, Cu substrate and saturated calomel, respectively.

The treatment was carried out at 1.0 V vs. Saturated calomel electrode (SCE) at room temperature (20 ºC) for 4 mins. Additionally, before electrodeposition the Cu was degreased in acetone and pickled in 20 vol.% S.G. 1.83 sulphuric acid.

Electrodeposition

Electrodeposition of Sn was undertaken from an acid sulphate proprietary bath containing 0.28 M $SnSO_4$, 1.29 M H_2SO_4 and 40 ml/L of a proprietary additive; Tinmac.

A three electrode cell was used with the Cu substrates masked off to leave a WE area of 4 cm^2, 0.25 mm thick 4 cm^2 Sn CE and SCE RE.

The Sn coatings were electrodeposited at a galvanostatic current density of 20 mA/cm^2 to a thickness of 5 µm using a potentiostat resulting in uniform and bright deposits.

Electrochemical Anodic Oxidation

After various storage periods (24 h, 17 days and 55 days) removal of the electrodeposited Sn layer was undertaken via an electrolytic process utilising H_2SO_4 at 0.9 M as the electrolyte. The anodic oxidation reaction was polarised potentiostatically utilising a potentiostat.

To produce samples for IMC analysis Sn electrodeposited samples were treated in the electrolyte at a potential of -0.26 V vs. SCE. The samples were then anodically polarised in a three electrode cell using Sn electroplated samples as the WE, Cu foil as the CE and SCE as the RE until the current due to Sn oxidation had reduced to approximately zero.

Results and Discussion

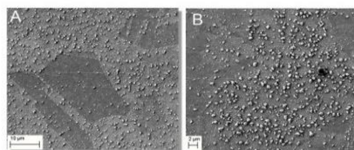


Figure 4 above: SEM micrographs of IMC growth after 24h on electropolished Cu substrates. Showing areas of low density 'A' and high density 'B' IMC growth.

As shown in Figure 4, after 24 h the IMC forms small individual particles with a variation in particle density across the surface of the substrate in distinct patterns. These distinct patterns are comparable in shape and size to the copper substrate grains. This indicates that initial IMC growth was highly dependent upon the underlying copper grain orientation.

Figure 5 shows that after 17 days the IMC particles coarsen and begin to coalesce in some areas. Furthermore, small individual particles of IMC are visible indicating that either IMC nucleation is still occurring or that there is a wide range of IMC particle coarsening rates taking place. In addition, in Figure 4 preferential IMC growth is apparent at the copper grain boundaries. It is clear that increased IMC particle nucleation, coarsening and coalescence is occurring on the copper grain boundaries.

It is clear from these results that initial IMC growth for this coating is more highly dependent upon the copper grain structure compared of the tin coating structure.

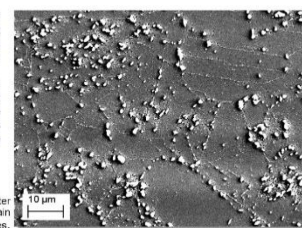


Figure 5: SEM micrograph of IMC growth after 17 days depicting IMC growth along Cu grain boundaries.

Figure 6 shows, after 55 days of storage an almost continuous layer of IMC with small, sporadic patches distributed across the whole of the sample where IMC particles had not formed were present. Additionally, samples left for longer time periods exhibited a fully continuous layer of IMC.

This rapid growth of a continuous layer of IMC that formed after 55 days of storage had not been reported before and may be why this Sn coating shows low levels of whisker growth.

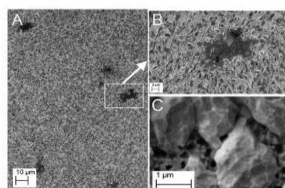


Figure 6: SEM micrographs showing the IMC growth after 55 days featuring an almost complete layer of IMC. 'A' and 'B' show areas with no IMC at increasing magnification, and 'C' at high magnification showing detail of the IMC structure.

Conclusions

- Complete anodic electrochemical removal of Sn electrodeposits using 0.9 M H_2SO_4 was effective in leaving the Cu substrates and IMCs unaffected thus allowing detailed analysis by SEM.
- IMC growth after 24 h of storage was dependent upon specific Cu grain orientations, grain boundaries and to lesser degree prior Sn electrodeposit grain boundaries highlighting the importance of substrate texture as opposed to coating structure.
- IMC growth after 55 days of storage resulted in almost complete interface coverage. It is thought that the growth of a uniform layer of IMCs is partly responsible for the low level of whisker growth observed for this Sn coating system.

REFERENCES

1. C. Xu, et al. *"Driving force for the formation of Sn whiskers: compressive stress-pathways for its generation and remedies for its elimination and minimization."* IEEE Transactions on Electronics Packaging Manufacturing, vol. 28, no. 1. pp. 31–35, Jan. 2005.
2. K. N. Tu, *"Interdiffusion and reaction in bimetallic Cu-Sn thin films,"* Acta Metallurgica, vol. 21, no. 4. pp. 347–354, Apr. 1973.
3. B.-Z. Lee, et al. *"Spontaneous growth mechanism of tin whiskers,"* Acta Materialia, vol. 46, no. 10, pp. 3701–3714, Jun. 1998.
4. G. Milad. *"The elimination of whiskers from electroplated tin."* Circuit World, vol. 37, no. 4. pp. 10–15. Nov. 2011.
5. G. T. Galyon. *"Annotated tin whisker bibliography and anthology."* IEEE Transactions on Electronics Packaging Manufacturing, vol. 28. no. 1, pp. 94–122, Jan. 2005.
6. R. Fisher, et al. *"Accelerated growth of tin whiskers,"* Acta Metallurgica, vol. 2, no. 3, pp. 368–373, May 1954.

ACKNOWLEDGEMENTS

- Loughborough University Graduate School Centenary Scholarship
- WHISKERMIT project based at Loughborough University funded by the IeMRC/EPSRC
- MacDiarmid plc. for the provision of the Tinmac additive used throughout this investigation

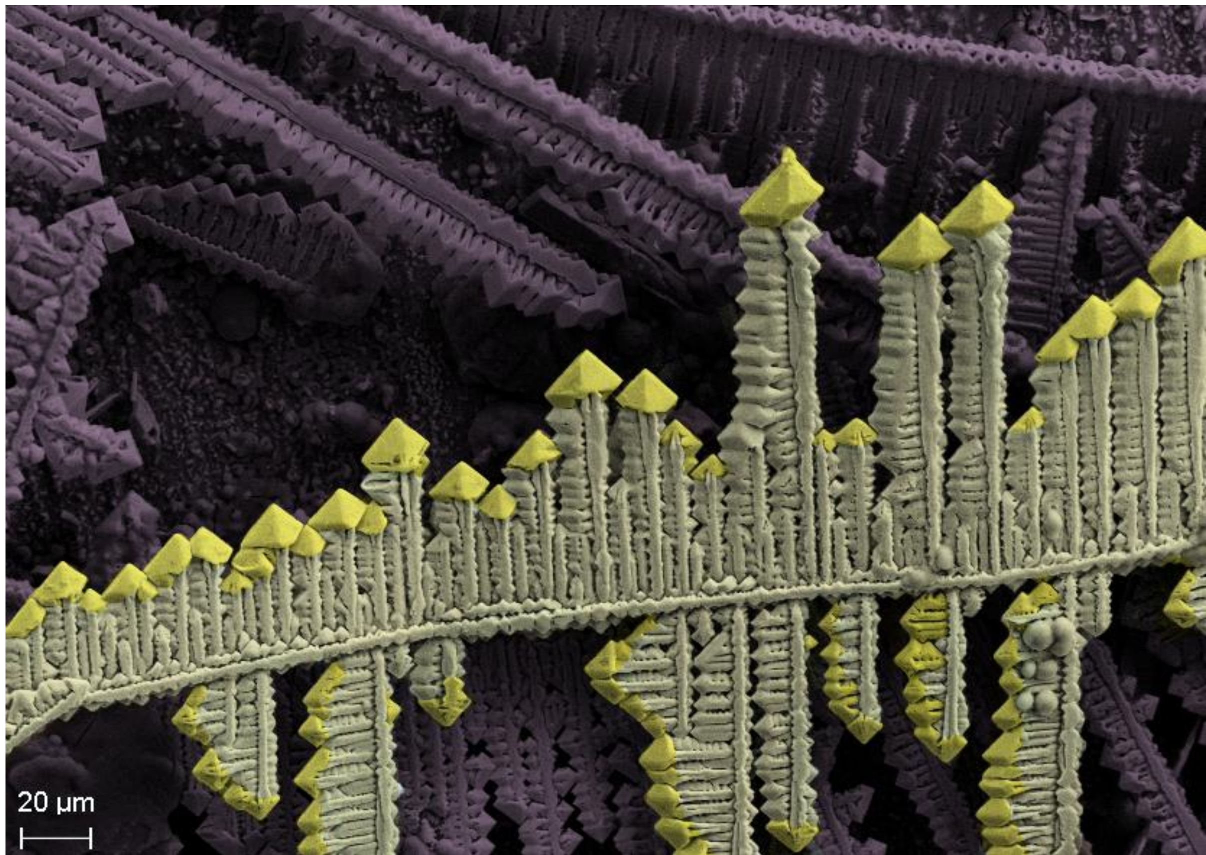
CONTACT INFORMATION
Department of Materials
Loughborough University
Leicestershire LE11 3TU, UK
C.Stuttle@lboro.ac.uk
www.lboro.ac.uk/materials

9.3 Oral presentation slides

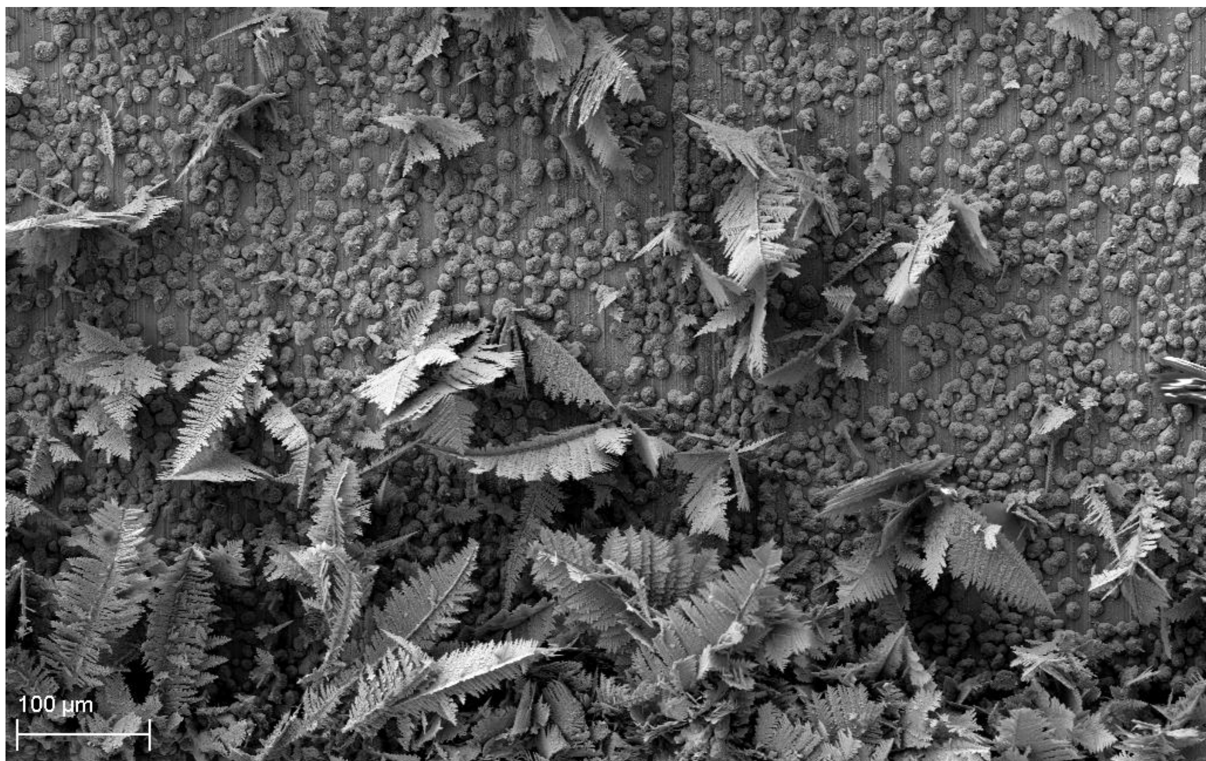
The oral presentation of this thesis has been presented internationally and within the UK.

Oral presentation location	Location	Date
Department of Materials, Research day	Loughborough	15/07/2013
7th International Symposium on Tin Whiskers	LA, USA	12/11/2013
Molten Salt Discussion Group	Warwick	10/01/2014

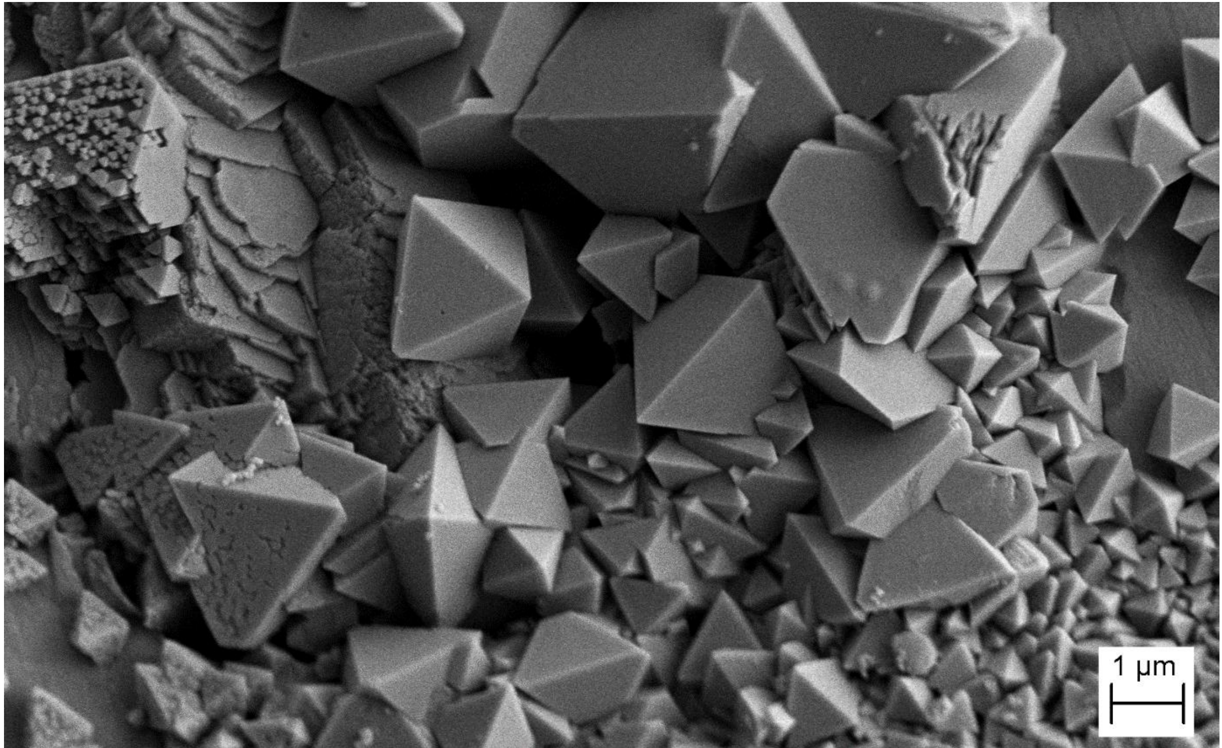
9.4 Competition winning micrographs



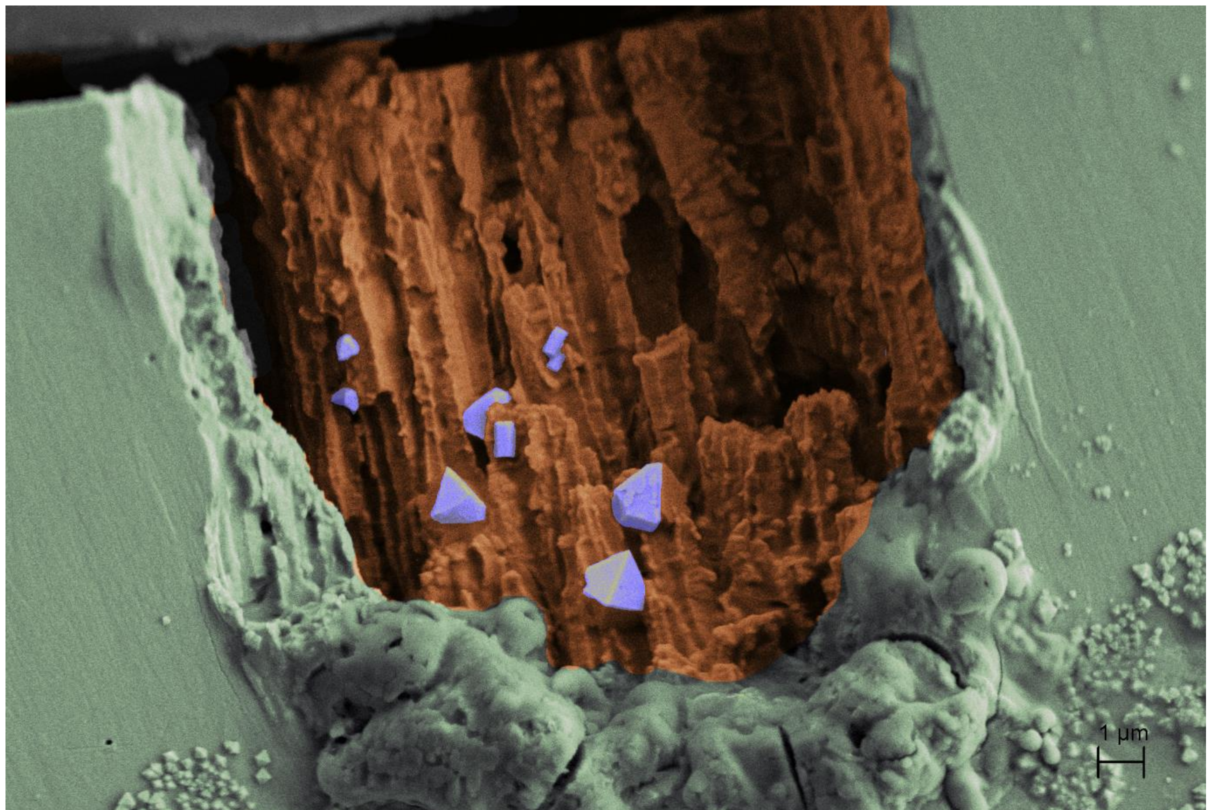
2013-1st Prize-Colour: Miniature Fern - Dendrites formed during the electrodeposition of tin onto a copper substrate



2013-2nd Prize-B&W: Christopher Stuttle, Autumn Fall - Dendrites formed during the electrodeposition of tin onto a copper substrate



2012-2nd Prize - B & W: Christopher Stuttle, Diamonds for the Jubilee – 10 µm thick electroplated tin on a copper substrate



2012-1st Prize - Colour: Christopher Stuttle, Sapphire Mine - 10µm thick electroplated tin on copper viewed in cross section

Chapter 10 Appendix

10.1 Non-aqueous electrolyte DC voltammetry for tin electrodeposition

Additional voltammetry, to that for Reline shown in Figure 4.56, is shown in Figure 10.1 for Ethaline and Figure 10.2 for Propeline, with additions of 0, 0.044, 80 and 0.71 M $SnCl_2$ depicts what was seen for all three electrolytes .

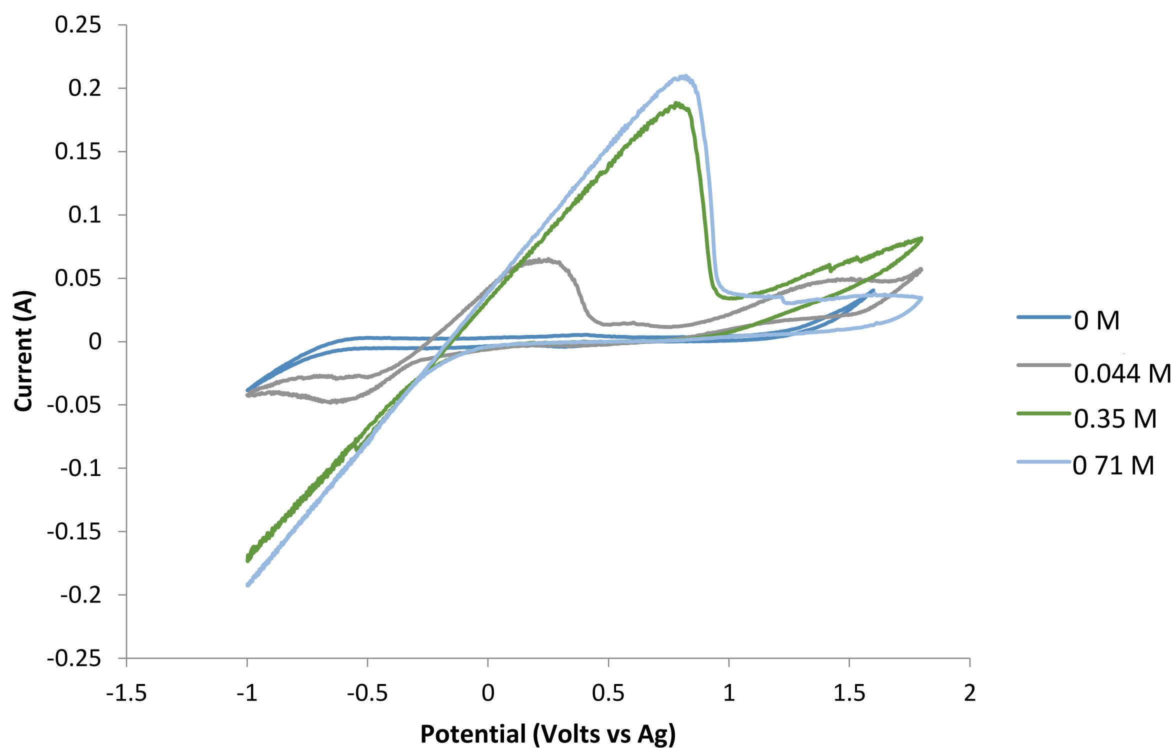


Figure 10.1: Voltammogram for Ethaline with additions of $SnCl_2.2H_2O$ to concentration of 0, 0.044, 0.35, 0.71 M at 70 ^oC.

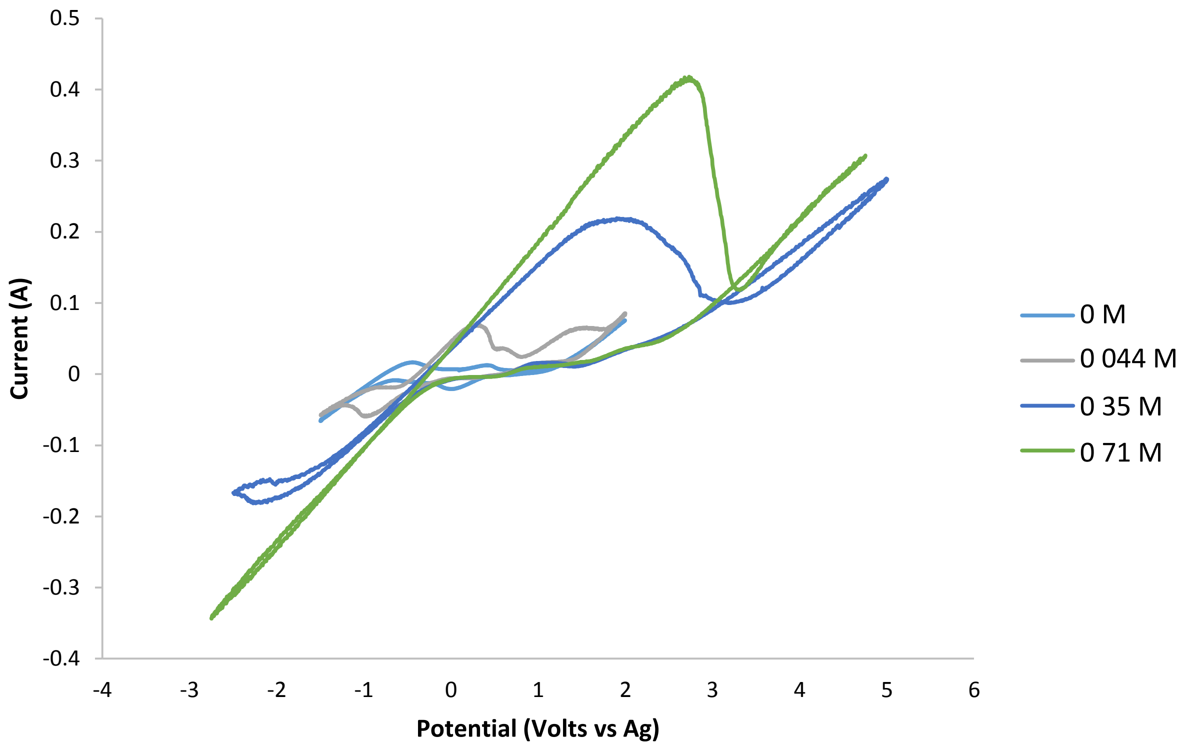


Figure 10.2: Voltammogram for Propeline with additions of $SnCl_2.2H_2O$ to concentration of 0, 0.044, 0.35, 0.71 M at 70 ^{0}C.

10.2 Aqueous electrodeposit grain size distributions

The distribution of grain area for electrodeposits created from the Tinmac aqueous proprietary electrolytes for a range of current densities, are shown In Figure 10.3; with agitation, and Figure 10.4; without agitation.

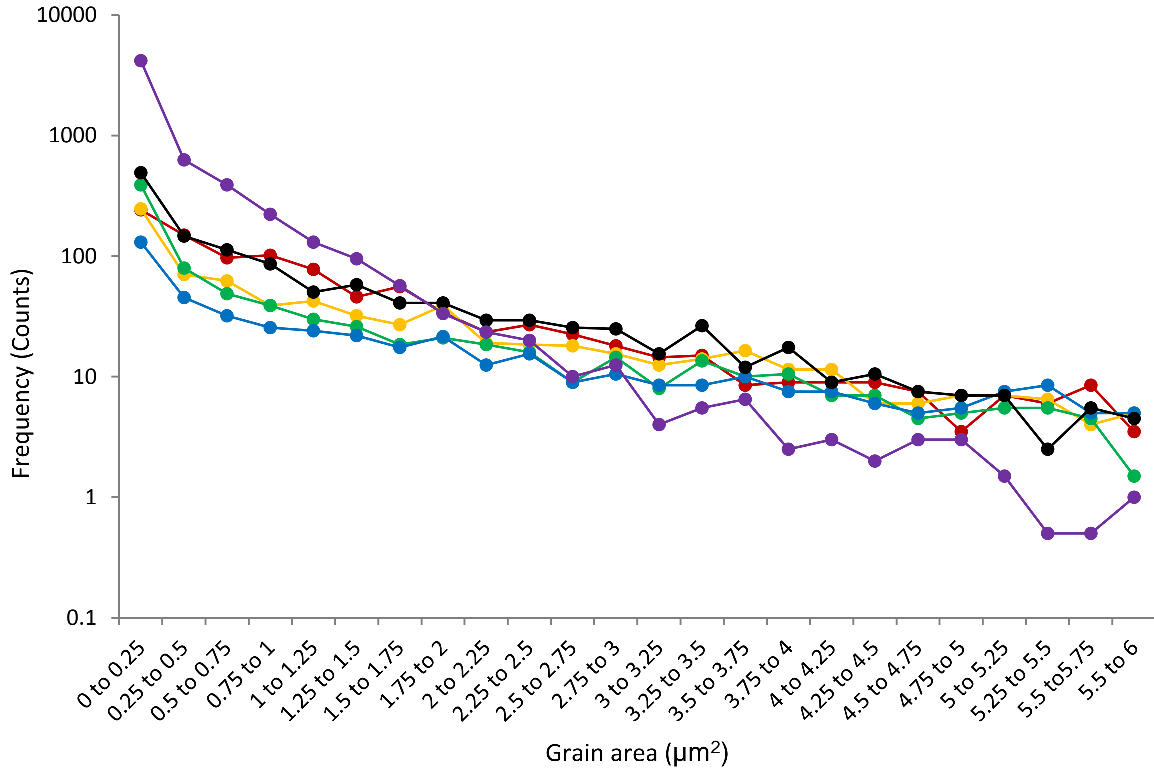


Figure 10.3: Log-normal histograms of grain area for electrodeposits created with agitation. Red - 5 mA/cm². Yellow - 10 mA/cm². Green - 20 mA/cm². Blue - 30 mA/cm². Black - 40 mA/cm². Purple - 50 mA/cm².

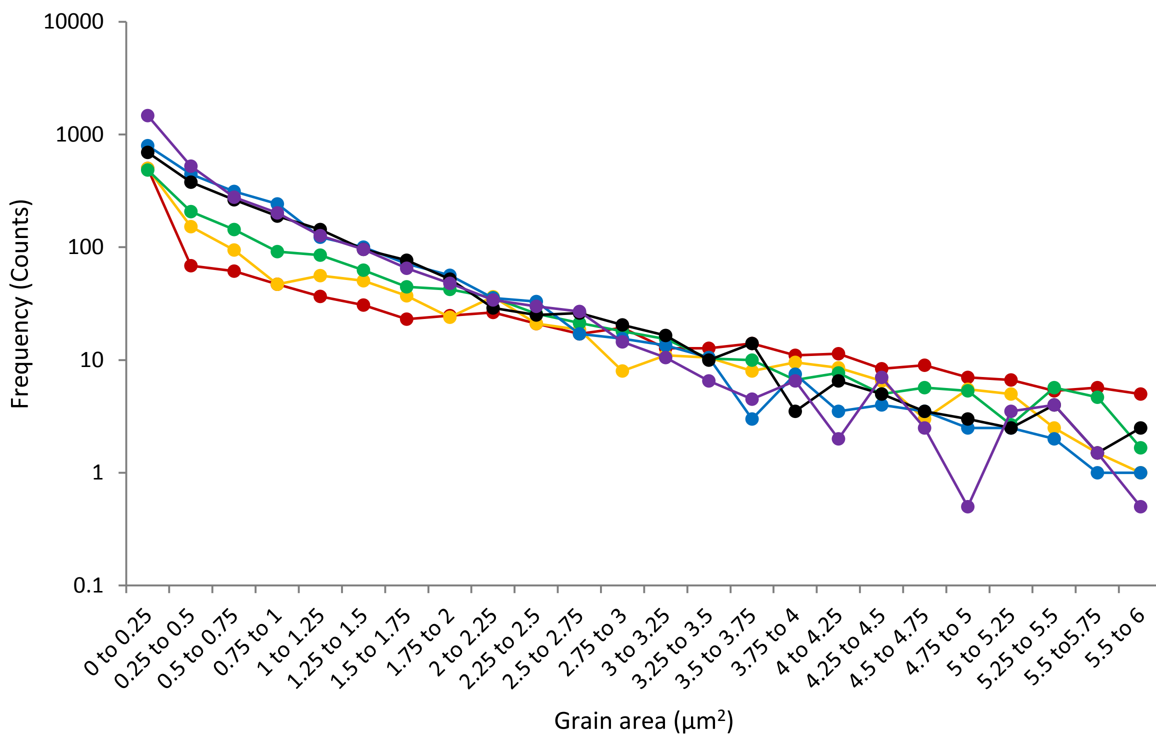


Figure 10.4: Log-normal histograms of grain area for electrodeposits created without agitation. Red; 5 mA/cm². Yellow; 10 mA/cm². Green; 20 mA/cm². Blue; 30 mA/cm². Black; 40 mA/cm². Purple; 50 mA/cm².

10.3 Thermodynamics – whisker growth by stress relaxation

By evaluating the energy of an object upon an experienced potential and energy held in its free surfaces we can obtain Equation 10.1.

$$\mathrm{U} = \mathrm{U_R} - \mathrm{U_A} + \mathrm{k} \qquad \text{Equation 10.1}$$

Where; U is the total energy of the system, U_R is the potential energy due to strain, U_A is the potential energy loss due to surface free energy and k denotes other unknown constant energies.

Moreover, by energy conservation the change in total energy, providing no frictional forces, ΔU must be greater or equal to zero when passing between two given states; 1 and 2. Therefore, we obtain Equation 10.3

$$\Delta U \leq {U_R}_2 - {U_A}_2 + k - {U_R}_1 + {U_A}_1 - k \qquad \text{Equation 10.2}$$

$$0 \leq \Delta U \leq \Delta U_R - \Delta U_A \qquad \text{Equation 10.3}$$

Additionally, by modelling a whisker as a rudimentary cylinder, and the volume undergoing stress relaxation as another cylinder within the coating, as shown in Equations 10.4 to 10.6 we obtain Equation 10.7.

$$U_R = \int \rho \, dV_R \qquad \text{Equation 10.4}$$

$$dV_R = 2\pi r_R \, dL_R dr_R \qquad \text{Equation 10.5}$$

$$\Delta U_R = {U_R}_2 - {U_R}_1 = \int \rho_2 \, d{V_R}_2 - \int \rho_1 \, d{V_R}_1 \qquad \text{Equation 10.6}$$

$$\Delta U_R = 2\pi \left[\iint \rho_2 \, {r_R}_2 \, {L_R}_2 d{r_R}_2 - \int \rho_1 \, {r_R}_1 \, {L_R}_1 d{r_R}_1 \right] \qquad \text{Equation 10.7}$$

Where the volume V_R experiencing strain energy density ρ is defined by variables radius r_R, length LR.

Strain energy density obviously, is always positive due to its parity with respect to r_R, but for the sake of energy flow this parity is considered odd; positive under tension and negative under compression.

Additionally, the energy change due to additional free surface area can be evaluated with Equation 10.9 which has been derived from its more explicit form in Equation 10.8.

$$U_A = \gamma A = \gamma[2\pi r_w(L_W - L_0) + \pi {r_w}^2] \qquad \text{Equation 10.8}$$

$$\Delta U_A = {U_A}_2 - {U_A}_1 = \gamma 2\pi r_w \left({L_W}_2 - {L_W}_1 \right) \qquad \text{Equation 10.9}$$

Where r_w is the radius of a whisker, L_w is the length of the whisker and L_0 is the length of the whisker surrounded by the tin coating.

By substituting Equations 10.9 and 10.7 into Equation 10.3 we obtain Equation 10.10.

$$0 \leq 2\left[L_{R_2} \int \rho_2 \, r_{R_2} \, dr_{R_2} - L_{R_1} \int \rho_1 \, r_{R_1} dr_{R_1}\right] - \gamma 2r_w \left(L_{W_2} - L_{W_1}\right) \qquad \text{Equation 10.10}$$

It is Equation 10.10 which provides a general model and allows the substitution of different conditions to represent whisker growth due to compressive stress relaxation.

10.3.1 *Whisker growth from the relaxation of its original grain*

Firstly by letting expansion to occur until fully relaxed, i.e. ρ_2=0, and letting the radius undergoing relaxation being equal to the radius of the whisker, i.e. modelling the stress relation taking place at the root of the whisker, Equation 10.10 becomes Equation 10.11.

$$0 \leq -2L_{R_1} \int \rho_1 \, r d\mathrm{r} - \gamma 2\mathrm{r}_w \left(L_{W_2} - L_{W_1}\right) \qquad \text{Equation 10.11}$$

Providing ρ(r) is constant with respect to r, Equation 10.11 can then be rearranged to give Equation 10.12.

$$\rho_1 \leq \frac{-2\,\gamma \Delta L_w}{r_w L_R} \qquad \text{Equation 10.12}$$

Where ΔL_w is the additional whisker length and L_R is the length under relaxation.

Additionally, if the volume under relaxation becomes the volume added to the whisker length this dictates that Equation 2.7 becomes Equation 2.8.

$$\rho_1 \leq \frac{-2\,\gamma}{r_w} \qquad \text{Equation 10.13}$$

Therefore the application of Equation 10.13 provides a limit of allowable whisker radii for a given strain energy density [77]. Although this is an interesting result it actually does not disallow many whisker radii. On the other hand it does suggest a value for strain energy density yield, which when exceeded for a grain, whisker growth is permitted. Although the total volume undergoing relaxation must be large, as seen in the following section, the volume undergoing relaxation directly leading to whisker growth must take place at the whisker root and therefore must obey Equation 10.13.

10.3.2 *Whisker growth from deposit relaxation*

Tin, an already highly diffusive material [267], is said to be able to form large whiskers by long range diffusion [77], [203], [268]. If we model the energy involved in whisker growth as before, but with the material under stress relaxation to be a volume of the deposit far larger than the whisker root itself, i.e. the conditions in Equation 10.14, by substituting these into 10.10 we can obtain Equation 10.15.

$$L_{R_1} = L_{R_2} = L_R,$$
$$r_{R_2} = r_{R_2} = r_R \neq r_w, \qquad \text{Equation 10.14}$$
$$\rho_1(r_R) = const.$$

$$0 \leq 2\left[L_R\left(\int \rho_2\, r_R\, dr_R - \rho_1 \frac{{r_R}^2}{2}\right)\right] - \gamma 2 r_w\left(L_{W_2} - L_{W_1}\right) \qquad \text{Equation 10.15}$$

Equation 10.15 requires the final strain energy density to be integrated over r_R. This requires two possible solutions of $\rho_2(r_R)$ as shown in Equation 10.16 and 10.17. Equation 10.16 depicts a uniform reduction in strain energy across a given radius, whereas Equation 10.17 represents a linear relationship between final stress energy and r_R.

$$\rho_2(r_R) = const. \qquad \text{Equation 10.16}$$

$$\rho_2(r_R) = Mr_R + K \qquad \text{Equation 10.17}$$

Where M and K are constants with respect to r_R

Both of these models of stress relaxation over a volume of material within the deposit are represented in Equation 10.17.

Constant reduction in strain energy density

Therefore by substituting Equation 10.16 into 10.15 and integrating we obtain Equation 10.18

$$0 \leq L_R {r_R}^2(\rho_2 - \rho_1) - \gamma 2 r_w \left(L_{W_2} - L_{W_1}\right) \qquad \text{Equation 10.18}$$

Additionally, by defining the final strain energy density in the volume undergoing relaxation to be zero, we obtain Equation 10.19.

$$\rho_2 = 0,$$

$$ {r_R}^2 \geq \frac{-2\gamma r_w \left(L_{W_2} - L_{W_1}\right)}{\rho_1 L_R} \qquad \text{Equation 10.19}$$

Linear reduction in strain energy density

A more realistic model is to suggest that there is a linear relationship of final stress in relation to r_R. Therefore by substituting Equation 10.17 into 10.15 and integrating we obtain Equation 10.20.

$$0 \leq 2\left[L_R\left(M\frac{{r_R}^3}{3} + K\frac{{r_R}^2}{2} - \rho_1\frac{{r_R}^2}{2}\right)\right] - \gamma 2 r_w \left(L_{W_2} - L_{W_1}\right) \qquad \text{Equation 10.20}$$

Therefore by defining the strain energy density to be zero at the whisker base and the gradient of the linear relationship to reach its initial levels of ρ_1 at r_R we obtain Equation 10.21.

$$\rho_2(0) = 0,$$

$$\rho_2(r_R) = \rho_1$$

$$K = 0,$$

$$M = \frac{\rho_1}{r_R}$$

$${r_R}^2 \geq -3\frac{2\gamma r_w \left(L_{W_2} - L_{W_1}\right)}{\rho_1 L_R} \qquad \text{Equation 10.21}$$

Therefore, Equation 10.19 and 10.21 can be represented graphically with variables of compressive stress and radius of deposit experiencing relaxation, as shown in Figure 10.5.

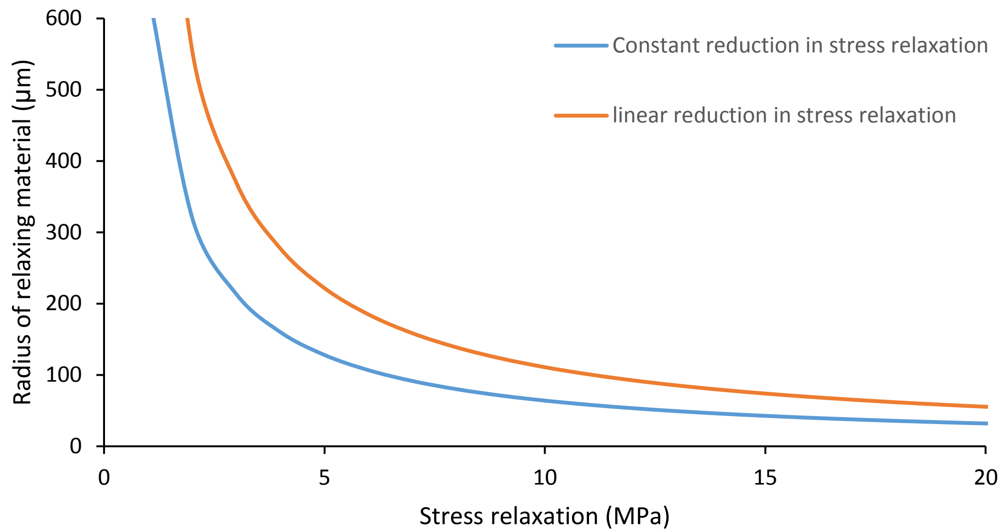


Figure 10.5: Graphical representation of Equation 10.19 (blue) and 10.21 (orange) relating the radius of material undergoing relaxation and the stress relaxation it is experiencing for a given whisker.

Figure 10.5 clearly shows that by using a linear relationship, compared to a constant value, between stress relaxation and radius, requires more material for a given volume of whisker growth. Moreover, provided sufficient stress relaxation occurs, the radius of material effected by whisker growth drops significantly.

10.4 3DView programme code

10.4.1 *Form appearance*

The code defining the layout of the form, the main programme screen. The form and its drop down boxes while in use are shown in Figure 10.6 and Figure 10.7.

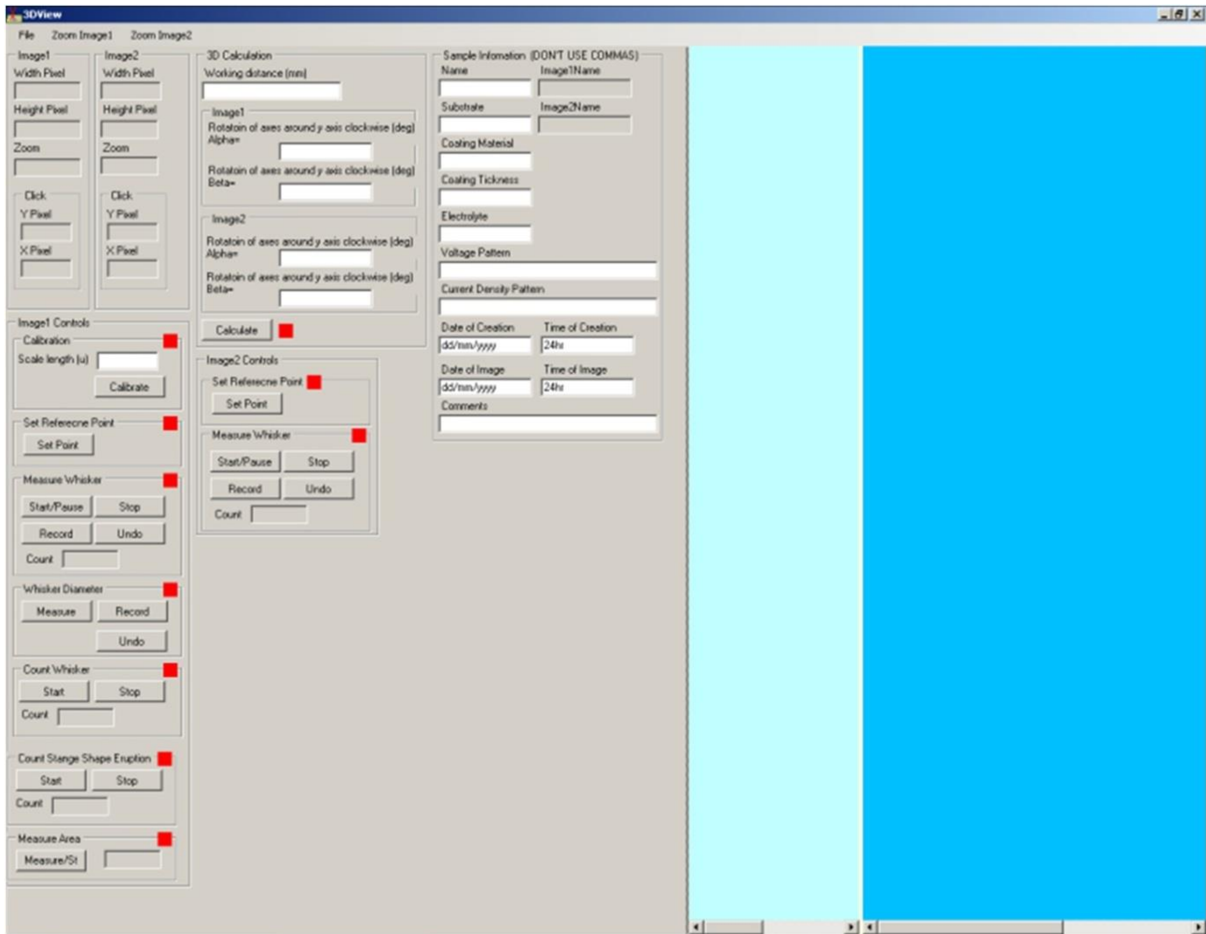


Figure 10.6: View of 3DView form on opening the application. Featuring: user buttons (left) with sample specification to its right, then with image 2 and image 1boxes (right).

In addition, user buttons tool strip controls at the top of the programme are seen in Figure 10.7.

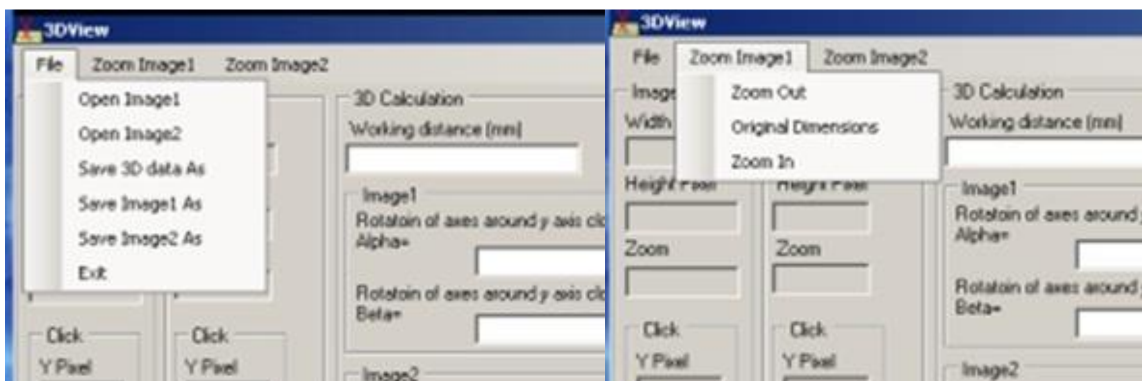


Figure 10.7: View of 3DView featuring tabs for the drop down options possible within the programme

10.4.2 *Programme code*

```
#pragma once
#include <windows.h>
#include <cmath>
#include <fstream>
#include <iostream>




namespace PictureViewer4 {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;


    //declaring global variable
    double ImageZoom = 1;   //image zoomvariable used in zoom buttons
    double Image2Zoom = 1;
    bool ImageLoaded = false;
    bool Image2Loaded = false;
    bool AreaMeasure = false;
    int AreaMeasureNumber = 1;
    int AreaMeasureArray[3];
    bool Cailbration = false;   //

    int CalibrationArray[3];
           //calibration array parts
    int CalibratePointNumber = 1;
           //
    bool CalibrationComplete = false;

    double LengthPerPixel;

    bool SetRefPoint1=false;
    bool SetRefPoint1Complete=false;
    bool SetRefPoint2=false;
    bool SetRefPoint2Complete=false;
    double RefPoint1Array[3];
    double RefPoint2Array[3];

    bool DiameterMeasure=false;
    double DiameterMeasureArray[700][2][2];
    double DiameterMeasurePointNumber=1;
    int w3=1;



    int dy;//for area
    int dx;
    bool AreaComplete = false;
    double ImageWidth;
    double Image2Width;
    double ImageHeight;
```

```
double Image2Height;
bool MeasureWhiskerImage1 = false;
bool MeasureWhiskerImage2 = false;
double WhiskerArrayImage1[700][30][3];
double WhiskerArrayImage1BaseOrigin[700][30][3];
double WhiskerArrayImage1Polar[700][30][4];
double WhiskerArrayImage1PolarImaginary[700][30][4];
double WhiskerArrayImage1RealXYZ[700][30][4];
double WhiskerArrayImage1RealXYZImaginary[700][30][4];
double WhiskerArrayImage2[700][30][3];
double WhiskerArrayImage2BaseOrigin[700][30][3];
double WhiskerArrayImage2Polar[700][30][4];
double WhiskerArrayImage2PolarImaginary[700][30][4];
double WhiskerArrayImage2RealXYZ[700][30][4];
double WhiskerArrayImage2RealXYZImaginary[700][30][4];
double WhiskerArray3d[700][30][4];
bool MeasureWhiskerComplete = false;
int a = 1;// used for whisker point collection
int a2 = 1;
//int b = 0; //used for whisker line drawing
int w = 1;// used in array denoting whisker number
int w2 = 1;
bool CalculateComplete = false;

double xab;
double yab;
double zab;
int K;
double Ua;
double Ub;

bool CountStangeShapeEruption = false;
double StangeShapeEruptionArray[400][3];
double StangeShapeEruptionArrayOrign[400][3];
double StangeShapeEruptionArrayImaginary[400][3];
double StangeShapeEruptionArray3d[400][4];
double StangeShapeEruptionArrayXYZ[400][4];
double StangeShapeEruptionArrayXYZImaginary[400][4];
int j = 1; //used for eruption count collection

bool CountWhisker = false;
double WhiskerCountArray[400][3];
double WhiskerCountArrayOrign[400][3];
double WhiskerCountArrayImaginary[400][3];
double WhiskerCountArray3d[400][4];
double WhiskerCountArrayXYZ[400][4];
double WhiskerCountArrayXYZImaginary[400][4];
int k = 1; //used for whisker count collection


double                                                                      PI=
3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628
0348253421;

double Image1Alpha;
double Image1Beta ;
double Image2Alpha;
double Image2Beta;
```

```
        //defining function to take 2 points and create the intersect with the x,y
plane to give erruption position

        double eruption(int K, double x1, double y1, double z1, double x2, double y2,
double z2)
        {
                double U =(-z1)/(z2-z1);
                double xa =x1+(U*(x2-x1));
                double ya =y1+(U*(y2-y1));
                double za =z1+(U*(z2-z1));
                if(K==1)
                {
                        return(xa);
                }
                if (K==2)
                {
                        return(ya);
                }
                if(K==3)
                {
                        return(za);
                }

        }


                                        //defining fuction to take 4 points and to give the
intersect from extrapolation
    double Cross(int K,
                                                        double x1, double y1, double z1,
                                                        double x2, double y2, double z2,
                                                        double x3, double y3, double z3,
                                                        double x4, double y4, double z4)
    {
    double x13=x1-x3;
    double y13=y1-y3;
    double z13=z1-z3;

    double x43=x4-x3;
    double y43=y4-y3;
    double z43=z4-z3;

    double x21=x2-x1;
    double y21=y2-y1;
    double z21=z2-z1;

    double d1343 = (x13*x43)+(y13*y43)+(z13*z43);
    double d4321 = (x43*x21)+(y43*y21)+(z43*z21);
    double d1321 = (x13*x21)+(y13*y21)+(z13*z21);
    double d4343 = (x43*x43)+(y43*y43)+(z43*z43);
    double d2121 = (x21*x21)+(y21*y21)+(z21*z21);


    double denum;
    double number;
    number = (d1343*d4321)-(d1321*d4343);
    denum = (d2121*d4343)-(d4321*d4321);
    double Ua=number / denum ;
    double Ub=((d1343+(d4321*Ua))/d4343);

    double xa=x1+(Ua*x21);
```

```
    double ya=y1+(Ua*y21);
    double za=z1+(Ua*z21);

    double xb=x3+(Ub*x43);
    double yb=y3+(Ub*y43);
    double zb=z3+(Ub*z43);

    //average point
    double xab=(xa+xb)/2;
    double yab=(ya+yb)/2;
    double zab=(za+zb)/2;

    if(K==1)
    {return (xab);}
    if(K==2)
    {return (yab);}
    if(K==3)
    {return (zab);}
    }

        /// <summary>
        /// Summary for Form1
        ///
        /// WARNING: If you change the name of this class, you will need to change the
        ///            'Resource File Name' property for the managed resource compiler
tool
        ///          associated with all .resx files this class depends on.  Otherwise,
        ///          the designers will not be able to interact properly with localized
        ///          resources associated with this form.
        /// </summary>
        public ref class Form1 : public System::Windows::Forms::Form
        {
        public:
                Form1(void)
                {
                        InitializeComponent();
                }

        protected:
                /// <summary>
                /// Clean up any resources being used.
                /// </summary>
                ~Form1()
                {
                        if (components)
                        {
                                delete components;
                        }
                }
        private: System::Windows::Forms::MenuStrip^  menuStrip1;
        protected:
        private: System::Windows::Forms::ToolStripMenuItem^  fileToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  openToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  exitToolStripMenuItem;
        private: System::Windows::Forms::PictureBox^  ImageBox;
        private: System::Windows::Forms::ToolStripMenuItem^  zoomToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  ZoomOutToolStripMenuItem;
        private:                                System::Windows::Forms::ToolStripMenuItem^
OriginalZoomToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  ZoomInToolStripMenuItem;
        private: System::Windows::Forms::SplitContainer^  splitContainer1;
        private: System::Windows::Forms::GroupBox^  groupBox2;
```

```
	private: System::Windows::Forms::Label^  label3;
	private: System::Windows::Forms::TextBox^  txtZoom;
	private: System::Windows::Forms::Label^  label2;
	private: System::Windows::Forms::Label^  label1;
	private: System::Windows::Forms::TextBox^  txtHeight;
	private: System::Windows::Forms::TextBox^  txtWidth;
	private: System::Windows::Forms::GroupBox^  groupBox7;
	private: System::Windows::Forms::Label^  label8;
	private: System::Windows::Forms::TextBox^  txtCountStangeShapeEruptionCount;


	private: System::Windows::Forms::PictureBox^  CountStrangeShapeEruptionLight;
	private: System::Windows::Forms::GroupBox^  groupBox1;
	private: System::Windows::Forms::Label^  label5;
	private: System::Windows::Forms::Label^  label4;
	private: System::Windows::Forms::TextBox^  txtXClick;
	private: System::Windows::Forms::TextBox^  txtYClick;

	private: System::Windows::Forms::Label^  label7;
	private: System::Windows::Forms::TextBox^  txtWhiskerStatsCount;
	private: System::Windows::Forms::GroupBox^  groupBox3;
	private: System::Windows::Forms::PictureBox^  CalibrationLight;
	private: System::Windows::Forms::TextBox^  txtRealLength;
	private: System::Windows::Forms::Label^  label6;

	private: System::Windows::Forms::GroupBox^  groupBox5;




	private: System::Windows::Forms::PictureBox^  MeasureWhiskerLight;
	private: System::Windows::Forms::GroupBox^  groupBox4;

	private: System::Windows::Forms::PictureBox^  AreaMeasureLight;
	private: System::Windows::Forms::TextBox^  txtArea;
	private: System::Windows::Forms::Button^  btnCalibrate;
	private: System::Windows::Forms::Button^  btnMeasureAreaMeasure;
	private: System::Windows::Forms::Button^  btnCountEruptionStart;
	private: System::Windows::Forms::Button^  btnCountErutionStop;
	private: System::Windows::Forms::Button^  btnMeasureWhiskerStart;
	private: System::Windows::Forms::Button^  btnMeasureWhiskerRecord;
	private: System::Windows::Forms::Button^  btnMeasureWhiskerUndo;
	private: System::Windows::Forms::Button^  btnMeasureWhiskerStop;
	private: System::Windows::Forms::SplitContainer^  splitContainer2;
	private: System::Windows::Forms::PictureBox^  ImageBox2;
	private: System::Windows::Forms::ToolStripMenuItem^  toolStripMenuItem1;
	private:                              System::Windows::Forms::ToolStripMenuItem^
zoomImage2ToolStripMenuItem;
	private: System::Windows::Forms::ToolStripMenuItem^  zoomOutToolStripMenuItem1;
	private:                              System::Windows::Forms::ToolStripMenuItem^
originalZoomToolStripMenuItem1;
	private: System::Windows::Forms::ToolStripMenuItem^  zoomInToolStripMenuItem1;
	private: System::Windows::Forms::GroupBox^  groupBox8;
	private: System::Windows::Forms::Label^  label11;
	private: System::Windows::Forms::TextBox^  txtImage2Zoom;

	private: System::Windows::Forms::TextBox^  txtImage2Width;
	private: System::Windows::Forms::Label^  label9;
	private: System::Windows::Forms::Label^  label10;
	private: System::Windows::Forms::TextBox^  txtImage2Height;
```

```
    private: System::Windows::Forms::GroupBox^  groupBox9;
    private: System::Windows::Forms::GroupBox^  groupBox6;
    private: System::Windows::Forms::Label^  label12;
    private: System::Windows::Forms::Label^  label13;
    private: System::Windows::Forms::TextBox^  txtXClickImage2;
    private: System::Windows::Forms::TextBox^  txtYClickImage2;
    private: System::Windows::Forms::GroupBox^  groupBox10;
    private: System::Windows::Forms::GroupBox^  groupBox11;
private: System::Windows::Forms::TextBox^  txtWhiskerStatsCountImage2;

    private: System::Windows::Forms::Label^  label14;
private: System::Windows::Forms::Button^  MeasureWhiskerStopImage2;



private: System::Windows::Forms::Button^  MeasureWhiskerRecordImage2;

private: System::Windows::Forms::Button^  MeasureWhiskerStartImage2;

private: System::Windows::Forms::PictureBox^  MeasureWhiskerLightImage2;
private: System::Windows::Forms::Button^  MeasureWhiskerUndoImage2;
private: System::Windows::Forms::GroupBox^  groupBox12;

private: System::Windows::Forms::GroupBox^  groupBox13;
private: System::Windows::Forms::Label^  label16;
private: System::Windows::Forms::Label^  label15;
private: System::Windows::Forms::Button^  btn3dCalculate;
private: System::Windows::Forms::TextBox^  txtImage1Beta;





private: System::Windows::Forms::TextBox^  txtImage1Alpha;
private: System::Windows::Forms::GroupBox^  groupBox14;
private: System::Windows::Forms::TextBox^  txtImage2Beta;

private: System::Windows::Forms::TextBox^  txtImage2Alpha;


private: System::Windows::Forms::Label^  label18;
private: System::Windows::Forms::Label^  label17;
private: System::Windows::Forms::ToolStripMenuItem^  toolStripMenuItem2;
private: System::Windows::Forms::GroupBox^  groupBox15;
private: System::Windows::Forms::TextBox^  txtSampleInfoCoating;

private: System::Windows::Forms::TextBox^  txtSampleInfoSubstrate;
private: System::Windows::Forms::Label^  label21;
private: System::Windows::Forms::Label^  label20;
private: System::Windows::Forms::TextBox^  txtSampleInfoName;
private: System::Windows::Forms::Label^  label19;
private: System::Windows::Forms::TextBox^  txtSampleInfoVoltage;
private: System::Windows::Forms::TextBox^  txtSampleInfoElectrolyte;
private: System::Windows::Forms::TextBox^  txtSampleInfoCoatingThickness;
private: System::Windows::Forms::Label^  label24;
private: System::Windows::Forms::Label^  label23;
private: System::Windows::Forms::Label^  label22;

private: System::Windows::Forms::TextBox^  txtSampleInfoDateImage;
private: System::Windows::Forms::TextBox^  txtSampleInfoDateCreation;

private: System::Windows::Forms::Label^  label26;
```

```
private: System::Windows::Forms::Label^  label25;
private: System::Windows::Forms::TextBox^  txtSampleInfoTimeCreation;
private: System::Windows::Forms::Label^  label27;
private: System::Windows::Forms::TextBox^  txtSampleInfoTimeImage;
private: System::Windows::Forms::Label^  label28;
private: System::Windows::Forms::TextBox^  txtSampleInfoComments;
private: System::Windows::Forms::Label^  label29;
private: System::Windows::Forms::PictureBox^  CalculateLight;
private: System::Windows::Forms::GroupBox^  groupBox17;
private: System::Windows::Forms::Button^  btnSetRefPoint2;

private: System::Windows::Forms::PictureBox^  setPoint2Light;

private: System::Windows::Forms::GroupBox^  groupBox16;
private: System::Windows::Forms::Button^  btnSetRefPoint1;
private: System::Windows::Forms::PictureBox^  setPoint1Light;
private: System::Windows::Forms::GroupBox^  groupBox18;


private: System::Windows::Forms::Button^  btnDiameteMeasure;
private: System::Windows::Forms::PictureBox^  WhiskerSiameterLight;
private: System::Windows::Forms::Button^  btnDiameterRecord;




private: System::Windows::Forms::Button^  btnWhiskerDiameterUndo;
private: System::Windows::Forms::ToolStripMenuItem^  toolStripMenuItem3;
private: System::Windows::Forms::TextBox^  txtSampleInfoCurrent;
private: System::Windows::Forms::Label^  label30;
private: System::Windows::Forms::ToolStripMenuItem^  toolStripMenuItem4;
private: System::Windows::Forms::TextBox^  txt3DCalculationWorkingDistance;
private: System::Windows::Forms::Label^  label31;
private: System::Windows::Forms::TextBox^  txtImage2Name;

private: System::Windows::Forms::Label^  label33;
private: System::Windows::Forms::TextBox^  txtImage1Name;
private: System::Windows::Forms::Label^  label32;
private: System::Windows::Forms::GroupBox^  groupBox19;


private: System::Windows::Forms::Label^  label34;
private: System::Windows::Forms::TextBox^  txtCountWhiskerCount;
private: System::Windows::Forms::PictureBox^  CountWhiskerLight;
private: System::Windows::Forms::Button^  btnCountWhiskerStop;
private: System::Windows::Forms::Button^  btnCountWhiskerStart;

    private: System::ComponentModel::IContainer^  components;
    private:
            /// <summary>
            /// Required designer variable.
            /// </summary>
```

```
#pragma region Windows Form Designer generated code
		/// <summary>
		/// Required method for Designer support - do not modify
		/// the contents of this method with the code editor.
		/// </summary>
		void InitializeComponent(void)
		{
			this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
			this->fileToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->openToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->toolStripMenuItem1 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->toolStripMenuItem2 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->toolStripMenuItem3 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->toolStripMenuItem4 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->exitToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->zoomToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->ZoomOutToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->OriginalZoomToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->ZoomInToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->zoomImage2ToolStripMenuItem = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->zoomOutToolStripMenuItem1 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->originalZoomToolStripMenuItem1 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->zoomInToolStripMenuItem1 = (gcnew 
System::Windows::Forms::ToolStripMenuItem());
			this->ImageBox = (gcnew System::Windows::Forms::PictureBox());
			this->splitContainer1 = (gcnew 
System::Windows::Forms::SplitContainer());
			this->groupBox15 = (gcnew System::Windows::Forms::GroupBox());
			this->txtSampleInfoTimeImage = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoDateImage = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoTimeCreation = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoDateCreation = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoComments = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoCurrent = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoVoltage = (gcnew 
System::Windows::Forms::TextBox());
			this->txtSampleInfoElectrolyte = (gcnew 
System::Windows::Forms::TextBox());
			this->label28 = (gcnew System::Windows::Forms::Label());
```

```
			this->txtSampleInfoCoatingThickness		=		(gcnew
System::Windows::Forms::TextBox());
			this->label29 = (gcnew System::Windows::Forms::Label());
			this->label26 = (gcnew System::Windows::Forms::Label());
			this->txtSampleInfoCoating		=		(gcnew
System::Windows::Forms::TextBox());
			this->label27 = (gcnew System::Windows::Forms::Label());
			this->label25 = (gcnew System::Windows::Forms::Label());
			this->txtImage2Name = (gcnew System::Windows::Forms::TextBox());
			this->txtSampleInfoSubstrate		=		(gcnew
System::Windows::Forms::TextBox());
			this->label30 = (gcnew System::Windows::Forms::Label());
			this->label24 = (gcnew System::Windows::Forms::Label());
			this->label23 = (gcnew System::Windows::Forms::Label());
			this->label22 = (gcnew System::Windows::Forms::Label());
			this->label33 = (gcnew System::Windows::Forms::Label());
			this->label21 = (gcnew System::Windows::Forms::Label());
			this->txtImage1Name = (gcnew System::Windows::Forms::TextBox());
			this->label20 = (gcnew System::Windows::Forms::Label());
			this->label32 = (gcnew System::Windows::Forms::Label());
			this->txtSampleInfoName		=		(gcnew
System::Windows::Forms::TextBox());
			this->label19 = (gcnew System::Windows::Forms::Label());
			this->groupBox10 = (gcnew System::Windows::Forms::GroupBox());
			this->groupBox17 = (gcnew System::Windows::Forms::GroupBox());
			this->btnSetRefPoint2 = (gcnew System::Windows::Forms::Button());
			this->setPoint2Light		=		(gcnew
System::Windows::Forms::PictureBox());
			this->groupBox11 = (gcnew System::Windows::Forms::GroupBox());
			this->MeasureWhiskerUndoImage2		=		(gcnew
System::Windows::Forms::Button());
			this->txtWhiskerStatsCountImage2		=		(gcnew
System::Windows::Forms::TextBox());
			this->label14 = (gcnew System::Windows::Forms::Label());
			this->MeasureWhiskerStopImage2		=		(gcnew
System::Windows::Forms::Button());
			this->MeasureWhiskerRecordImage2		=		(gcnew
System::Windows::Forms::Button());
			this->MeasureWhiskerStartImage2		=		(gcnew
System::Windows::Forms::Button());
			this->MeasureWhiskerLightImage2		=		(gcnew
System::Windows::Forms::PictureBox());
			this->groupBox4 = (gcnew System::Windows::Forms::GroupBox());
			this->btnMeasureAreaMeasure		=		(gcnew
System::Windows::Forms::Button());
			this->AreaMeasureLight		=		(gcnew
System::Windows::Forms::PictureBox());
			this->txtArea = (gcnew System::Windows::Forms::TextBox());
			this->groupBox12 = (gcnew System::Windows::Forms::GroupBox());
			this->txt3DCalculationWorkingDistance		=		(gcnew
System::Windows::Forms::TextBox());
			this->label31 = (gcnew System::Windows::Forms::Label());
			this->groupBox14 = (gcnew System::Windows::Forms::GroupBox());
			this->txtImage2Beta = (gcnew System::Windows::Forms::TextBox());
			this->txtImage2Alpha = (gcnew System::Windows::Forms::TextBox());
			this->label18 = (gcnew System::Windows::Forms::Label());
			this->label17 = (gcnew System::Windows::Forms::Label());
			this->CalculateLight		=		(gcnew
System::Windows::Forms::PictureBox());
			this->groupBox13 = (gcnew System::Windows::Forms::GroupBox());
			this->txtImage1Beta = (gcnew System::Windows::Forms::TextBox());
			this->txtImage1Alpha = (gcnew System::Windows::Forms::TextBox());
```

```
                    this->label16 = (gcnew System::Windows::Forms::Label());
                    this->label15 = (gcnew System::Windows::Forms::Label());
                    this->btn3dCalculate = (gcnew System::Windows::Forms::Button());
                    this->groupBox9 = (gcnew System::Windows::Forms::GroupBox());
                    this->groupBox18 = (gcnew System::Windows::Forms::GroupBox());
                    this->btnWhiskerDiameterUndo              =              (gcnew
System::Windows::Forms::Button());
                    this->btnDiameterRecord                   =              (gcnew
System::Windows::Forms::Button());
                    this->btnDiameteMeasure                   =              (gcnew
System::Windows::Forms::Button());
                    this->WhiskerSiameterLight                =              (gcnew
System::Windows::Forms::PictureBox());
                    this->groupBox16 = (gcnew System::Windows::Forms::GroupBox());
                    this->btnSetRefPoint1 = (gcnew System::Windows::Forms::Button());
                    this->setPoint1Light                      =              (gcnew
System::Windows::Forms::PictureBox());
                    this->groupBox5 = (gcnew System::Windows::Forms::GroupBox());
                    this->txtWhiskerStatsCount                =              (gcnew
System::Windows::Forms::TextBox());
                    this->label7 = (gcnew System::Windows::Forms::Label());
                    this->btnMeasureWhiskerStop               =              (gcnew
System::Windows::Forms::Button());
                    this->btnMeasureWhiskerUndo               =              (gcnew
System::Windows::Forms::Button());
                    this->btnMeasureWhiskerRecord             =              (gcnew
System::Windows::Forms::Button());
                    this->btnMeasureWhiskerStart              =              (gcnew
System::Windows::Forms::Button());
                    this->MeasureWhiskerLight                 =              (gcnew
System::Windows::Forms::PictureBox());
                    this->groupBox19 = (gcnew System::Windows::Forms::GroupBox());
                    this->btnCountWhiskerStop                 =              (gcnew
System::Windows::Forms::Button());
                    this->btnCountWhiskerStart                =              (gcnew
System::Windows::Forms::Button());
                    this->label34 = (gcnew System::Windows::Forms::Label());
                    this->txtCountWhiskerCount                =              (gcnew
System::Windows::Forms::TextBox());
                    this->CountWhiskerLight                   =              (gcnew
System::Windows::Forms::PictureBox());
                    this->groupBox7 = (gcnew System::Windows::Forms::GroupBox());
                    this->btnCountErutionStop                 =              (gcnew
System::Windows::Forms::Button());
                    this->btnCountEruptionStart               =              (gcnew
System::Windows::Forms::Button());
                    this->label8 = (gcnew System::Windows::Forms::Label());
                    this->txtCountStangeShapeEruptionCount         =         (gcnew
System::Windows::Forms::TextBox());
                    this->CountStrangeShapeEruptionLight           =         (gcnew
System::Windows::Forms::PictureBox());
                    this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
                    this->btnCalibrate = (gcnew System::Windows::Forms::Button());
                    this->CalibrationLight                    =              (gcnew
System::Windows::Forms::PictureBox());
                    this->txtRealLength = (gcnew System::Windows::Forms::TextBox());
                    this->label6 = (gcnew System::Windows::Forms::Label());
                    this->groupBox8 = (gcnew System::Windows::Forms::GroupBox());
                    this->groupBox6 = (gcnew System::Windows::Forms::GroupBox());
                    this->label12 = (gcnew System::Windows::Forms::Label());
                    this->label13 = (gcnew System::Windows::Forms::Label());
```

```
			this->txtXClickImage2                    =                   (gcnew
System::Windows::Forms::TextBox());
			this->txtYClickImage2                    =                   (gcnew
System::Windows::Forms::TextBox());
			this->label11 = (gcnew System::Windows::Forms::Label());
			this->txtImage2Zoom = (gcnew System::Windows::Forms::TextBox());
			this->txtImage2Width = (gcnew System::Windows::Forms::TextBox());
			this->label9 = (gcnew System::Windows::Forms::Label());
			this->label10 = (gcnew System::Windows::Forms::Label());
			this->txtImage2Height                    =                   (gcnew
System::Windows::Forms::TextBox());
			this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
			this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
			this->label5 = (gcnew System::Windows::Forms::Label());
			this->label4 = (gcnew System::Windows::Forms::Label());
			this->txtXClick = (gcnew System::Windows::Forms::TextBox());
			this->txtYClick = (gcnew System::Windows::Forms::TextBox());
			this->label3 = (gcnew System::Windows::Forms::Label());
			this->txtZoom = (gcnew System::Windows::Forms::TextBox());
			this->label2 = (gcnew System::Windows::Forms::Label());
			this->label1 = (gcnew System::Windows::Forms::Label());
			this->txtHeight = (gcnew System::Windows::Forms::TextBox());
			this->txtWidth = (gcnew System::Windows::Forms::TextBox());
			this->splitContainer2                    =                   (gcnew
System::Windows::Forms::SplitContainer());
			this->ImageBox2 = (gcnew System::Windows::Forms::PictureBox());
			this->menuStrip1->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->ImageBox))->BeginInit();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer1))->BeginInit();
			this->splitContainer1->Panel1->SuspendLayout();
			this->splitContainer1->Panel2->SuspendLayout();
			this->splitContainer1->SuspendLayout();
			this->groupBox15->SuspendLayout();
			this->groupBox10->SuspendLayout();
			this->groupBox17->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->setPoint2Light))->BeginInit();
			this->groupBox11->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->MeasureWhiskerLightImage2))->BeginInit();
			this->groupBox4->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->AreaMeasureLight))->BeginInit();
			this->groupBox12->SuspendLayout();
			this->groupBox14->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CalculateLight))->BeginInit();
			this->groupBox13->SuspendLayout();
			this->groupBox9->SuspendLayout();
			this->groupBox18->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->WhiskerSiameterLight))->BeginInit();
			this->groupBox16->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->setPoint1Light))->BeginInit();
			this->groupBox5->SuspendLayout();
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->MeasureWhiskerLight))->BeginInit();
			this->groupBox19->SuspendLayout();
```

```
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CountWhiskerLight))->BeginInit();
                    this->groupBox7->SuspendLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CountStrangeShapeEruptionLight))->BeginInit();
                    this->groupBox3->SuspendLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CalibrationLight))->BeginInit();
                    this->groupBox8->SuspendLayout();
                    this->groupBox6->SuspendLayout();
                    this->groupBox2->SuspendLayout();
                    this->groupBox1->SuspendLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer2))->BeginInit();
                    this->splitContainer2->Panel1->SuspendLayout();
                    this->splitContainer2->Panel2->SuspendLayout();
                    this->splitContainer2->SuspendLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->ImageBox2))->BeginInit();
                    this->SuspendLayout();
                    // 
                    // menuStrip1
                    // 
                    this->menuStrip1->Items->AddRange(gcnew                cli::array<
System::Windows::Forms::ToolStripItem^  >(3) {this->fileToolStripMenuItem,
                           this->zoomToolStripMenuItem,                     this-
>zoomImage2ToolStripMenuItem});
                    this->menuStrip1->Location = System::Drawing::Point(0, 0);
                    this->menuStrip1->Name = L"menuStrip1";
                    this->menuStrip1->Size = System::Drawing::Size(797, 24);
                    this->menuStrip1->TabIndex = 0;
                    this->menuStrip1->Text = L"menuStrip1";
                    // 
                    // fileToolStripMenuItem
                    // 
                    this->fileToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^  >(6) {this->openToolStripMenuItem,
                           this->toolStripMenuItem1,  this->toolStripMenuItem2,  this-
>toolStripMenuItem3, this->toolStripMenuItem4, this->exitToolStripMenuItem});
                    this->fileToolStripMenuItem->Name = L"fileToolStripMenuItem";
                    this->fileToolStripMenuItem->Size    =    System::Drawing::Size(35,
20);
                    this->fileToolStripMenuItem->Text = L"File";
                    // 
                    // openToolStripMenuItem
                    // 
                    this->openToolStripMenuItem->Name = L"openToolStripMenuItem";
                    this->openToolStripMenuItem->Size   =   System::Drawing::Size(154,
22);
                    this->openToolStripMenuItem->Text = L"Open Image1";
                    this->openToolStripMenuItem->Click           +=            gcnew
System::EventHandler(this, &Form1::openToolStripMenuItem_Click);
                    // 
                    // toolStripMenuItem1
                    // 
                    this->toolStripMenuItem1->Name = L"toolStripMenuItem1";
                    this->toolStripMenuItem1->Size = System::Drawing::Size(154, 22);
                    this->toolStripMenuItem1->Text = L"Open Image2";
                    this->toolStripMenuItem1->Click            +=            gcnew
System::EventHandler(this, &Form1::toolStripMenuItem1_Click);
                    // 
                    // toolStripMenuItem2
```

```
                    //
                    this->toolStripMenuItem2->Name = L"toolStripMenuItem2";
                    this->toolStripMenuItem2->Size = System::Drawing::Size(154, 22);
                    this->toolStripMenuItem2->Text = L"Save 3D data As";
                    this->toolStripMenuItem2->Click              +=              gcnew
System::EventHandler(this, &Form1::toolStripMenuItem2_Click);
                    //
                    // toolStripMenuItem3
                    //
                    this->toolStripMenuItem3->Name = L"toolStripMenuItem3";
                    this->toolStripMenuItem3->Size = System::Drawing::Size(154, 22);
                    this->toolStripMenuItem3->Text = L"Save Image1 As";
                    this->toolStripMenuItem3->Click              +=              gcnew
System::EventHandler(this, &Form1::toolStripMenuItem3_Click);
                    //
                    // toolStripMenuItem4
                    //
                    this->toolStripMenuItem4->Name = L"toolStripMenuItem4";
                    this->toolStripMenuItem4->Size = System::Drawing::Size(154, 22);
                    this->toolStripMenuItem4->Text = L"Save Image2 As";
                    this->toolStripMenuItem4->Click              +=              gcnew
System::EventHandler(this, &Form1::toolStripMenuItem4_Click);
                    //
                    // exitToolStripMenuItem
                    //
                    this->exitToolStripMenuItem->Name = L"exitToolStripMenuItem";
                    this->exitToolStripMenuItem->Size   =   System::Drawing::Size(154,
22);
                    this->exitToolStripMenuItem->Text = L"Exit";
                    this->exitToolStripMenuItem->Click              +=              gcnew
System::EventHandler(this, &Form1::exitToolStripMenuItem_Click);
                    //
                    // zoomToolStripMenuItem
                    //
                    this->zoomToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array<       System::Windows::Forms::ToolStripItem^             >(3)       {this-
>ZoomOutToolStripMenuItem,
                           this->OriginalZoomToolStripMenuItem,                   this-
>ZoomInToolStripMenuItem});
                    this->zoomToolStripMenuItem->Name = L"zoomToolStripMenuItem";
                    this->zoomToolStripMenuItem->Size    =    System::Drawing::Size(84,
20);
                    this->zoomToolStripMenuItem->Text = L"Zoom Image1";
                    //
                    // ZoomOutToolStripMenuItem
                    //
                    this->ZoomOutToolStripMenuItem->Name                              =
L"ZoomOutToolStripMenuItem";
                    this->ZoomOutToolStripMenuItem->Size  =  System::Drawing::Size(166,
22);
                    this->ZoomOutToolStripMenuItem->Text = L"Zoom Out";
                    this->ZoomOutToolStripMenuItem->Click           +=            gcnew
System::EventHandler(this, &Form1::x01ToolStripMenuItem_Click);
                    //
                    // OriginalZoomToolStripMenuItem
                    //
                    this->OriginalZoomToolStripMenuItem->Name                         =
L"OriginalZoomToolStripMenuItem";
                    this->OriginalZoomToolStripMenuItem->Size                         =
System::Drawing::Size(166, 22);
                    this->OriginalZoomToolStripMenuItem->Text       =       L"Original
Dimensions";
```

```
                    this->OriginalZoomToolStripMenuItem->Click        +=        gcnew
System::EventHandler(this, &Form1::x1ToolStripMenuItem_Click);
                    //
                    // ZoomInToolStripMenuItem
                    //
                    this->ZoomInToolStripMenuItem->Name = L"ZoomInToolStripMenuItem";
                    this->ZoomInToolStripMenuItem->Size  =  System::Drawing::Size(166,
22);
                    this->ZoomInToolStripMenuItem->Text = L"Zoom In";
                    this->ZoomInToolStripMenuItem->Click           +=          gcnew
System::EventHandler(this, &Form1::x10ToolStripMenuItem_Click);
                    //
                    // zoomImage2ToolStripMenuItem
                    //
                    this->zoomImage2ToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array<       System::Windows::Forms::ToolStripItem^            >(3)       {this-
>zoomOutToolStripMenuItem1,
                           this->originalZoomToolStripMenuItem1,                this-
>zoomInToolStripMenuItem1});
                    this->zoomImage2ToolStripMenuItem->Name                         =
L"zoomImage2ToolStripMenuItem";
                    this->zoomImage2ToolStripMenuItem->Size                         =
System::Drawing::Size(84, 20);
                    this->zoomImage2ToolStripMenuItem->Text = L"Zoom Image2";
                    //
                    // zoomOutToolStripMenuItem1
                    //
                    this->zoomOutToolStripMenuItem1->Name                           =
L"zoomOutToolStripMenuItem1";
                    this->zoomOutToolStripMenuItem1->Size                           =
System::Drawing::Size(139, 22);
                    this->zoomOutToolStripMenuItem1->Text = L"Zoom Out";
                    this->zoomOutToolStripMenuItem1->Click          +=           gcnew
System::EventHandler(this, &Form1::zoomOutToolStripMenuItem1_Click);
                    //
                    // originalZoomToolStripMenuItem1
                    //
                    this->originalZoomToolStripMenuItem1->Name                      =
L"originalZoomToolStripMenuItem1";
                    this->originalZoomToolStripMenuItem1->Size                      =
System::Drawing::Size(139, 22);
                    this->originalZoomToolStripMenuItem1->Text = L"Original Zoom";
                    this->originalZoomToolStripMenuItem1->Click       +=        gcnew
System::EventHandler(this, &Form1::originalZoomToolStripMenuItem1_Click);
                    //
                    // zoomInToolStripMenuItem1
                    //
                    this->zoomInToolStripMenuItem1->Name                            =
L"zoomInToolStripMenuItem1";
                    this->zoomInToolStripMenuItem1->Size  =  System::Drawing::Size(139,
22);
                    this->zoomInToolStripMenuItem1->Text = L"Zoom in";
                    this->zoomInToolStripMenuItem1->Click           +=           gcnew
System::EventHandler(this, &Form1::zoomInToolStripMenuItem1_Click);
                    //
                    // ImageBox
                    //
                    this->ImageBox->Cursor = System::Windows::Forms::Cursors::Cross;
                    this->ImageBox->Location = System::Drawing::Point(0, 0);
                    this->ImageBox->Name = L"ImageBox";
                    this->ImageBox->Size = System::Drawing::Size(618, 465);
```

```
			this->ImageBox->SizeMode = 
System::Windows::Forms::PictureBoxSizeMode::StretchImage;
			this->ImageBox->TabIndex = 1;
			this->ImageBox->TabStop = false;
			this->ImageBox->Click += gcnew System::EventHandler(this, 
&Form1::ImageBox_Click);
			this->ImageBox->Paint += gcnew 
System::Windows::Forms::PaintEventHandler(this, &Form1::ImageBox_Paint);
			// 
			// splitContainer1
			// 
			this->splitContainer1->Dock = 
System::Windows::Forms::DockStyle::Fill;
			this->splitContainer1->Location = System::Drawing::Point(0, 24);
			this->splitContainer1->Name = L"splitContainer1";
			// 
			// splitContainer1.Panel1
			// 
			this->splitContainer1->Panel1->AutoScroll = true;
			this->splitContainer1->Panel1->Controls->Add(this->groupBox15);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox10);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox4);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox12);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox9);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox8);
			this->splitContainer1->Panel1->Controls->Add(this->groupBox2);
			// 
			// splitContainer1.Panel2
			// 
			this->splitContainer1->Panel2->AutoScroll = true;
			this->splitContainer1->Panel2->BackColor = 
System::Drawing::SystemColors::Info;
			this->splitContainer1->Panel2->Controls->Add(this-
>splitContainer2);
			this->splitContainer1->Size = System::Drawing::Size(797, 970);
			this->splitContainer1->SplitterDistance = 687;
			this->splitContainer1->TabIndex = 0;
			// 
			// groupBox15
			// 
			this->groupBox15->Controls->Add(this->txtSampleInfoTimeImage);
			this->groupBox15->Controls->Add(this->txtSampleInfoDateImage);
			this->groupBox15->Controls->Add(this->txtSampleInfoTimeCreation);
			this->groupBox15->Controls->Add(this->txtSampleInfoDateCreation);
			this->groupBox15->Controls->Add(this->txtSampleInfoComments);
			this->groupBox15->Controls->Add(this->txtSampleInfoCurrent);
			this->groupBox15->Controls->Add(this->txtSampleInfoVoltage);
			this->groupBox15->Controls->Add(this->txtSampleInfoElectrolyte);
			this->groupBox15->Controls->Add(this->label28);
			this->groupBox15->Controls->Add(this-
>txtSampleInfoCoatingThickness);
			this->groupBox15->Controls->Add(this->label29);
			this->groupBox15->Controls->Add(this->label26);
			this->groupBox15->Controls->Add(this->txtSampleInfoCoating);
			this->groupBox15->Controls->Add(this->label27);
			this->groupBox15->Controls->Add(this->label25);
			this->groupBox15->Controls->Add(this->txtImage2Name);
			this->groupBox15->Controls->Add(this->txtSampleInfoSubstrate);
			this->groupBox15->Controls->Add(this->label30);
			this->groupBox15->Controls->Add(this->label24);
			this->groupBox15->Controls->Add(this->label23);
			this->groupBox15->Controls->Add(this->label22);
```

```
                    this->groupBox15->Controls->Add(this->label33);
                    this->groupBox15->Controls->Add(this->label21);
                    this->groupBox15->Controls->Add(this->txtImage1Name);
                    this->groupBox15->Controls->Add(this->label20);
                    this->groupBox15->Controls->Add(this->label32);
                    this->groupBox15->Controls->Add(this->txtSampleInfoName);
                    this->groupBox15->Controls->Add(this->label19);
                    this->groupBox15->Location = System::Drawing::Point(456, 3);
                    this->groupBox15->Name = L"groupBox15";
                    this->groupBox15->Size = System::Drawing::Size(246, 420);
                    this->groupBox15->TabIndex = 0;
                    this->groupBox15->TabStop = false;
                    this->groupBox15->Text   =   L"Sample   Infomation     (DON\'T   USE
COMMAS)";
                    //
                    // txtSampleInfoTimeImage
                    //
                    this->txtSampleInfoTimeImage->Location                           =
System::Drawing::Point(115, 352);
                    this->txtSampleInfoTimeImage->Name = L"txtSampleInfoTimeImage";
                    this->txtSampleInfoTimeImage->Size   =   System::Drawing::Size(100,
20);
                    this->txtSampleInfoTimeImage->TabIndex = 10;
                    this->txtSampleInfoTimeImage->Text = L"24hr";
                    this->txtSampleInfoTimeImage->MouseClick         +=         gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::txtSampleInfoTimeImage_MouseClick);
                    //
                    // txtSampleInfoDateImage
                    //
                    this->txtSampleInfoDateImage->Location                           =
System::Drawing::Point(6, 352);
                    this->txtSampleInfoDateImage->Name = L"txtSampleInfoDateImage";
                    this->txtSampleInfoDateImage->Size   =   System::Drawing::Size(100,
20);
                    this->txtSampleInfoDateImage->TabIndex = 9;
                    this->txtSampleInfoDateImage->Text = L"dd/mm/yyyy";
                    this->txtSampleInfoDateImage->MouseClick         +=         gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::txtSampleInfoDateImage_MouseClick);
                    //
                    // txtSampleInfoTimeCreation
                    //
                    this->txtSampleInfoTimeCreation->Location                        =
System::Drawing::Point(115, 307);
                    this->txtSampleInfoTimeCreation->Name                            =
L"txtSampleInfoTimeCreation";
                    this->txtSampleInfoTimeCreation->Size                            =
System::Drawing::Size(100, 20);
                    this->txtSampleInfoTimeCreation->TabIndex = 8;
                    this->txtSampleInfoTimeCreation->Text = L"24hr";
                    this->txtSampleInfoTimeCreation->MouseClick       +=        gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::txtSampleInfoTimeCreation_MouseClick);
                    //
                    // txtSampleInfoDateCreation
                    //
                    this->txtSampleInfoDateCreation->Location                        =
System::Drawing::Point(6, 307);
                    this->txtSampleInfoDateCreation->Name                            =
L"txtSampleInfoDateCreation";
```

```
			this->txtSampleInfoDateCreation->Size                    =
System::Drawing::Size(100, 20);
			this->txtSampleInfoDateCreation->TabIndex = 7;
			this->txtSampleInfoDateCreation->Text = L"dd/mm/yyyy";
			this->txtSampleInfoDateCreation->MouseClick       +=       gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::txtSampleInfoDateCreation_MouseClick);
			//
			// txtSampleInfoComments
			//
			this->txtSampleInfoComments->Location  =  System::Drawing::Point(6,
391);
			this->txtSampleInfoComments->Name = L"txtSampleInfoComments";
			this->txtSampleInfoComments->ScrollBars                    =
System::Windows::Forms::ScrollBars::Horizontal;
			this->txtSampleInfoComments->Size   =   System::Drawing::Size(234,
20);
			this->txtSampleInfoComments->TabIndex = 11;
			//
			// txtSampleInfoCurrent
			//
			this->txtSampleInfoCurrent->Location  =  System::Drawing::Point(6,
266);
			this->txtSampleInfoCurrent->Name = L"txtSampleInfoCurrent";
			this->txtSampleInfoCurrent->ScrollBars                     =
System::Windows::Forms::ScrollBars::Horizontal;
			this->txtSampleInfoCurrent->Size    =    System::Drawing::Size(234,
20);
			this->txtSampleInfoCurrent->TabIndex = 6;
			//
			// txtSampleInfoVoltage
			//
			this->txtSampleInfoVoltage->Location  =  System::Drawing::Point(6,
227);
			this->txtSampleInfoVoltage->Name = L"txtSampleInfoVoltage";
			this->txtSampleInfoVoltage->ScrollBars                     =
System::Windows::Forms::ScrollBars::Horizontal;
			this->txtSampleInfoVoltage->Size    =    System::Drawing::Size(234,
20);
			this->txtSampleInfoVoltage->TabIndex = 5;
			//
			// txtSampleInfoElectrolyte
			//
			this->txtSampleInfoElectrolyte->Location                   =
System::Drawing::Point(6, 188);
			this->txtSampleInfoElectrolyte->Name                       =
L"txtSampleInfoElectrolyte";
			this->txtSampleInfoElectrolyte->Size = System::Drawing::Size(100,
20);
			this->txtSampleInfoElectrolyte->TabIndex = 4;
			//
			// label28
			//
			this->label28->AutoSize = true;
			this->label28->Location = System::Drawing::Point(115, 336);
			this->label28->Name = L"label28";
			this->label28->Size = System::Drawing::Size(74, 13);
			this->label28->TabIndex = 11;
			this->label28->Text = L"Time of Image";
			//
			// txtSampleInfoCoatingThickness
			//
```

```
                    this->txtSampleInfoCoatingThickness->Location                    =
System::Drawing::Point(6, 149);
                    this->txtSampleInfoCoatingThickness->Name                        =
L"txtSampleInfoCoatingThickness";
                    this->txtSampleInfoCoatingThickness->Size                        =
System::Drawing::Size(100, 20);
                    this->txtSampleInfoCoatingThickness->TabIndex = 3;
                    //
                    // label29
                    //
                    this->label29->AutoSize = true;
                    this->label29->Location = System::Drawing::Point(6, 375);
                    this->label29->Name = L"label29";
                    this->label29->Size = System::Drawing::Size(56, 13);
                    this->label29->TabIndex = 0;
                    this->label29->Text = L"Comments";
                    //
                    // label26
                    //
                    this->label26->AutoSize = true;
                    this->label26->Location = System::Drawing::Point(6, 336);
                    this->label26->Name = L"label26";
                    this->label26->Size = System::Drawing::Size(74, 13);
                    this->label26->TabIndex = 0;
                    this->label26->Text = L"Date of Image";
                    //
                    // txtSampleInfoCoating
                    //
                    this->txtSampleInfoCoating->Location  =  System::Drawing::Point(6,
110);
                    this->txtSampleInfoCoating->Name = L"txtSampleInfoCoating";
                    this->txtSampleInfoCoating->Size    =    System::Drawing::Size(100,
20);
                    this->txtSampleInfoCoating->TabIndex = 2;
                    //
                    // label27
                    //
                    this->label27->AutoSize = true;
                    this->label27->Location = System::Drawing::Point(115, 291);
                    this->label27->Name = L"label27";
                    this->label27->Size = System::Drawing::Size(84, 13);
                    this->label27->TabIndex = 0;
                    this->label27->Text = L"Time of Creation";
                    //
                    // label25
                    //
                    this->label25->AutoSize = true;
                    this->label25->Location = System::Drawing::Point(6, 291);
                    this->label25->Name = L"label25";
                    this->label25->Size = System::Drawing::Size(84, 13);
                    this->label25->TabIndex = 0;
                    this->label25->Text = L"Date of Creation";
                    //
                    // txtImage2Name
                    //
                    this->txtImage2Name->Location = System::Drawing::Point(112, 71);
                    this->txtImage2Name->Name = L"txtImage2Name";
                    this->txtImage2Name->ReadOnly = true;
                    this->txtImage2Name->Size = System::Drawing::Size(100, 20);
                    this->txtImage2Name->TabIndex = 1;
                    //
                    // txtSampleInfoSubstrate
```

```
                    //
                    this->txtSampleInfoSubstrate->Location                          =
System::Drawing::Point(6, 71);
                    this->txtSampleInfoSubstrate->Name = L"txtSampleInfoSubstrate";
                    this->txtSampleInfoSubstrate->Size  =  System::Drawing::Size(100,
20);
                    this->txtSampleInfoSubstrate->TabIndex = 1;
                    //
                    // label30
                    //
                    this->label30->AutoSize = true;
                    this->label30->Location = System::Drawing::Point(6, 250);
                    this->label30->Name = L"label30";
                    this->label30->Size = System::Drawing::Size(116, 13);
                    this->label30->TabIndex = 0;
                    this->label30->Text = L"Current Density Pattern";
                    //
                    // label24
                    //
                    this->label24->AutoSize = true;
                    this->label24->Location = System::Drawing::Point(6, 211);
                    this->label24->Name = L"label24";
                    this->label24->Size = System::Drawing::Size(80, 13);
                    this->label24->TabIndex = 0;
                    this->label24->Text = L"Voltage Pattern";
                    //
                    // label23
                    //
                    this->label23->AutoSize = true;
                    this->label23->Location = System::Drawing::Point(6, 172);
                    this->label23->Name = L"label23";
                    this->label23->Size = System::Drawing::Size(56, 13);
                    this->label23->TabIndex = 0;
                    this->label23->Text = L"Electrolyte";
                    //
                    // label22
                    //
                    this->label22->AutoSize = true;
                    this->label22->Location = System::Drawing::Point(6, 133);
                    this->label22->Name = L"label22";
                    this->label22->Size = System::Drawing::Size(89, 13);
                    this->label22->TabIndex = 0;
                    this->label22->Text = L"Coating Tickness";
                    //
                    // label33
                    //
                    this->label33->AutoSize = true;
                    this->label33->Location = System::Drawing::Point(112, 55);
                    this->label33->Name = L"label33";
                    this->label33->Size = System::Drawing::Size(70, 13);
                    this->label33->TabIndex = 0;
                    this->label33->Text = L"Image2Name";
                    //
                    // label21
                    //
                    this->label21->AutoSize = true;
                    this->label21->Location = System::Drawing::Point(6, 94);
                    this->label21->Name = L"label21";
                    this->label21->Size = System::Drawing::Size(83, 13);
                    this->label21->TabIndex = 0;
                    this->label21->Text = L"Coating Material";
                    //
```

```
// txtImage1Name
//
this->txtImage1Name->Location = System::Drawing::Point(112, 32);
this->txtImage1Name->Name = L"txtImage1Name";
this->txtImage1Name->ReadOnly = true;
this->txtImage1Name->Size = System::Drawing::Size(100, 20);
this->txtImage1Name->TabIndex = 0;
//
// label20
//
this->label20->AutoSize = true;
this->label20->Location = System::Drawing::Point(6, 55);
this->label20->Name = L"label20";
this->label20->Size = System::Drawing::Size(52, 13);
this->label20->TabIndex = 0;
this->label20->Text = L"Substrate";
//
// label32
//
this->label32->AutoSize = true;
this->label32->Location = System::Drawing::Point(112, 16);
this->label32->Name = L"label32";
this->label32->Size = System::Drawing::Size(70, 13);
this->label32->TabIndex = 0;
this->label32->Text = L"Image1Name";
//
// txtSampleInfoName
//
this->txtSampleInfoName->Location    =    System::Drawing::Point(6,
32);

this->txtSampleInfoName->Name = L"txtSampleInfoName";
this->txtSampleInfoName->Size = System::Drawing::Size(100, 20);
this->txtSampleInfoName->TabIndex = 0;
//
// label19
//
this->label19->AutoSize = true;
this->label19->Location = System::Drawing::Point(6, 16);
this->label19->Name = L"label19";
this->label19->Size = System::Drawing::Size(35, 13);
this->label19->TabIndex = 0;
this->label19->Text = L"Name";
//
// groupBox10
//
this->groupBox10->Controls->Add(this->groupBox17);
this->groupBox10->Controls->Add(this->groupBox11);
this->groupBox10->Location = System::Drawing::Point(204, 329);
this->groupBox10->Name = L"groupBox10";
this->groupBox10->Size = System::Drawing::Size(195, 195);
this->groupBox10->TabIndex = 4;
this->groupBox10->TabStop = false;
this->groupBox10->Text = L"Image2 Controls";
//
// groupBox17
//
this->groupBox17->Controls->Add(this->btnSetRefPoint2);
this->groupBox17->Controls->Add(this->setPoint2Light);
this->groupBox17->Location = System::Drawing::Point(6, 23);
this->groupBox17->Name = L"groupBox17";
this->groupBox17->Size = System::Drawing::Size(181, 55);
this->groupBox17->TabIndex = 4;
```

```
                    this->groupBox17->TabStop = false;
                    this->groupBox17->Text = L"Set Referecne Point";
                    //
                    // btnSetRefPoint2
                    //
                    this->btnSetRefPoint2->Location = System::Drawing::Point(11, 19);
                    this->btnSetRefPoint2->Name = L"btnSetRefPoint2";
                    this->btnSetRefPoint2->Size = System::Drawing::Size(75, 23);
                    this->btnSetRefPoint2->TabIndex = 0;
                    this->btnSetRefPoint2->Text = L"Set Point";
                    this->btnSetRefPoint2->UseVisualStyleBackColor = true;
                    this->btnSetRefPoint2->Click  +=  gcnew  System::EventHandler(this,
&Form1::btnSetRefPoint2_Click);
                    //
                    // setPoint2Light
                    //
                    this->setPoint2Light->BackColor = System::Drawing::Color::Red;
                    this->setPoint2Light->Location = System::Drawing::Point(112, 0);
                    this->setPoint2Light->Name = L"setPoint2Light";
                    this->setPoint2Light->Size = System::Drawing::Size(15, 15);
                    this->setPoint2Light->TabIndex = 14;
                    this->setPoint2Light->TabStop = false;
                    //
                    // groupBox11
                    //
                    this->groupBox11->Controls->Add(this->MeasureWhiskerUndoImage2);
                    this->groupBox11->Controls->Add(this-
>txtWhiskerStatsCountImage2);
                    this->groupBox11->Controls->Add(this->label14);
                    this->groupBox11->Controls->Add(this->MeasureWhiskerStopImage2);
                    this->groupBox11->Controls->Add(this-
>MeasureWhiskerRecordImage2);
                    this->groupBox11->Controls->Add(this->MeasureWhiskerStartImage2);
                    this->groupBox11->Controls->Add(this->MeasureWhiskerLightImage2);
                    this->groupBox11->Location = System::Drawing::Point(6, 80);
                    this->groupBox11->Name = L"groupBox11";
                    this->groupBox11->Size = System::Drawing::Size(181, 111);
                    this->groupBox11->TabIndex = 0;
                    this->groupBox11->TabStop = false;
                    this->groupBox11->Text = L"Measure Whisker";
                    //
                    // MeasureWhiskerUndoImage2
                    //
                    this->MeasureWhiskerUndoImage2->Location                          =
System::Drawing::Point(87, 53);
                    this->MeasureWhiskerUndoImage2->Name                              =
L"MeasureWhiskerUndoImage2";
                    this->MeasureWhiskerUndoImage2->Size  =  System::Drawing::Size(75,
23);
                    this->MeasureWhiskerUndoImage2->TabIndex = 5;
                    this->MeasureWhiskerUndoImage2->TabStop = false;
                    this->MeasureWhiskerUndoImage2->Text = L"Undo";
                    this->MeasureWhiskerUndoImage2->UseVisualStyleBackColor = true;
                    this->MeasureWhiskerUndoImage2->Click          +=           gcnew
System::EventHandler(this, &Form1::MeasureWhiskerUndoImage2_Click);
                    //
                    // txtWhiskerStatsCountImage2
                    //
                    this->txtWhiskerStatsCountImage2->Location                        =
System::Drawing::Point(52, 82);
                    this->txtWhiskerStatsCountImage2->Name                            =
L"txtWhiskerStatsCountImage2";
```

```
			this->txtWhiskerStatsCountImage2->ReadOnly = true;
			this->txtWhiskerStatsCountImage2->Size                    =
System::Drawing::Size(61, 20);
			this->txtWhiskerStatsCountImage2->TabIndex = 0;
			this->txtWhiskerStatsCountImage2->TabStop = false;
			//
			// label14
			//
			this->label14->AutoSize = true;
			this->label14->Location = System::Drawing::Point(11, 85);
			this->label14->Name = L"label14";
			this->label14->Size = System::Drawing::Size(35, 13);
			this->label14->TabIndex = 1;
			this->label14->Text = L"Count";
			//
			// MeasureWhiskerStopImage2
			//
			this->MeasureWhiskerStopImage2->Location                    =
System::Drawing::Point(87, 24);
			this->MeasureWhiskerStopImage2->Name                        =
L"MeasureWhiskerStopImage2";
			this->MeasureWhiskerStopImage2->Size  =  System::Drawing::Size(75,
23);
			this->MeasureWhiskerStopImage2->TabIndex = 2;
			this->MeasureWhiskerStopImage2->TabStop = false;
			this->MeasureWhiskerStopImage2->Text = L"Stop";
			this->MeasureWhiskerStopImage2->UseVisualStyleBackColor = true;
			this->MeasureWhiskerStopImage2->Click          +=          gcnew
System::EventHandler(this, &Form1::button1_Click);
			//
			// MeasureWhiskerRecordImage2
			//
			this->MeasureWhiskerRecordImage2->Location                  =
System::Drawing::Point(9, 53);
			this->MeasureWhiskerRecordImage2->Name                      =
L"MeasureWhiskerRecordImage2";
			this->MeasureWhiskerRecordImage2->Size                      =
System::Drawing::Size(75, 23);
			this->MeasureWhiskerRecordImage2->TabIndex = 3;
			this->MeasureWhiskerRecordImage2->TabStop = false;
			this->MeasureWhiskerRecordImage2->Text = L"Record";
			this->MeasureWhiskerRecordImage2->UseVisualStyleBackColor = true;
			this->MeasureWhiskerRecordImage2->Click        +=          gcnew
System::EventHandler(this, &Form1::button3_Click);
			//
			// MeasureWhiskerStartImage2
			//
			this->MeasureWhiskerStartImage2->Location                   =
System::Drawing::Point(9, 24);
			this->MeasureWhiskerStartImage2->Name                       =
L"MeasureWhiskerStartImage2";
			this->MeasureWhiskerStartImage2->Size  =  System::Drawing::Size(75,
23);
			this->MeasureWhiskerStartImage2->TabIndex = 4;
			this->MeasureWhiskerStartImage2->TabStop = false;
			this->MeasureWhiskerStartImage2->Text = L"Start/Pause";
			this->MeasureWhiskerStartImage2->UseVisualStyleBackColor = true;
			this->MeasureWhiskerStartImage2->Click         +=          gcnew
System::EventHandler(this, &Form1::button4_Click);
			//
			// MeasureWhiskerLightImage2
			//
```

```
                this->MeasureWhiskerLightImage2->BackColor                 =
System::Drawing::Color::Red;
                this->MeasureWhiskerLightImage2->Location                  =
System::Drawing::Point(160, 0);
                this->MeasureWhiskerLightImage2->Name                      =
L"MeasureWhiskerLightImage2";
                this->MeasureWhiskerLightImage2->Size = System::Drawing::Size(15,
15);
                this->MeasureWhiskerLightImage2->TabIndex = 14;
                this->MeasureWhiskerLightImage2->TabStop = false;
                //
                // groupBox4
                //
                this->groupBox4->Controls->Add(this->btnMeasureAreaMeasure);
                this->groupBox4->Controls->Add(this->AreaMeasureLight);
                this->groupBox4->Controls->Add(this->txtArea);
                this->groupBox4->Location = System::Drawing::Point(3, 841);
                this->groupBox4->Name = L"groupBox4";
                this->groupBox4->Size = System::Drawing::Size(181, 53);
                this->groupBox4->TabIndex = 0;
                this->groupBox4->TabStop = false;
                this->groupBox4->Text = L"Measure Area";
                //
                // btnMeasureAreaMeasure
                //
                this->btnMeasureAreaMeasure->Location = System::Drawing::Point(9,
19);
                this->btnMeasureAreaMeasure->Name = L"btnMeasureAreaMeasure";
                this->btnMeasureAreaMeasure->Size    =    System::Drawing::Size(75,
23);
                this->btnMeasureAreaMeasure->TabIndex = 1;
                this->btnMeasureAreaMeasure->TabStop = false;
                this->btnMeasureAreaMeasure->Text = L"Measure/Stop";
                this->btnMeasureAreaMeasure->UseVisualStyleBackColor = true;
                this->btnMeasureAreaMeasure->Click            +=            gcnew
System::EventHandler(this, &Form1::btnMeasureAreaMeasure_Click);
                //
                // AreaMeasureLight
                //
                this->AreaMeasureLight->BackColor = System::Drawing::Color::Red;
                this->AreaMeasureLight->Location    =    System::Drawing::Point(160,
0);
                this->AreaMeasureLight->Name = L"AreaMeasureLight";
                this->AreaMeasureLight->Size = System::Drawing::Size(15, 15);
                this->AreaMeasureLight->TabIndex = 14;
                this->AreaMeasureLight->TabStop = false;
                //
                // txtArea
                //
                this->txtArea->Location = System::Drawing::Point(103, 19);
                this->txtArea->Name = L"txtArea";
                this->txtArea->ReadOnly = true;
                this->txtArea->Size = System::Drawing::Size(61, 20);
                this->txtArea->TabIndex = 0;
                this->txtArea->TabStop = false;
                //
                // groupBox12
                //
                this->groupBox12->Controls->Add(this-
>txt3DCalculationWorkingDistance);
                this->groupBox12->Controls->Add(this->label31);
                this->groupBox12->Controls->Add(this->groupBox14);
```

```
                    this->groupBox12->Controls->Add(this->CalculateLight);
                    this->groupBox12->Controls->Add(this->groupBox13);
                    this->groupBox12->Controls->Add(this->btn3dCalculate);
                    this->groupBox12->Location = System::Drawing::Point(204, 3);
                    this->groupBox12->Name = L"groupBox12";
                    this->groupBox12->Size = System::Drawing::Size(246, 320);
                    this->groupBox12->TabIndex = 1;
                    this->groupBox12->TabStop = false;
                    this->groupBox12->Text = L"3D Calculation";
                    //
                    // txt3DCalculationWorkingDistance
                    //
                    this->txt3DCalculationWorkingDistance->Location                  =
System::Drawing::Point(6, 35);
                    this->txt3DCalculationWorkingDistance->Name                      =
L"txt3DCalculationWorkingDistance";
                    this->txt3DCalculationWorkingDistance->Size                      =
System::Drawing::Size(149, 20);
                    this->txt3DCalculationWorkingDistance->TabIndex = 17;
                    //
                    // label31
                    //
                    this->label31->AutoSize = true;
                    this->label31->Location = System::Drawing::Point(6, 19);
                    this->label31->Name = L"label31";
                    this->label31->Size = System::Drawing::Size(115, 13);
                    this->label31->TabIndex = 16;
                    this->label31->Text = L"Working distance (mm)";
                    //
                    // groupBox14
                    //
                    this->groupBox14->Controls->Add(this->txtImage2Beta);
                    this->groupBox14->Controls->Add(this->txtImage2Alpha);
                    this->groupBox14->Controls->Add(this->label18);
                    this->groupBox14->Controls->Add(this->label17);
                    this->groupBox14->Location = System::Drawing::Point(6, 175);
                    this->groupBox14->Name = L"groupBox14";
                    this->groupBox14->Size = System::Drawing::Size(231, 108);
                    this->groupBox14->TabIndex = 0;
                    this->groupBox14->TabStop = false;
                    this->groupBox14->Text = L"Image2";
                    //
                    // txtImage2Beta
                    //
                    this->txtImage2Beta->Location = System::Drawing::Point(82, 82);
                    this->txtImage2Beta->Name = L"txtImage2Beta";
                    this->txtImage2Beta->Size = System::Drawing::Size(100, 20);
                    this->txtImage2Beta->TabIndex = 14;
                    //
                    // txtImage2Alpha
                    //
                    this->txtImage2Alpha->Location = System::Drawing::Point(82, 39);
                    this->txtImage2Alpha->Name = L"txtImage2Alpha";
                    this->txtImage2Alpha->Size = System::Drawing::Size(100, 20);
                    this->txtImage2Alpha->TabIndex = 13;
                    //
                    // label18
                    //
                    this->label18->AutoSize = true;
                    this->label18->Location = System::Drawing::Point(2, 62);
                    this->label18->Name = L"label18";
                    this->label18->Size = System::Drawing::Size(229, 26);
```

```
            this->label18->TabIndex = 0;
            this->label18->Text = L"Rotatoin of axes around y axis clockwise
(deg) \r\nBeta=";
            //
            // label17
            //
            this->label17->AutoSize = true;
            this->label17->Location = System::Drawing::Point(2, 24);
            this->label17->Name = L"label17";
            this->label17->Size = System::Drawing::Size(229, 26);
            this->label17->TabIndex = 0;
            this->label17->Text = L"Rotatoin of axes around y axis clockwise
(deg) \r\nAlpha=";
            //
            // CalculateLight
            //
            this->CalculateLight->BackColor = System::Drawing::Color::Red;
            this->CalculateLight->Location = System::Drawing::Point(88, 294);
            this->CalculateLight->Name = L"CalculateLight";
            this->CalculateLight->Size = System::Drawing::Size(15, 15);
            this->CalculateLight->TabIndex = 14;
            this->CalculateLight->TabStop = false;
            //
            // groupBox13
            //
            this->groupBox13->Controls->Add(this->txtImage1Beta);
            this->groupBox13->Controls->Add(this->txtImage1Alpha);
            this->groupBox13->Controls->Add(this->label16);
            this->groupBox13->Controls->Add(this->label15);
            this->groupBox13->Location = System::Drawing::Point(6, 61);
            this->groupBox13->Name = L"groupBox13";
            this->groupBox13->Size = System::Drawing::Size(229, 108);
            this->groupBox13->TabIndex = 0;
            this->groupBox13->TabStop = false;
            this->groupBox13->Text = L"Image1";
            //
            // txtImage1Beta
            //
            this->txtImage1Beta->Location = System::Drawing::Point(82, 82);
            this->txtImage1Beta->Name = L"txtImage1Beta";
            this->txtImage1Beta->Size = System::Drawing::Size(100, 20);
            this->txtImage1Beta->TabIndex = 12;
            //
            // txtImage1Alpha
            //
            this->txtImage1Alpha->Location = System::Drawing::Point(82, 39);
            this->txtImage1Alpha->Name = L"txtImage1Alpha";
            this->txtImage1Alpha->Size = System::Drawing::Size(100, 20);
            this->txtImage1Alpha->TabIndex = 11;
            //
            // label16
            //
            this->label16->AutoSize = true;
            this->label16->Location = System::Drawing::Point(4, 62);
            this->label16->Name = L"label16";
            this->label16->Size = System::Drawing::Size(229, 26);
            this->label16->TabIndex = 2;
            this->label16->Text = L"Rotatoin of axes around y axis clockwise
(deg) \r\nBeta=";
            //
            // label15
            //
```

```
                    this->label15->AutoSize = true;
                    this->label15->Location = System::Drawing::Point(4, 16);
                    this->label15->Name = L"label15";
                    this->label15->Size = System::Drawing::Size(229, 26);
                    this->label15->TabIndex = 0;
                    this->label15->Text = L"Rotatoin of axes around y axis clockwise
(deg) \r\nAlpha=";
                    //
                    // btn3dCalculate
                    //
                    this->btn3dCalculate->Location = System::Drawing::Point(6, 289);
                    this->btn3dCalculate->Name = L"btn3dCalculate";
                    this->btn3dCalculate->Size = System::Drawing::Size(75, 23);
                    this->btn3dCalculate->TabIndex = 15;
                    this->btn3dCalculate->Text = L"Calculate";
                    this->btn3dCalculate->UseVisualStyleBackColor = true;
                    this->btn3dCalculate->Click  +=  gcnew  System::EventHandler(this,
&Form1::btn3dCalculate_Click);
                    //
                    // groupBox9
                    //
                    this->groupBox9->Controls->Add(this->groupBox18);
                    this->groupBox9->Controls->Add(this->groupBox16);
                    this->groupBox9->Controls->Add(this->groupBox5);
                    this->groupBox9->Controls->Add(this->groupBox19);
                    this->groupBox9->Controls->Add(this->groupBox7);
                    this->groupBox9->Controls->Add(this->groupBox3);
                    this->groupBox9->Location = System::Drawing::Point(3, 289);
                    this->groupBox9->Name = L"groupBox9";
                    this->groupBox9->Size = System::Drawing::Size(195, 611);
                    this->groupBox9->TabIndex = 1;
                    this->groupBox9->TabStop = false;
                    this->groupBox9->Text = L"Image1 Controls";
                    //
                    // groupBox18
                    //
                    this->groupBox18->Controls->Add(this->btnWhiskerDiameterUndo);
                    this->groupBox18->Controls->Add(this->btnDiameterRecord);
                    this->groupBox18->Controls->Add(this->btnDiameteMeasure);
                    this->groupBox18->Controls->Add(this->WhiskerSiameterLight);
                    this->groupBox18->Location = System::Drawing::Point(6, 285);
                    this->groupBox18->Name = L"groupBox18";
                    this->groupBox18->Size = System::Drawing::Size(181, 80);
                    this->groupBox18->TabIndex = 5;
                    this->groupBox18->TabStop = false;
                    this->groupBox18->Text = L"Whisker Diameter";
                    //
                    // btnWhiskerDiameterUndo
                    //
                    this->btnWhiskerDiameterUndo->Location                          =
System::Drawing::Point(89, 51);
                    this->btnWhiskerDiameterUndo->Name = L"btnWhiskerDiameterUndo";
                    this->btnWhiskerDiameterUndo->Size   =   System::Drawing::Size(75,
23);
                    this->btnWhiskerDiameterUndo->TabIndex = 16;
                    this->btnWhiskerDiameterUndo->Text = L"Undo";
                    this->btnWhiskerDiameterUndo->UseVisualStyleBackColor = true;
                    this->btnWhiskerDiameterUndo->Click           +=           gcnew
System::EventHandler(this, &Form1::btnWhiskerDiameterUndo_Click);
                    //
                    // btnDiameterRecord
                    //
```

```
                    this->btnDiameterRecord->Location   =   System::Drawing::Point(90,
19);
                    this->btnDiameterRecord->Name = L"btnDiameterRecord";
                    this->btnDiameterRecord->Size = System::Drawing::Size(75, 23);
                    this->btnDiameterRecord->TabIndex = 15;
                    this->btnDiameterRecord->Text = L"Record";
                    this->btnDiameterRecord->UseVisualStyleBackColor = true;
                    this->btnDiameterRecord->Click              +=              gcnew
System::EventHandler(this, &Form1::btnDiameterRecord_Click);
                    //
                    // btnDiameteMeasure
                    //
                    this->btnDiameteMeasure->Location    =    System::Drawing::Point(9,
19);
                    this->btnDiameteMeasure->Name = L"btnDiameteMeasure";
                    this->btnDiameteMeasure->Size = System::Drawing::Size(75, 23);
                    this->btnDiameteMeasure->TabIndex = 4;
                    this->btnDiameteMeasure->Text = L"Measure";
                    this->btnDiameteMeasure->UseVisualStyleBackColor = true;
                    this->btnDiameteMeasure->Click              +=              gcnew
System::EventHandler(this, &Form1::btnDiameteMeasure_Click);
                    //
                    // WhiskerSiameterLight
                    //
                    this->WhiskerSiameterLight->BackColor                          =
System::Drawing::Color::Red;
                    this->WhiskerSiameterLight->Location                           =
System::Drawing::Point(160, 0);
                    this->WhiskerSiameterLight->Name = L"WhiskerSiameterLight";
                    this->WhiskerSiameterLight->Size = System::Drawing::Size(15, 15);
                    this->WhiskerSiameterLight->TabIndex = 14;
                    this->WhiskerSiameterLight->TabStop = false;
                    //
                    // groupBox16
                    //
                    this->groupBox16->Controls->Add(this->btnSetRefPoint1);
                    this->groupBox16->Controls->Add(this->setPoint1Light);
                    this->groupBox16->Location = System::Drawing::Point(6, 107);
                    this->groupBox16->Name = L"groupBox16";
                    this->groupBox16->Size = System::Drawing::Size(181, 55);
                    this->groupBox16->TabIndex = 4;
                    this->groupBox16->TabStop = false;
                    this->groupBox16->Text = L"Set Referecne Point";
                    //
                    // btnSetRefPoint1
                    //
                    this->btnSetRefPoint1->Location = System::Drawing::Point(11, 19);
                    this->btnSetRefPoint1->Name = L"btnSetRefPoint1";
                    this->btnSetRefPoint1->Size = System::Drawing::Size(75, 23);
                    this->btnSetRefPoint1->TabIndex = 0;
                    this->btnSetRefPoint1->Text = L"Set Point";
                    this->btnSetRefPoint1->UseVisualStyleBackColor = true;
                    this->btnSetRefPoint1->Click  +=  gcnew  System::EventHandler(this,
&Form1::btnSetRefPoint1_Click);
                    //
                    // setPoint1Light
                    //
                    this->setPoint1Light->BackColor = System::Drawing::Color::Red;
                    this->setPoint1Light->Location = System::Drawing::Point(160, 0);
                    this->setPoint1Light->Name = L"setPoint1Light";
                    this->setPoint1Light->Size = System::Drawing::Size(15, 15);
                    this->setPoint1Light->TabIndex = 14;
```

```
                    this->setPoint1Light->TabStop = false;
                    //
                    // groupBox5
                    //
                    this->groupBox5->Controls->Add(this->txtWhiskerStatsCount);
                    this->groupBox5->Controls->Add(this->label7);
                    this->groupBox5->Controls->Add(this->btnMeasureWhiskerStop);
                    this->groupBox5->Controls->Add(this->btnMeasureWhiskerUndo);
                    this->groupBox5->Controls->Add(this->btnMeasureWhiskerRecord);
                    this->groupBox5->Controls->Add(this->btnMeasureWhiskerStart);
                    this->groupBox5->Controls->Add(this->MeasureWhiskerLight);
                    this->groupBox5->Location = System::Drawing::Point(6, 168);
                    this->groupBox5->Name = L"groupBox5";
                    this->groupBox5->Size = System::Drawing::Size(181, 111);
                    this->groupBox5->TabIndex = 2;
                    this->groupBox5->TabStop = false;
                    this->groupBox5->Text = L"Measure Whisker";
                    //
                    // txtWhiskerStatsCount
                    //
                    this->txtWhiskerStatsCount->Location  =  System::Drawing::Point(52,
82);
                    this->txtWhiskerStatsCount->Name = L"txtWhiskerStatsCount";
                    this->txtWhiskerStatsCount->ReadOnly = true;
                    this->txtWhiskerStatsCount->Size = System::Drawing::Size(61, 20);
                    this->txtWhiskerStatsCount->TabIndex = 0;
                    this->txtWhiskerStatsCount->TabStop = false;
                    //
                    // label7
                    //
                    this->label7->AutoSize = true;
                    this->label7->Location = System::Drawing::Point(11, 85);
                    this->label7->Name = L"label7";
                    this->label7->Size = System::Drawing::Size(35, 13);
                    this->label7->TabIndex = 1;
                    this->label7->Text = L"Count";
                    //
                    // btnMeasureWhiskerStop
                    //
                    this->btnMeasureWhiskerStop->Location                            =
System::Drawing::Point(87, 24);
                    this->btnMeasureWhiskerStop->Name = L"btnMeasureWhiskerStop";
                    this->btnMeasureWhiskerStop->Size    =    System::Drawing::Size(75,
23);
                    this->btnMeasureWhiskerStop->TabIndex = 2;
                    this->btnMeasureWhiskerStop->TabStop = false;
                    this->btnMeasureWhiskerStop->Text = L"Stop";
                    this->btnMeasureWhiskerStop->UseVisualStyleBackColor = true;
                    this->btnMeasureWhiskerStop->Click            +=             gcnew
System::EventHandler(this, &Form1::btnMeasureWhiskerStop_Click);
                    //
                    // btnMeasureWhiskerUndo
                    //
                    this->btnMeasureWhiskerUndo->Location                            =
System::Drawing::Point(87, 53);
                    this->btnMeasureWhiskerUndo->Name = L"btnMeasureWhiskerUndo";
                    this->btnMeasureWhiskerUndo->Size    =    System::Drawing::Size(75,
23);
                    this->btnMeasureWhiskerUndo->TabIndex = 3;
                    this->btnMeasureWhiskerUndo->TabStop = false;
                    this->btnMeasureWhiskerUndo->Text = L"Undo";
                    this->btnMeasureWhiskerUndo->UseVisualStyleBackColor = true;
```

```
			this->btnMeasureWhiskerUndo->Click += gcnew System::EventHandler(this, &Form1::btnMeasureWhiskerUndo_Click);
			//
			// btnMeasureWhiskerRecord
			//
			this->btnMeasureWhiskerRecord->Location = System::Drawing::Point(9, 53);
			this->btnMeasureWhiskerRecord->Name = L"btnMeasureWhiskerRecord";
			this->btnMeasureWhiskerRecord->Size = System::Drawing::Size(75, 23);
			this->btnMeasureWhiskerRecord->TabIndex = 4;
			this->btnMeasureWhiskerRecord->TabStop = false;
			this->btnMeasureWhiskerRecord->Text = L"Record";
			this->btnMeasureWhiskerRecord->UseVisualStyleBackColor = true;
			this->btnMeasureWhiskerRecord->Click += gcnew System::EventHandler(this, &Form1::btnMeasureWhiskerRecord_Click);
			//
			// btnMeasureWhiskerStart
			//
			this->btnMeasureWhiskerStart->Location = System::Drawing::Point(9, 24);
			this->btnMeasureWhiskerStart->Name = L"btnMeasureWhiskerStart";
			this->btnMeasureWhiskerStart->Size = System::Drawing::Size(75, 23);
			this->btnMeasureWhiskerStart->TabIndex = 5;
			this->btnMeasureWhiskerStart->TabStop = false;
			this->btnMeasureWhiskerStart->Text = L"Start/Pause";
			this->btnMeasureWhiskerStart->UseVisualStyleBackColor = true;
			this->btnMeasureWhiskerStart->Click += gcnew System::EventHandler(this, &Form1::btnMeasureWhiskerStart_Click);
			//
			// MeasureWhiskerLight
			//
			this->MeasureWhiskerLight->BackColor = System::Drawing::Color::Red;
			this->MeasureWhiskerLight->Location = System::Drawing::Point(160, 0);
			this->MeasureWhiskerLight->Name = L"MeasureWhiskerLight";
			this->MeasureWhiskerLight->Size = System::Drawing::Size(15, 15);
			this->MeasureWhiskerLight->TabIndex = 14;
			this->MeasureWhiskerLight->TabStop = false;
			//
			// groupBox19
			//
			this->groupBox19->Controls->Add(this->btnCountWhiskerStop);
			this->groupBox19->Controls->Add(this->btnCountWhiskerStart);
			this->groupBox19->Controls->Add(this->label34);
			this->groupBox19->Controls->Add(this->txtCountWhiskerCount);
			this->groupBox19->Controls->Add(this->CountWhiskerLight);
			this->groupBox19->Location = System::Drawing::Point(6, 371);
			this->groupBox19->Name = L"groupBox19";
			this->groupBox19->Size = System::Drawing::Size(181, 80);
			this->groupBox19->TabIndex = 3;
			this->groupBox19->TabStop = false;
			this->groupBox19->Text = L"Count Whisker";
			//
			// btnCountWhiskerStop
			//
			this->btnCountWhiskerStop->Location = System::Drawing::Point(87, 19);
			this->btnCountWhiskerStop->Name = L"btnCountWhiskerStop";
			this->btnCountWhiskerStop->Size = System::Drawing::Size(75, 23);
```

```
                    this->btnCountWhiskerStop->TabIndex = 16;
                    this->btnCountWhiskerStop->Text = L"Stop";
                    this->btnCountWhiskerStop->UseVisualStyleBackColor = true;
                    this->btnCountWhiskerStop->Click           +=             gcnew
System::EventHandler(this, &Form1::btnCountWhiskerStop_Click);
                    //
                    // btnCountWhiskerStart
                    //
                    this->btnCountWhiskerStart->Location  =  System::Drawing::Point(6,
19);
                    this->btnCountWhiskerStart->Name = L"btnCountWhiskerStart";
                    this->btnCountWhiskerStart->Size = System::Drawing::Size(75, 23);
                    this->btnCountWhiskerStart->TabIndex = 15;
                    this->btnCountWhiskerStart->Text = L"Start";
                    this->btnCountWhiskerStart->UseVisualStyleBackColor = true;
                    this->btnCountWhiskerStart->Click           +=            gcnew
System::EventHandler(this, &Form1::btnCountWhiskerStart_Click);
                    //
                    // label34
                    //
                    this->label34->AutoSize = true;
                    this->label34->Location = System::Drawing::Point(6, 48);
                    this->label34->Name = L"label34";
                    this->label34->Size = System::Drawing::Size(35, 13);
                    this->label34->TabIndex = 1;
                    this->label34->Text = L"Count";
                    //
                    // txtCountWhiskerCount
                    //
                    this->txtCountWhiskerCount->Location  =  System::Drawing::Point(47,
48);
                    this->txtCountWhiskerCount->Name = L"txtCountWhiskerCount";
                    this->txtCountWhiskerCount->ReadOnly = true;
                    this->txtCountWhiskerCount->Size = System::Drawing::Size(61, 20);
                    this->txtCountWhiskerCount->TabIndex = 0;
                    this->txtCountWhiskerCount->TabStop = false;
                    //
                    // CountWhiskerLight
                    //
                    this->CountWhiskerLight->BackColor = System::Drawing::Color::Red;
                    this->CountWhiskerLight->Location   =   System::Drawing::Point(160,
0);
                    this->CountWhiskerLight->Name = L"CountWhiskerLight";
                    this->CountWhiskerLight->Size = System::Drawing::Size(15, 15);
                    this->CountWhiskerLight->TabIndex = 14;
                    this->CountWhiskerLight->TabStop = false;
                    //
                    // groupBox7
                    //
                    this->groupBox7->Controls->Add(this->btnCountErutionStop);
                    this->groupBox7->Controls->Add(this->btnCountEruptionStart);
                    this->groupBox7->Controls->Add(this->label8);
                    this->groupBox7->Controls->Add(this-
>txtCountStangeShapeEruptionCount);
                    this->groupBox7->Controls->Add(this-
>CountStrangeShapeEruptionLight);
                    this->groupBox7->Location = System::Drawing::Point(0, 466);
                    this->groupBox7->Name = L"groupBox7";
                    this->groupBox7->Size = System::Drawing::Size(181, 80);
                    this->groupBox7->TabIndex = 3;
                    this->groupBox7->TabStop = false;
                    this->groupBox7->Text = L"Count Stange Shape Eruption";
```

```
                    //
                    // btnCountErutionStop
                    //
                    this->btnCountErutionStop->Location  =  System::Drawing::Point(90,
19);
                    this->btnCountErutionStop->Name = L"btnCountErutionStop";
                    this->btnCountErutionStop->Size = System::Drawing::Size(75, 23);
                    this->btnCountErutionStop->TabIndex = 2;
                    this->btnCountErutionStop->TabStop = false;
                    this->btnCountErutionStop->Text = L"Stop";
                    this->btnCountErutionStop->UseVisualStyleBackColor = true;
                    this->btnCountErutionStop->Click             +=             gcnew
System::EventHandler(this, &Form1::btnCountErutionStop_Click);
                    //
                    // btnCountEruptionStart
                    //
                    this->btnCountEruptionStart->Location  =  System::Drawing::Point(9,
19);
                    this->btnCountEruptionStart->Name = L"btnCountEruptionStart";
                    this->btnCountEruptionStart->Size    =    System::Drawing::Size(75,
23);
                    this->btnCountEruptionStart->TabIndex = 3;
                    this->btnCountEruptionStart->TabStop = false;
                    this->btnCountEruptionStart->Text = L"Start";
                    this->btnCountEruptionStart->UseVisualStyleBackColor = true;
                    this->btnCountEruptionStart->Click            +=            gcnew
System::EventHandler(this, &Form1::btnCountEruptionStart_Click);
                    //
                    // label8
                    //
                    this->label8->AutoSize = true;
                    this->label8->Location = System::Drawing::Point(6, 48);
                    this->label8->Name = L"label8";
                    this->label8->Size = System::Drawing::Size(35, 13);
                    this->label8->TabIndex = 1;
                    this->label8->Text = L"Count";
                    //
                    // txtCountStangeShapeEruptionCount
                    //
                    this->txtCountStangeShapeEruptionCount->Location                 =
System::Drawing::Point(47, 48);
                    this->txtCountStangeShapeEruptionCount->Name                     =
L"txtCountStangeShapeEruptionCount";
                    this->txtCountStangeShapeEruptionCount->ReadOnly = true;
                    this->txtCountStangeShapeEruptionCount->Size                     =
System::Drawing::Size(61, 20);
                    this->txtCountStangeShapeEruptionCount->TabIndex = 0;
                    this->txtCountStangeShapeEruptionCount->TabStop = false;
                    //
                    // CountStrangeShapeEruptionLight
                    //
                    this->CountStrangeShapeEruptionLight->BackColor                  =
System::Drawing::Color::Red;
                    this->CountStrangeShapeEruptionLight->Location                   =
System::Drawing::Point(160, 0);
                    this->CountStrangeShapeEruptionLight->Name                       =
L"CountStrangeShapeEruptionLight";
                    this->CountStrangeShapeEruptionLight->Size                       =
System::Drawing::Size(15, 15);
                    this->CountStrangeShapeEruptionLight->TabIndex = 14;
                    this->CountStrangeShapeEruptionLight->TabStop = false;
                    //
```

```
            // groupBox3
            //
            this->groupBox3->Controls->Add(this->btnCalibrate);
            this->groupBox3->Controls->Add(this->CalibrationLight);
            this->groupBox3->Controls->Add(this->txtRealLength);
            this->groupBox3->Controls->Add(this->label6);
            this->groupBox3->Location = System::Drawing::Point(6, 19);
            this->groupBox3->Name = L"groupBox3";
            this->groupBox3->Size = System::Drawing::Size(181, 82);
            this->groupBox3->TabIndex = 1;
            this->groupBox3->TabStop = false;
            this->groupBox3->Text = L"Calibration";
            //
            // btnCalibrate
            //
            this->btnCalibrate->Location = System::Drawing::Point(87, 45);
            this->btnCalibrate->Name = L"btnCalibrate";
            this->btnCalibrate->Size = System::Drawing::Size(75, 23);
            this->btnCalibrate->TabIndex = 17;
            this->btnCalibrate->Text = L"Calibrate";
            this->btnCalibrate->UseVisualStyleBackColor = true;
            this->btnCalibrate->Click  +=  gcnew  System::EventHandler(this,
&Form1::btnCalibrate_Click);
            //
            // CalibrationLight
            //
            this->CalibrationLight->BackColor = System::Drawing::Color::Red;
            this->CalibrationLight->Location  =  System::Drawing::Point(160,
0);
            this->CalibrationLight->Name = L"CalibrationLight";
            this->CalibrationLight->Size = System::Drawing::Size(15, 15);
            this->CalibrationLight->TabIndex = 14;
            this->CalibrationLight->TabStop = false;
            //
            // txtRealLength
            //
            this->txtRealLength->Location = System::Drawing::Point(89, 19);
            this->txtRealLength->Name = L"txtRealLength";
            this->txtRealLength->Size = System::Drawing::Size(66, 20);
            this->txtRealLength->TabIndex = 16;
            //
            // label6
            //
            this->label6->AutoSize = true;
            this->label6->Location = System::Drawing::Point(2, 20);
            this->label6->Name = L"label6";
            this->label6->Size = System::Drawing::Size(81, 13);
            this->label6->TabIndex = 2;
            this->label6->Text = L"Scale length (u)";
            //
            // groupBox8
            //
            this->groupBox8->Controls->Add(this->groupBox6);
            this->groupBox8->Controls->Add(this->label11);
            this->groupBox8->Controls->Add(this->txtImage2Zoom);
            this->groupBox8->Controls->Add(this->txtImage2Width);
            this->groupBox8->Controls->Add(this->label9);
            this->groupBox8->Controls->Add(this->label10);
            this->groupBox8->Controls->Add(this->txtImage2Height);
            this->groupBox8->Location = System::Drawing::Point(96, 3);
            this->groupBox8->Name = L"groupBox8";
            this->groupBox8->Size = System::Drawing::Size(102, 280);
```

```
this->groupBox8->TabIndex = 3;
this->groupBox8->TabStop = false;
this->groupBox8->Text = L"Image2";
//
// groupBox6
//
this->groupBox6->Controls->Add(this->label12);
this->groupBox6->Controls->Add(this->label13);
this->groupBox6->Controls->Add(this->txtXClickImage2);
this->groupBox6->Controls->Add(this->txtYClickImage2);
this->groupBox6->Location = System::Drawing::Point(6, 150);
this->groupBox6->Name = L"groupBox6";
this->groupBox6->Size = System::Drawing::Size(72, 109);
this->groupBox6->TabIndex = 4;
this->groupBox6->TabStop = false;
this->groupBox6->Text = L"Click";
//
// label12
//
this->label12->AutoSize = true;
this->label12->Location = System::Drawing::Point(4, 59);
this->label12->Name = L"label12";
this->label12->Size = System::Drawing::Size(39, 13);
this->label12->TabIndex = 3;
this->label12->Text = L"X Pixel";
//
// label13
//
this->label13->AutoSize = true;
this->label13->Location = System::Drawing::Point(4, 20);
this->label13->Name = L"label13";
this->label13->Size = System::Drawing::Size(39, 13);
this->label13->TabIndex = 2;
this->label13->Text = L"Y Pixel";
//
// txtXClickImage2
//
this->txtXClickImage2->Location = System::Drawing::Point(7, 75);
this->txtXClickImage2->Name = L"txtXClickImage2";
this->txtXClickImage2->ReadOnly = true;
this->txtXClickImage2->Size = System::Drawing::Size(55, 20);
this->txtXClickImage2->TabIndex = 0;
this->txtXClickImage2->TabStop = false;
//
// txtYClickImage2
//
this->txtYClickImage2->Location = System::Drawing::Point(7, 36);
this->txtYClickImage2->Name = L"txtYClickImage2";
this->txtYClickImage2->ReadOnly = true;
this->txtYClickImage2->Size = System::Drawing::Size(55, 20);
this->txtYClickImage2->TabIndex = 1;
this->txtYClickImage2->TabStop = false;
//
// label11
//
this->label11->AutoSize = true;
this->label11->Location = System::Drawing::Point(6, 99);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(34, 13);
this->label11->TabIndex = 3;
this->label11->Text = L"Zoom";
//
```

```
// txtImage2Zoom
//
this->txtImage2Zoom->Location = System::Drawing::Point(6, 115);
this->txtImage2Zoom->Name = L"txtImage2Zoom";
this->txtImage2Zoom->ReadOnly = true;
this->txtImage2Zoom->Size = System::Drawing::Size(62, 20);
this->txtImage2Zoom->TabIndex = 2;
this->txtImage2Zoom->TabStop = false;
//
// txtImage2Width
//
this->txtImage2Width->Location = System::Drawing::Point(6, 35);
this->txtImage2Width->Name = L"txtImage2Width";
this->txtImage2Width->ReadOnly = true;
this->txtImage2Width->Size = System::Drawing::Size(65, 20);
this->txtImage2Width->TabIndex = 1;
this->txtImage2Width->TabStop = false;
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(6, 19);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(60, 13);
this->label9->TabIndex = 0;
this->label9->Text = L"Width Pixel";
//
// label10
//
this->label10->AutoSize = true;
this->label10->Location = System::Drawing::Point(6, 58);
this->label10->Name = L"label10";
this->label10->Size = System::Drawing::Size(63, 13);
this->label10->TabIndex = 1;
this->label10->Text = L"Height Pixel";
//
// txtImage2Height
//
this->txtImage2Height->Location = System::Drawing::Point(6, 76);
this->txtImage2Height->Name = L"txtImage2Height";
this->txtImage2Height->ReadOnly = true;
this->txtImage2Height->Size = System::Drawing::Size(62, 20);
this->txtImage2Height->TabIndex = 1;
this->txtImage2Height->TabStop = false;
//
// groupBox2
//
this->groupBox2->Controls->Add(this->groupBox1);
this->groupBox2->Controls->Add(this->label3);
this->groupBox2->Controls->Add(this->txtZoom);
this->groupBox2->Controls->Add(this->label2);
this->groupBox2->Controls->Add(this->label1);
this->groupBox2->Controls->Add(this->txtHeight);
this->groupBox2->Controls->Add(this->txtWidth);
this->groupBox2->Location = System::Drawing::Point(3, 3);
this->groupBox2->Name = L"groupBox2";
this->groupBox2->Size = System::Drawing::Size(89, 280);
this->groupBox2->TabIndex = 2;
this->groupBox2->TabStop = false;
this->groupBox2->Text = L"Image1";
//
// groupBox1
```

```
//
this->groupBox1->Controls->Add(this->label5);
this->groupBox1->Controls->Add(this->label4);
this->groupBox1->Controls->Add(this->txtXClick);
this->groupBox1->Controls->Add(this->txtYClick);
this->groupBox1->Location = System::Drawing::Point(7, 150);
this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(72, 109);
this->groupBox1->TabIndex = 6;
this->groupBox1->TabStop = false;
this->groupBox1->Text = L"Click";
//
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(4, 59);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(39, 13);
this->label5->TabIndex = 3;
this->label5->Text = L"X Pixel";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(4, 20);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(39, 13);
this->label4->TabIndex = 2;
this->label4->Text = L"Y Pixel";
//
// txtXClick
//
this->txtXClick->Location = System::Drawing::Point(7, 75);
this->txtXClick->Name = L"txtXClick";
this->txtXClick->ReadOnly = true;
this->txtXClick->Size = System::Drawing::Size(55, 20);
this->txtXClick->TabIndex = 0;
this->txtXClick->TabStop = false;
//
// txtYClick
//
this->txtYClick->Location = System::Drawing::Point(7, 36);
this->txtYClick->Name = L"txtYClick";
this->txtYClick->ReadOnly = true;
this->txtYClick->Size = System::Drawing::Size(55, 20);
this->txtYClick->TabIndex = 1;
this->txtYClick->TabStop = false;
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(4, 99);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(34, 13);
this->label3->TabIndex = 3;
this->label3->Text = L"Zoom";
//
// txtZoom
//
this->txtZoom->Location = System::Drawing::Point(7, 117);
this->txtZoom->Name = L"txtZoom";
this->txtZoom->ReadOnly = true;
```

```
			this->txtZoom->Size = System::Drawing::Size(72, 20);
			this->txtZoom->TabIndex = 2;
			this->txtZoom->TabStop = false;
			//
			// label2
			//
			this->label2->AutoSize = true;
			this->label2->Location = System::Drawing::Point(4, 58);
			this->label2->Name = L"label2";
			this->label2->Size = System::Drawing::Size(63, 13);
			this->label2->TabIndex = 1;
			this->label2->Text = L"Height Pixel";
			//
			// label1
			//
			this->label1->AutoSize = true;
			this->label1->Location = System::Drawing::Point(4, 19);
			this->label1->Name = L"label1";
			this->label1->Size = System::Drawing::Size(60, 13);
			this->label1->TabIndex = 5;
			this->label1->Text = L"Width Pixel";
			//
			// txtHeight
			//
			this->txtHeight->Location = System::Drawing::Point(7, 76);
			this->txtHeight->Name = L"txtHeight";
			this->txtHeight->ReadOnly = true;
			this->txtHeight->Size = System::Drawing::Size(72, 20);
			this->txtHeight->TabIndex = 0;
			this->txtHeight->TabStop = false;
			//
			// txtWidth
			//
			this->txtWidth->Location = System::Drawing::Point(7, 35);
			this->txtWidth->Name = L"txtWidth";
			this->txtWidth->ReadOnly = true;
			this->txtWidth->Size = System::Drawing::Size(72, 20);
			this->txtWidth->TabIndex = 4;
			this->txtWidth->TabStop = false;
			//
			// splitContainer2
			//
			this->splitContainer2->Dock                              =
System::Windows::Forms::DockStyle::Fill;
			this->splitContainer2->Location = System::Drawing::Point(0, 0);
			this->splitContainer2->Name = L"splitContainer2";
			//
			// splitContainer2.Panel1
			//
			this->splitContainer2->Panel1->AutoScroll = true;
			this->splitContainer2->Panel1->BackColor                 =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(
192)),

	static_cast<System::Int32>(static_cast<System::Byte>(255)),
static_cast<System::Int32>(static_cast<System::Byte>(255)));
			this->splitContainer2->Panel1->Controls->Add(this->ImageBox2);
			//
			// splitContainer2.Panel2
			//
			this->splitContainer2->Panel2->AutoScroll = true;
```

```
                    this->splitContainer2->Panel2->BackColor                        =
System::Drawing::Color::DeepSkyBlue;
                    this->splitContainer2->Panel2->Controls->Add(this->ImageBox);
                    this->splitContainer2->Size = System::Drawing::Size(106, 970);
                    this->splitContainer2->SplitterDistance = 35;
                    this->splitContainer2->TabIndex = 0;
                    //
                    // ImageBox2
                    //
                    this->ImageBox2->Cursor = System::Windows::Forms::Cursors::Cross;
                    this->ImageBox2->Location = System::Drawing::Point(0, 0);
                    this->ImageBox2->Name = L"ImageBox2";
                    this->ImageBox2->Size = System::Drawing::Size(427, 465);
                    this->ImageBox2->SizeMode                                        =
System::Windows::Forms::PictureBoxSizeMode::StretchImage;
                    this->ImageBox2->TabIndex = 2;
                    this->ImageBox2->TabStop = false;
                    this->ImageBox2->Paint                  +=                  gcnew
System::Windows::Forms::PaintEventHandler(this, &Form1::ImageBox2_Paint);
                    this->ImageBox2->MouseClick                +=               gcnew
System::Windows::Forms::MouseEventHandler(this, &Form1::ImageBox2_MouseClick);
                    //
                    // Form1
                    //
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode                                              =
System::Windows::Forms::AutoScaleMode::Font;
                    this->AutoScroll = true;
                    this->ClientSize = System::Drawing::Size(797, 994);
                    this->Controls->Add(this->splitContainer1);
                    this->Controls->Add(this->menuStrip1);
                    this->MainMenuStrip = this->menuStrip1;
                    this->Name = L"Form1";
                    this->Text = L"3DView";
                    this->WindowState                                                =
System::Windows::Forms::FormWindowState::Maximized;
                    this->Load       +=       gcnew       System::EventHandler(this,
&Form1::Form1_Load);
                    this->menuStrip1->ResumeLayout(false);
                    this->menuStrip1->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->ImageBox))->EndInit();
                    this->splitContainer1->Panel1->ResumeLayout(false);
                    this->splitContainer1->Panel2->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer1))->EndInit();
                    this->splitContainer1->ResumeLayout(false);
                    this->groupBox15->ResumeLayout(false);
                    this->groupBox15->PerformLayout();
                    this->groupBox10->ResumeLayout(false);
                    this->groupBox17->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->setPoint2Light))->EndInit();
                    this->groupBox11->ResumeLayout(false);
                    this->groupBox11->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->MeasureWhiskerLightImage2))->EndInit();
                    this->groupBox4->ResumeLayout(false);
                    this->groupBox4->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->AreaMeasureLight))->EndInit();
                    this->groupBox12->ResumeLayout(false);
```

```
                    this->groupBox12->PerformLayout();
                    this->groupBox14->ResumeLayout(false);
                    this->groupBox14->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CalculateLight))->EndInit();
                    this->groupBox13->ResumeLayout(false);
                    this->groupBox13->PerformLayout();
                    this->groupBox9->ResumeLayout(false);
                    this->groupBox18->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->WhiskerSiameterLight))->EndInit();
                    this->groupBox16->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->setPoint1Light))->EndInit();
                    this->groupBox5->ResumeLayout(false);
                    this->groupBox5->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->MeasureWhiskerLight))->EndInit();
                    this->groupBox19->ResumeLayout(false);
                    this->groupBox19->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CountWhiskerLight))->EndInit();
                    this->groupBox7->ResumeLayout(false);
                    this->groupBox7->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CountStrangeShapeEruptionLight))->EndInit();
                    this->groupBox3->ResumeLayout(false);
                    this->groupBox3->PerformLayout();
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->CalibrationLight))->EndInit();
                    this->groupBox8->ResumeLayout(false);
                    this->groupBox8->PerformLayout();
                    this->groupBox6->ResumeLayout(false);
                    this->groupBox6->PerformLayout();
                    this->groupBox2->ResumeLayout(false);
                    this->groupBox2->PerformLayout();
                    this->groupBox1->ResumeLayout(false);
                    this->groupBox1->PerformLayout();
                    this->splitContainer2->Panel1->ResumeLayout(false);
                    this->splitContainer2->Panel2->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer2))->EndInit();
                    this->splitContainer2->ResumeLayout(false);
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->ImageBox2))->EndInit();
                    this->ResumeLayout(false);
                    this->PerformLayout();

             }
   #pragma endregion
      private:  System::Void  openToolStripMenuItem_Click(System::Object^    sender,
System::EventArgs^  e)
                     {
                            CalibrationComplete = false;
                            this->CalibrationLight->BackColor                   =
System::Drawing::Color::Red;
                            OpenFileDialog ^ dialog = gcnew OpenFileDialog;
                            dialog->Filter               =                 "Image
Files|*.png;*.jpg;*.gif;*.jpeg;*.tif;";
                                   if(dialog->ShowDialog()                     ==
Windows::Forms::DialogResult::OK)
                                   {
```

```
                                           ImageBox->Image   =   Image::FromFile(dialog-
>FileName);
                                           txtImage1Name->Text    =    dialog->FileName-
>ToString();
                                           ImageZoom = 1;
                                           txtZoom->Text = ImageZoom.ToString();
                                           ImageWidth = ImageBox->Image->Width;
                                           ImageHeight = ImageBox->Image->Height;
                                           txtWidth->Text = ImageWidth.ToString();
                                           txtHeight->Text = ImageHeight.ToString();
                                           this->ImageBox->Size                        =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                           ImageLoaded = true;
                                    }

                      }

   private:    System::Void    x1ToolStripMenuItem_Click(System::Object^       sender,
System::EventArgs^  e) //original zoom
               {
                      if(ImageLoaded == true)
                            {
                            ImageWidth = ImageBox->Image->Width;
                            ImageHeight = ImageBox->Image->Height;
                            ImageZoom = 1;
                            txtWidth->Text = (ImageWidth*ImageZoom).ToString();
                            txtHeight->Text = (ImageHeight*ImageZoom).ToString();

                            txtZoom->Text = ImageZoom.ToString();

                            this->ImageBox->Size                                    =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                      }
                      else
                      {MessageBox::Show(L"Please Load Image");}
               }
   private:    System::Void    x01ToolStripMenuItem_Click(System::Object^      sender,
System::EventArgs^  e) //zoom out
               {
                      if(ImageLoaded == true)
                      {
                                   ImageWidth = ImageBox->Image->Width;
                                   ImageHeight = ImageBox->Image->Height;
                                   if(ImageZoom > 1)
                                   {
                                           ImageZoom += -1;
                                    }
                                   else
                                   {
                                           ImageZoom += -0.1;
                                    }
                                   txtWidth->Text = (ImageWidth*ImageZoom).ToString();
                                   txtHeight->Text                                 =
(ImageHeight*ImageZoom).ToString();
                                   txtZoom->Text = ImageZoom.ToString();
                                   this->ImageBox->Size                            =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                      }
                      else
                      {MessageBox::Show(L"Please Load Image");}
               }
```

```
    private:    System::Void    x10ToolStripMenuItem_Click(System::Object^       sender,
System::EventArgs^  e) //zoom in
               {

                      if(ImageLoaded == true)
                      {
                                    ImageWidth = ImageBox->Image->Width;
                                    ImageHeight = ImageBox->Image->Height;
                                     if(ImageZoom >= 14)
                                    {MessageBox::Show(L"Zoom   Has   Exceeded   Its   Max
Value");}
                                    else if(ImageZoom >= 1)
                                    {
                                            ImageZoom += 1;
                                     }
                                    else
                                    {
                                            ImageZoom +=0.1;
                                    }
                                    txtWidth->Text = (ImageWidth*ImageZoom).ToString();
                                    txtHeight->Text                                    =
(ImageHeight*ImageZoom).ToString();
                                    txtZoom->Text = ImageZoom.ToString();
                                    this->ImageBox->Size                               =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                      }
                      else
                      {MessageBox::Show(L"Please Load Image");}
               }

    private:    System::Void    exitToolStripMenuItem_Click(System::Object^      sender,
System::EventArgs^  e)
               {
                      this->Close();
               }

    System::Void ImageBox_Click(System::Object^  sender, System::EventArgs^  e)
               {
                      if(ImageLoaded == true)
                      {
                                    POINT ClickCoordinates = {0};
                                    ::GetCursorPos(&ClickCoordinates);
                                            //click screen coordinate
                                    Point ImageBoxForm = ImageBox->Location;
       //location of image in form
                                    Control^  control  =  dynamic_cast<Control^>(sender);
       //point on form to point on screen tranform
                                    Point   ImageBoxScreen   =   control->PointToScreen(
ImageBoxForm );                     //location   of   image   in   screen   Y-vertical   X-
horozontal
                                    double     ImageYCoo     =     ((ClickCoordinates.y     -
ImageBoxScreen.Y + ImageBoxForm.Y ) / ImageZoom);
                                    double     ImageXCoo     =     ((ClickCoordinates.x     -
ImageBoxScreen.X + ImageBoxForm.X ) / ImageZoom);
                                    //ImageYCoo = ((int)(ImageYCoo*10 + 0.5))/10;
                                    //ImageXCoo = ((int)(ImageXCoo*10 + 0.5))/10;
                                    txtYClick->Text      =      ((int)(ImageYCoo      +
0.5)).ToString();
                                    txtXClick->Text      =      ((int)(ImageXCoo      +
0.5)).ToString();
```

```
                            if(Cailbration == true)
                            {
                                    if(CalibratePointNumber == 1)
                                    {
                                                    CalibrationArray[0]          =
ImageXCoo;
                                                    CalibrationArray[1]          =
ImageYCoo;
                                                    CalibratePointNumber = 2;
                                    }
                                    else if(CalibratePointNumber == 2)
                                    {
                                                    CalibrationArray[2]          =
ImageXCoo;
                                                    CalibrationArray[3]          =
ImageYCoo;
                                                    Cailbration = false;
                                                    CalibratePointNumber = 1;

                                                    double       CaiLength       =
sqrt((double)pow((double)CalibrationArray[2]                                     -
(double)CalibrationArray[0],2)+pow((double)CalibrationArray[3]                   -
(double)CalibrationArray[1],2));
                                                    double RealLength;
                                                    try
                                                    {
                                                            RealLength           =
((double::Parse(txtRealLength->Text))*0.000001);
                                                            LengthPerPixel       =
(RealLength / CaiLength);
                                                            if(CaiLength > 0)
                                                            {}

                                                            else
                                                            {
                                                                    LengthPerPixel  =
(-LengthPerPixel);
                                                            }
                                                            double
RoundedLengthPerPixel = (int)(LengthPerPixel*10000000000 + 0.5);
                                                            RoundedLengthPerPixel  =
RoundedLengthPerPixel/10000000000;
                                                            this->CalibrationLight-
>BackColor = System::Drawing::Color::LawnGreen;
                                                            CalibrationComplete    =
true;
                                                            this->ImageBox->Size   =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                                            this->ImageBox->Size   =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                                            MessageBox::Show("Length
Per Picel = " + (RoundedLengthPerPixel).ToString(), "Calibration Complete");
                                                    }
                                                    catch(FormatException ^)
                                                    {
                                                            this->CalibrationLight-
>BackColor = System::Drawing::Color::Red;
                                                            MessageBox::Show("Please
Enter Scale's Length and then restart Calibration", "Calibration warning");
                                                    }
```

```
                                        }
                                }

                                else if(CountWhisker == true)
                                {
                                        WhiskerCountArray[k][1]= ImageXCoo;
                                        WhiskerCountArray[k][2]= ImageYCoo;
                                        txtCountWhiskerCount->Text = k.ToString();
                                        k +=1;
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                }

                                else if(DiameterMeasure == true)
                                {
                                        if(DiameterMeasurePointNumber==1)
                                        {

                                        DiameterMeasureArray[w3][1][1] = ImageXCoo;
                                        DiameterMeasureArray[w3][1][2] = ImageYCoo;
                                        DiameterMeasurePointNumber=2;
                                        }
                                        else if(DiameterMeasurePointNumber==2)
                                        {

                                        DiameterMeasureArray[w3][2][1] = ImageXCoo;
                                        DiameterMeasureArray[w3][2][2] = ImageYCoo;
                                        DiameterMeasurePointNumber=1;
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);

                                        }
                                }



                                else if (SetRefPoint1 == true)
                                {
                                        RefPoint1Array[1]=ImageXCoo;
                                        RefPoint1Array[2]=ImageYCoo;
                                        SetRefPoint1Complete = true;
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                        this->ImageBox->Size                    =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                        this->setPoint1Light-
>BackColor=System::Drawing::Color::LawnGreen;
                                        MessageBox::Show("Reference    Point    Set",
"Ref");

                                        SetRefPoint1= false;
                                }


                                else if (MeasureWhiskerImage1 == true)
                                        {
                                                WhiskerArrayImage1[w][a][1]         =
ImageXCoo;
```

```
                                                  WhiskerArrayImage1[w][a][2]          =
ImageYCoo;
                                                  a += 1;
                                                  this->ImageBox->Size                 =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                                  this->ImageBox->Size                 =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                          }

                                  else if (CountStangeShapeEruption == true)
                                  {
                                          StangeShapeEruptionArray[j][1] = ImageXCoo;
                                          StangeShapeEruptionArray[j][2] = ImageYCoo;
                                          txtCountStangeShapeEruptionCount->Text       =
j.ToString();
                                          j += 1;


                                          this->ImageBox->Size                         =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                          this->ImageBox->Size                         =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                  }




                                  else if (AreaMeasure == true && CalibrationComplete
== true)
                                  {
                                          if (AreaMeasureNumber == 1)
                                          {
                                                  AreaMeasureArray[0] = ImageXCoo;
                                                  AreaMeasureArray[1] = ImageYCoo;
                                                  AreaMeasureNumber = 2;
                                          }
                                          else if(AreaMeasureNumber == 2)
                                          {
                                                          AreaMeasureArray[2]          =
ImageXCoo;
                                                          AreaMeasureArray[3]          =
ImageYCoo;
                                                          AreaMeasure = false;
                                                          AreaMeasureNumber = 1;

                                                          dx    =   (AreaMeasureArray[0]   -
AreaMeasureArray[2]);
                                                          if(dx > 0)
                                                          {}

                                                          else
                                                          {
                                                                  dx = (-dx);
                                                          }
                                                          dy    =   (AreaMeasureArray[1]   -
AreaMeasureArray[3]);
                                                          if(dy > 0)
                                                          {}

                                                          else
                                                          {
```

```
                                                dy = (-dy);
                                        }

                                        int Area = dx*dy;
                                        double          RealArea          =
Area*LengthPerPixel*LengthPerPixel;
                                        double      RoundedRealArea       =
(int)(RealArea*100000000000000 + 0.5);
                                        RoundedRealArea                   =
RoundedRealArea/100000000000000;
                                        txtArea->Text                     =
RoundedRealArea.ToString();
                                        this->AreaMeasureLight-
>BackColor = System::Drawing::Color::LawnGreen;
                                        AreaComplete = true;
                                        this->ImageBox->Size              =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                        this->ImageBox->Size              =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                                        MessageBox::Show("Area  (m^2)  =
" + (RoundedRealArea).ToString(), "Area");


                                }
                        }
                }
                else{}
        }
   private:     System::Void      ImageBox_Paint(System::Object^          sender,
System::Windows::Forms::PaintEventArgs^  e)
         {
                // Create pen.
                  Pen^ redPen = gcnew Pen( Color::Red,2.0 * ImageZoom );
                  Pen^   greenPen    =   gcnew    Pen(   Color::LawnGreen,2.0    *
ImageZoom );
                  Pen^   bluePen    =   gcnew    Pen(   Color::DodgerBlue,2.0    *
ImageZoom );
                  Pen^ orangePen = gcnew Pen( Color::Orange,2.0 * ImageZoom
);

                    // Create location and size of rectangle.
                       int x = AreaMeasureArray[0] * ImageZoom ;
                       int y = AreaMeasureArray[1] * ImageZoom ;
                       int width = dx * ImageZoom ;
                       int height = dy * ImageZoom ;
        // Draw rectangle to screen.
                       e->Graphics->DrawRectangle(  redPen,  x,  y,  width,
height );

                       int x1 = CalibrationArray[0]* ImageZoom ;
                       int y1 = CalibrationArray[1]* ImageZoom ;
                       int x2 = CalibrationArray[2]* ImageZoom ;
                       int y2 = CalibrationArray[3]* ImageZoom ;
                       e->Graphics->DrawLine( greenPen, x1,y1,x2,y2);

                       int Wid =1;
                       while (Wid <650)
                       {
                              if(DiameterMeasureArray[Wid][1][1] !=0)
                              {
                                     e->Graphics->DrawLine( greenPen,
```

```
(int)(DiameterMeasureArray[Wid][1][1]* ImageZoom),

(int)(DiameterMeasureArray[Wid][1][2]* ImageZoom),

(int)(DiameterMeasureArray[Wid][2][1]* ImageZoom),

(int)(DiameterMeasureArray[Wid][2][2]* ImageZoom));
                                        }
                                        Wid+=1;
                                }



                                int W=1;
                                while (W <650)
                                        {
                                                if((int)(WhiskerArrayImage1[W][1][1] *
ImageZoom) != 0 )

                                                {
                                                        String^       SampleText       =
W.ToString();

                                                        System::Drawing::Font^ drawFont
= gcnew System::Drawing::Font( "Arial",16*ImageZoom );
                                                        SolidBrush^  drawBrush  =  gcnew
SolidBrush( Color::Aqua );

                                                        e->Graphics->DrawString(
SampleText,  drawFont,  drawBrush  ,(int)(WhiskerArrayImage1[W][1][1]  *  ImageZoom  ),
(int)(WhiskerArrayImage1[W][1][2] * ImageZoom ));
                                                }
                                                int b = 1;
                                                while (b < 30)
                                                {

        if((int)(WhiskerArrayImage1[W][b+1][1] * ImageZoom) != 0 )
                                                        {
                                                                e->Graphics->DrawLine(
bluePen,

        (int)(WhiskerArrayImage1[W][b][1] * ImageZoom ),

        (int)(WhiskerArrayImage1[W][b][2] * ImageZoom ),

        (int)(WhiskerArrayImage1[W][b+1][1] * ImageZoom ),

        (int)(WhiskerArrayImage1[W][b+1][2] * ImageZoom ));
                                                        }
                                                b = b + 1;
                                                }
                                        W = W+1;
                                        }


                                int J = 1;
                                while (J < 399 )
                                {
                                        if((int)(StangeShapeEruptionArray[J][1]) != 0
)

                                        {
                                                e->Graphics->DrawArc(orangePen,

        (int)((StangeShapeEruptionArray[J][1] * ImageZoom ) - ((6 * ImageZoom )/2)),
```

```
    (int)((StangeShapeEruptionArray[J][2] * ImageZoom ) - ((6 * ImageZoom )/2)),
                                                6 * ImageZoom,
                                                6 * ImageZoom,
                                                0,
                                                360);
                                        }
                                                J +=1;

                                }




                                int J5 = 1;
                                while (J5 < 399 )
                                {
                                        if((int)(WhiskerCountArray[J5][1]) != 0 )
                                        {
                                                e->Graphics->DrawArc(greenPen,
                                                        (int)((WhiskerCountArray[J5][1]
* ImageZoom ) - ((6 * ImageZoom )/2)),
                                                        (int)((WhiskerCountArray[J5][2]
* ImageZoom ) - ((6 * ImageZoom )/2)),
                                                        6 * ImageZoom,
                                                        6 * ImageZoom,
                                                        0,
                                                        360);
                                        }
                                                J5 +=1;

                                }


                                if((int)(RefPoint1Array[1]) != 0 )
                                        {
                                                e->Graphics->DrawArc(redPen,
                                                        (int)((RefPoint1Array[1]        *
ImageZoom ) - ((6 * ImageZoom )/2)),
                                                        (int)((RefPoint1Array[2]        *
ImageZoom ) - ((6 * ImageZoom )/2)),
                                                        6 * ImageZoom,
                                                        6 * ImageZoom,
                                                        0,
                                                        360);
                                        }


                }



   private:      System::Void      btnCalibrate_Click(System::Object^           sender,
System::EventArgs^  e)
                {
                        if(ImageLoaded == true)
                        {
                                Cailbration = !Cailbration;
                                this->CalibrationLight->BackColor                        =
System::Drawing::Color::Orange;
```

```
                                MessageBox::Show("Input Scale Length then click either
end of the scale line", "Calibration Instructions");
                        }
                        else
                        {}
                }

    private: System::Void btnMeasureAreaMeasure_Click(System::Object^ sender,
System::EventArgs^ e)
                {

                        if(ImageLoaded == true)
                        {
                                if (MeasureWhiskerImage1 = true)
                                {
                                MeasureWhiskerImage1 = false ;
                                this->MeasureWhiskerLight->BackColor =
System::Drawing::Color::Red;
                                }
                               if (CalibrationComplete == true)
                               {
                                       AreaMeasure = !AreaMeasure;
                                       this->AreaMeasureLight->BackColor =
System::Drawing::Color::Orange;
                                       MessageBox::Show("Click the top left corner and
then bottom right of desired area", "Area Measurement");
                               }
                               else
                               {
                                       MessageBox::Show("Please Calibrate first", "Area
Measurement");
                               }
                        }
                        else
                        {}
                }
    private: System::Void btnCountEruptionStart_Click(System::Object^ sender,
System::EventArgs^ e)
                {
                        if(ImageLoaded == true)
                        {
                        MeasureWhiskerImage1 = false ;
                        this->MeasureWhiskerLight->BackColor =
System::Drawing::Color::Red;
                        CountWhisker=false;
                        this->CountWhiskerLight->BackColor =
System::Drawing::Color::Red;
                        CountStangeShapeEruption = true;
                        this->CountStrangeShapeEruptionLight->BackColor =
System::Drawing::Color::Orange;
                        }
                        else
                        {}

                }
    private: System::Void btnCountErutionStop_Click(System::Object^ sender,
System::EventArgs^ e)
                {
                        CountStangeShapeEruption = false;
                        this->CountStrangeShapeEruptionLight->BackColor =
System::Drawing::Color::Red;
```

```
            }
    private:    System::Void    btnMeasureWhiskerStart_Click(System::Object^       sender,
System::EventArgs^  e)
            {
                    if(ImageLoaded == true)
                    {
                        if (CalibrationComplete == true)
                        {
                            MeasureWhiskerImage1 = !MeasureWhiskerImage1;
                            if(MeasureWhiskerImage1 == true)
                            {
                                this->MeasureWhiskerLight->BackColor        =
System::Drawing::Color::Orange;
                                MessageBox::Show("starting from the centre of
the base click the centre of kinks untill the end", "Whisker Measurement");
                            }
                            else
                            {
                                this->MeasureWhiskerLight->BackColor        =
System::Drawing::Color::LightBlue;
                            }
                        }
                        else
                        {
                            MessageBox::Show("Please  Calibrate  first",   "Area
Measurement");
                        }
                    }
                    else
                    {}
            }
    private:   System::Void   btnMeasureWhiskerRecord_Click(System::Object^      sender,
System::EventArgs^  e)
            {
                    if(ImageLoaded == true)
                    {
                    if(MeasureWhiskerImage1 == true)
                    {
                    txtWhiskerStatsCount->Text = (w).ToString();
                    w += 1;
                    a = 1;
                    }
                    }
                    else
                    {}
           }
    private:    System::Void    btnMeasureWhiskerUndo_Click(System::Object^       sender,
System::EventArgs^  e)
            {
                    if(ImageLoaded == true)
                    {
                    if (MeasureWhiskerImage1 == true)
                    {
                        a=1;
                        int A = 0;
                        while (A < 30)
                        {
                            WhiskerArrayImage1[w][A][1] = 0;
                            WhiskerArrayImage1[w][A][2] = 0;
                            A += 1;
                        }
```

```
                            this->ImageBox->Size                                            =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                            this->ImageBox->Size                                            =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                      }
                      }
                      else
                      {}
               }
   private:    System::Void    btnMeasureWhiskerStop_Click(System::Object^       sender,
System::EventArgs^  e)
               {

                     int A = 1;
                     a=1;
                     while (A < 30)
                     {
                                   WhiskerArrayImage1[w][A][1] = 0;
                                   WhiskerArrayImage1[w][A][2] = 0;
                                   A += 1;
                     }
                     MeasureWhiskerImage1 = false;
                     this->ImageBox->Size  =  System::Drawing::Size(ImageWidth*ImageZoom
+ 1, ImageHeight*ImageZoom + 1);
                     this->ImageBox->Size                                               =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                     this->MeasureWhiskerLight->BackColor                               =
System::Drawing::Color::LawnGreen;
               }
   private:    System::Void    toolStripMenuItem1_Click(System::Object^        sender,
System::EventArgs^  e)
               {

                             OpenFileDialog ^ dialogImage2 = gcnew OpenFileDialog;
                             dialogImage2->Filter                 =                 "Image
Files|*.png;*.jpg;*.gif;*.jpeg;*.tif;";
                                    if(dialogImage2->ShowDialog()                       ==
Windows::Forms::DialogResult::OK)
                                    {
                                           ImageBox2->Image                              =
Image::FromFile(dialogImage2->FileName);
                                           Image2Zoom = 1;
                                           txtImage2Name->Text      =      dialogImage2-
>FileName->ToString();
                                           txtImage2Zoom->Text = Image2Zoom.ToString();
                                           Image2Width = ImageBox2->Image->Width;
                                           Image2Height = ImageBox2->Image->Height;
                                           txtImage2Width->Text                          =
Image2Width.ToString();
                                           txtImage2Height->Text                         =
Image2Height.ToString();
                                           this->ImageBox2->Size                         =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                                           Image2Loaded = true;
                                    }


               }
   private:     System::Void      originalZoomToolStripMenuItem1_Click(System::Object^
sender, System::EventArgs^  e)
               {
                      if(Image2Loaded == true)
```

```
                            {
                            Image2Width = ImageBox2->Image->Width;
                            Image2Height = ImageBox2->Image->Height;
                            Image2Zoom = 1;
                            txtImage2Width->Text                                 =
(Image2Width*Image2Zoom).ToString();
                            txtImage2Height->Text                                =
(Image2Height*Image2Zoom).ToString();
                            txtImage2Zoom->Text = Image2Zoom.ToString();

                            this->ImageBox2->Size                                =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                      }
                      else
                      {MessageBox::Show(L"Please Load Image");}

               }
   private:   System::Void   zoomInToolStripMenuItem1_Click(System::Object^     sender,
System::EventArgs^  e)
               {
                     if(Image2Loaded == true)
                      {
                                   Image2Width = ImageBox2->Image->Width;
                                   Image2Height = ImageBox2->Image->Height;
                                    if(Image2Zoom >= 14)
                                   {MessageBox::Show(L"Zoom   Has   Exceeded   Its   Max
Value");}
                                   else if(Image2Zoom >= 1)
                                   {
                                           Image2Zoom += 1;
                                    }
                                   else
                                   {
                                           Image2Zoom +=0.1;
                                   }
                                   txtImage2Width->Text                          =
(Image2Width*Image2Zoom).ToString();
                                   txtImage2Height->Text                         =
(Image2Height*Image2Zoom).ToString();
                                   txtImage2Zoom->Text = Image2Zoom.ToString();
                                   this->ImageBox2->Size                         =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                      }
                      else
                      {MessageBox::Show(L"Please Load Image");}

               }
   private:   System::Void  zoomOutToolStripMenuItem1_Click(System::Object^     sender,
System::EventArgs^  e)
               {
                      if(Image2Loaded == true)
                      {
                                   Image2Width = ImageBox2->Image->Width;
                                   Image2Height = ImageBox2->Image->Height;
                                   if(Image2Zoom > 1)
                                   {
                                           Image2Zoom += -1;
                                    }
                                   else
                                   {
                                           Image2Zoom += -0.1;
                                    }
```

```
                                   txtImage2Width->Text                              =
(Image2Width*Image2Zoom).ToString();
                                   txtImage2Height->Text                             =
(Image2Height*Image2Zoom).ToString();
                                   txtImage2Zoom->Text = Image2Zoom.ToString();
                                   this->ImageBox2->Size                             =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                      }
                      else
                      {
                             MessageBox::Show(L"Please Load Image");
                      }
               }
   private:     System::Void     ImageBox2_MouseClick(System::Object^          sender,
System::Windows::Forms::MouseEventArgs^  e)
               {
                      if(Image2Loaded == true)
                      {
                                   POINT ClickCoordinatesImage2 = {0};
                                   ::GetCursorPos(&ClickCoordinatesImage2);
                                                  //click screen coordinate
                                   Point ImageBox2Form = ImageBox2->Location;
              //location of image in form
                                   Control^ control2 = dynamic_cast<Control^>(sender);
       //point on form to point on screen tranform
                                   Point   ImageBox2Screen   =   control2->PointToScreen(
ImageBox2Form );                   //location   of   image   in   screen   Y-vertical   X-
horozontal
                                   double  Image2YCoo   =   ((ClickCoordinatesImage2.y   -
ImageBox2Screen.Y + ImageBox2Form.Y ) / ImageZoom);
                                   double  Image2XCoo   =   ((ClickCoordinatesImage2.x   -
ImageBox2Screen.X + ImageBox2Form.X ) / ImageZoom);
                                   //ImageYCoo = ((int)(ImageYCoo*10 + 0.5))/10;
                                   //ImageXCoo = ((int)(ImageXCoo*10 + 0.5))/10;
                                   txtYClickImage2->Text    =    ((int)(Image2YCoo    +
0.5)).ToString();
                                   txtXClickImage2->Text    =    ((int)(Image2XCoo    +
0.5)).ToString();

                                   if (MeasureWhiskerImage2 == true)
                                          {
                                                 WhiskerArrayImage2[w2][a2][1]        =
Image2XCoo;
                                                 WhiskerArrayImage2[w2][a2][2]        =
Image2YCoo;
                                                 a2 += 1;
                                                 this->ImageBox2->Size                =
System::Drawing::Size(Image2Width*Image2Zoom + 1, Image2Height*Image2Zoom + 1);
                                                 this->ImageBox2->Size                =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                                          }
                                   else if (SetRefPoint2 == true)
                                   {
                                          RefPoint2Array[1]=Image2XCoo;
                                          RefPoint2Array[2]=Image2YCoo;

                                          SetRefPoint2Complete = true;
                                          SetRefPoint2=false;
                                          this->ImageBox2->Size                       =
System::Drawing::Size(Image2Width*Image2Zoom + 1, Image2Height*Image2Zoom + 1);
                                          this->ImageBox2->Size                       =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
```

```
                                        this->setPoint2Light->BackColor          =
System::Drawing::Color::LawnGreen;
                                        MessageBox::Show("Reference  Point  two  Set",
"Ref");

                                  }




                      }
                      else{}

                }
   private:  System::Void  button4_Click(System::Object^   sender,  System::EventArgs^
e)
                {
                if(Image2Loaded == true)
                      {
                            MeasureWhiskerImage2 = !MeasureWhiskerImage2;
                            if(MeasureWhiskerImage2 == true)
                                  {
                                        this->MeasureWhiskerLightImage2->BackColor  =
System::Drawing::Color::Orange;
                                        MessageBox::Show("starting from the centre of
the base click the centre of kinks untill the end", "Whisker Measurement");
                                  }
                                  else
                                  {
                                        this->MeasureWhiskerLightImage2->BackColor  =
System::Drawing::Color::LightBlue;
                                  }
                      }
                      else{}
               }
   private:     System::Void      ImageBox2_Paint(System::Object^           sender,
System::Windows::Forms::PaintEventArgs^  e)
                {
                      // Create pens.
                      Pen^ redPen = gcnew Pen( Color::Red,2.0 * Image2Zoom );
                      Pen^ greenPen = gcnew Pen( Color::LawnGreen,2.0 * Image2Zoom  );
                      Pen^ bluePen = gcnew Pen( Color::DodgerBlue,2.0 * Image2Zoom   );
                      Pen^ orangePen = gcnew Pen( Color::Orange,2.0 * Image2Zoom   );



                      int W=1;
                      while (W <650)
                            {
                                  if((int)(WhiskerArrayImage2[W][1][1]  *  Image2Zoom)
!= 0 )
                                              {
                                                    String^      SampleText      =
W.ToString();
                                                    System::Drawing::Font^ drawFont
= gcnew System::Drawing::Font( "Arial",16*Image2Zoom );
                                                    SolidBrush^  drawBrush  =  gcnew
SolidBrush( Color::Aqua );
                                                    e->Graphics->DrawString(
SampleText,  drawFont,  drawBrush  ,(int)(WhiskerArrayImage2[W][1][1]  *  Image2Zoom),
(int)(WhiskerArrayImage2[W][1][2] * Image2Zoom ));
```

```
                                                    }
                                        int b = 1;
                                        while (b < 30)
                                        {
                                               if((int)(WhiskerArrayImage2[W][b+1][1]      *
Image2Zoom) != 0 )
                                               {
                                                      e->Graphics->DrawLine( bluePen,

       (int)(WhiskerArrayImage2[W][b][1] * Image2Zoom  ),

       (int)(WhiskerArrayImage2[W][b][2] * Image2Zoom  ),

       (int)(WhiskerArrayImage2[W][b+1][1] * Image2Zoom ),

       (int)(WhiskerArrayImage2[W][b+1][2] * Image2Zoom ));
                                               }
                                        b = b + 1;
                                        }
                                 W = W+1;
                                 }

                          if((int)(RefPoint2Array[1]) != 0 )
                                               {
                                                      e->Graphics->DrawArc(redPen,
                                                             (int)((RefPoint2Array[1]      *
Image2Zoom ) - ((6 * Image2Zoom )/2)),
                                                             (int)((RefPoint2Array[2]      *
Image2Zoom ) - ((6 * Image2Zoom )/2)),
                                                             6 * Image2Zoom,
                                                             6 * Image2Zoom,
                                                             0,
                                                             360);
                                               }


               }
   private:  System::Void  button1_Click(System::Object^    sender,  System::EventArgs^
e)
               {
                      int A = 1;
                      a2=1;
                      while (A < 30)
                      {
                                    WhiskerArrayImage2[w2][A][1] = 0;
                                    WhiskerArrayImage2[w2][A][2] = 0;
                                    A += 1;
                      }
                      MeasureWhiskerImage2 = false;
                      this->ImageBox2->Size                                              =
System::Drawing::Size(Image2Width*Image2Zoom + 1, Image2Height*Image2Zoom + 1);
                      this->ImageBox2->Size                                              =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                      this->MeasureWhiskerLightImage2->BackColor                         =
System::Drawing::Color::LawnGreen;

               }
   private:  System::Void  button3_Click(System::Object^    sender,  System::EventArgs^
e)
               {
                       if(MeasureWhiskerImage2 == true)
                       {
```

```
                        txtWhiskerStatsCountImage2->Text = (w2).ToString();
                        w2 += 1;
                        a2 = 1;
                        }

                }

   private:   System::Void   MeasureWhiskerUndoImage2_Click(System::Object^     sender,
System::EventArgs^  e)
                {

                        if (MeasureWhiskerImage2 == true)
                        {
                              a2=1;
                              int A = 0;
                              while (A < 30)
                              {
                                     WhiskerArrayImage2[w2][A][1] = 0;
                                     WhiskerArrayImage2[w2][A][2] = 0;
                                     A += 1;
                              }
                              this->ImageBox2->Size                                    =
System::Drawing::Size(Image2Width*Image2Zoom + 1, Image2Height*Image2Zoom + 1);
                              this->ImageBox2->Size                                    =
System::Drawing::Size(Image2Width*Image2Zoom, Image2Height*Image2Zoom);
                        }


                }
   private:     System::Void     btn3dCalculate_Click(System::Object^          sender,
System::EventArgs^  e)
                {
                        if (SetRefPoint1Complete == true && SetRefPoint2Complete ==true)
                        {
                              CalculateComplete = true;

                              //create array for bases at the origins
                               //image1 and image 2


                              int W1=1;
                               while (W1 < 699)
                                     {
                                            int b1=1;
                                            while ( b1 < 29)
                                            {
                                                   if(WhiskerArrayImage1[W1][b1][1] != 0)
                                                   {

       WhiskerArrayImage1BaseOrigin[W1][b1][1]=(WhiskerArrayImage1[W1][b1][1]-
RefPoint1Array[1]);

       WhiskerArrayImage1BaseOrigin[W1][b1][2]=-(WhiskerArrayImage1[W1][b1][2]-
RefPoint1Array[2]);


       WhiskerArrayImage2BaseOrigin[W1][b1][1]=(WhiskerArrayImage2[W1][b1][1]-
RefPoint2Array[1]);

       WhiskerArrayImage2BaseOrigin[W1][b1][2]=-(WhiskerArrayImage2[W1][b1][2]-
RefPoint2Array[2]);
                                                   }
```

```
                                        b1=b1+1;
                                }
                                W1 = W1+1;
                        }

                        int b4=1;
                        while ( b4 < 399)
                        {
                                if(StangeShapeEruptionArray[b4][1] != 0)
                                {

        StangeShapeEruptionArrayOrign[b4][1]=StangeShapeEruptionArray[b4][1]-
RefPoint1Array[1];
                                        StangeShapeEruptionArrayOrign[b4][2]=-
(StangeShapeEruptionArray[b4][2]-RefPoint1Array[2]);
                                }
                                b4=b4+1;
                        }

                        int b7=1;
                        while ( b7 < 399)
                        {
                                if(WhiskerCountArray[b7][1] != 0)
                                {

        WhiskerCountArrayOrign[b7][1]=WhiskerCountArray[b7][1]-RefPoint1Array[1];
                                        WhiskerCountArrayOrign[b7][2]=-
(WhiskerCountArray[b7][2]-RefPoint1Array[2]);
                                }
                                b7=b7+1;
                        }


                //image1 and image2 and image1 imaginary and image2 imaginary
                //Real coordinates relative to coordinates of the surface of the
sample along the x,y surface

                        int WR1=1;
                        int z1=0;

                        try
                        {
                                double          WorkingDistance          =
(double::Parse(txt3DCalculationWorkingDistance->Text))*0.001;
                                double
ImagionaryZ=WorkingDistance/LengthPerPixel;
                                double    CentreOfImagex     =    (ImageBox-
>Width)/(ImageZoom*2);
                                double    CentreOfImagey     =    (ImageBox-
>Height)/(ImageZoom*2);
                                double    CentreOfImagexOrign=CentreOfImagex-
RefPoint1Array[1];
                                double    CentreOfImageyOrign=CentreOfImagey-
RefPoint1Array[2];
                                double CamraPositionX=CentreOfImagexOrign;
                                double CamraPositionY=CentreOfImageyOrign;
                                double CamraPositionZ=ImagionaryZ;

                                double    CentreOfImage2x     =    (ImageBox2-
>Width)/(Image2Zoom*2);
                                double    CentreOfImage2y     =    (ImageBox2-
>Height)/(Image2Zoom*2);
```

```
                                        double  CentreOfImage2xOrign=CentreOfImage2x-
RefPoint2Array[1];
                                        double  CentreOfImage2yOrign=CentreOfImage2y-
RefPoint2Array[2];
                                        double CamraPosition2X=CentreOfImage2xOrign;
                                        double CamraPosition2Y=CentreOfImage2yOrign;
                                        double CamraPosition2Z=ImagionaryZ;



                                        double       Image1Alpha       =        -
(double::Parse(txtImage1Alpha->Text)*(PI/180));
                                        double       Image1Beta        =        -
(double::Parse(txtImage1Beta->Text)*(PI/180));
                                        double       Image2Alpha       =        -
(double::Parse(txtImage2Alpha->Text)*(PI/180));
                                        double       Image2Beta        =        -
(double::Parse(txtImage2Beta->Text)*(PI/180));

                                        while (WR1 <699)
                                        {
                                               int bR1=1;
                                               while (bR1 < 29)
                                               {

       WhiskerArrayImage1RealXYZ[WR1][bR1][1]=(WhiskerArrayImage1BaseOrigin[WR1][bR1][
1]*cos(Image1Alpha)*cos(Image1Beta))+(z1*sin(Image1Alpha)*cos(Image1Beta))-
(WhiskerArrayImage1BaseOrigin[WR1][bR1][2]*sin(Image1Beta));

       WhiskerArrayImage1RealXYZ[WR1][bR1][2]=(WhiskerArrayImage1BaseOrigin[WR1][bR1][
2]*cos(Image1Beta))+(WhiskerArrayImage1BaseOrigin[WR1][bR1][1]*cos(Image1Alpha)*sin(Im
age1Beta))+(z1*sin(Image1Alpha)*sin(Image1Beta));

       WhiskerArrayImage1RealXYZ[WR1][bR1][3]=(z1*cos(Image1Alpha))-
(WhiskerArrayImage1BaseOrigin[WR1][bR1][1]*sin(Image1Alpha));


       WhiskerArrayImage1RealXYZImaginary[WR1][bR1][1]=(CamraPositionX*cos(Image1Alpha
)*cos(Image1Beta))+(CamraPositionZ*sin(Image1Alpha)*cos(Image1Beta))-
(CamraPositionY*sin(Image1Beta));

       WhiskerArrayImage1RealXYZImaginary[WR1][bR1][2]=(CamraPositionY*cos(Image1Beta)
)+(CamraPositionX*cos(Image1Alpha)*sin(Image1Beta))+(CamraPositionZ*sin(Image1Alpha)*s
in(Image1Beta));

       WhiskerArrayImage1RealXYZImaginary[WR1][bR1][3]=(CamraPositionZ*cos(Image1Alpha
))-(CamraPositionX*sin(Image1Alpha));



       WhiskerArrayImage2RealXYZ[WR1][bR1][1]=(WhiskerArrayImage2BaseOrigin[WR1][bR1][
1]*cos(Image2Alpha)*cos(Image2Beta))+(z1*sin(Image2Alpha)*cos(Image2Beta))-
(WhiskerArrayImage2BaseOrigin[WR1][bR1][2]*sin(Image2Beta));

       WhiskerArrayImage2RealXYZ[WR1][bR1][2]=(WhiskerArrayImage2BaseOrigin[WR1][bR1][
2]*cos(Image2Beta))+(WhiskerArrayImage2BaseOrigin[WR1][bR1][1]*cos(Image2Alpha)*sin(Im
age2Beta))+(z1*sin(Image2Alpha)*sin(Image2Beta));

       WhiskerArrayImage2RealXYZ[WR1][bR1][3]=(z1*cos(Image2Alpha))-
(WhiskerArrayImage2BaseOrigin[WR1][bR1][1]*sin(Image2Alpha));
```

```
	WhiskerArrayImage2RealXYZImaginary[WR1][bR1][1]=(CamraPosition2X*cos(Image2Alpha)*cos(Image2Beta))+(CamraPosition2Z*sin(Image2Alpha)*cos(Image2Beta))-(CamraPosition2Y*sin(Image2Beta));

	WhiskerArrayImage2RealXYZImaginary[WR1][bR1][2]=(CamraPosition2Y*cos(Image2Beta))+(CamraPosition2X*cos(Image2Alpha)*sin(Image2Beta))+(CamraPosition2Z*sin(Image2Alpha)*sin(Image2Beta));

	WhiskerArrayImage2RealXYZImaginary[WR1][bR1][3]=(CamraPosition2Z*cos(Image2Alpha))-(CamraPosition2X*sin(Image2Alpha));

							bR1 +=1;

						}
					WR1 +=1;
					}
				}
				catch(FormatException ^)
				{
					MessageBox::Show("Please Enter angles of images", "data Warning");
				}

				try
				{
					double WorkingDistance = (double::Parse(txt3DCalculationWorkingDistance->Text))*0.001;
					double ImagionaryZ=WorkingDistance/LengthPerPixel;
					double CentreOfImagex = (ImageBox->Width)/(ImageZoom*2);
					double CentreOfImagey = (ImageBox->Height)/(ImageZoom*2);
					double CentreOfImagexOrign=CentreOfImagex-RefPoint1Array[1];
					double CentreOfImageyOrign=CentreOfImagey-RefPoint1Array[2];
					double CamraPositionX=CentreOfImagexOrign;
					double CamraPositionY=CentreOfImageyOrign;
					double CamraPositionZ=ImagionaryZ;

					 Image1Alpha = -(double::Parse(txtImage1Alpha->Text)*(PI/180));
					 Image1Beta = -(double::Parse(txtImage1Beta->Text)*(PI/180));
					 Image2Alpha = -(double::Parse(txtImage2Alpha->Text)*(PI/180));
					 Image2Beta = -(double::Parse(txtImage2Beta->Text)*(PI/180));
			int b5=1;
			while (b5 < 399)
			{

	StangeShapeEruptionArrayXYZ[b5][1]=(StangeShapeEruptionArrayOrign[b5][1]*cos(Image1Alpha)*cos(Image1Beta))+(z1*sin(Image1Alpha)*cos(Image1Beta))-(StangeShapeEruptionArrayOrign[b5][2]*sin(Image1Beta));

	StangeShapeEruptionArrayXYZ[b5][2]=(StangeShapeEruptionArrayOrign[b5][2]*cos(Image1Beta))+(StangeShapeEruptionArrayOrign[b5][1]*cos(Image1Alpha)*sin(Image1Beta))+(z1*sin(Image1Alpha)*sin(Image1Beta));
```

```
                            StangeShapeEruptionArrayXYZ[b5][3]=(z1*cos(Image1Alpha))-
(StangeShapeEruptionArrayOrign[b5][1]*sin(Image1Alpha));


       StangeShapeEruptionArrayXYZImaginary[b5][1]=(CamraPositionX*cos(Image1Alpha)*co
s(Image1Beta))+(CamraPositionZ*sin(Image1Alpha)*cos(Image1Beta))-
(CamraPositionY*sin(Image1Beta));

       StangeShapeEruptionArrayXYZImaginary[b5][2]=(CamraPositionY*cos(Image1Beta))+(C
amraPositionX*cos(Image1Alpha)*sin(Image1Beta))+(CamraPositionZ*sin(Image1Alpha)*sin(I
mage1Beta));

       StangeShapeEruptionArrayXYZImaginary[b5][3]=(CamraPositionZ*cos(Image1Alpha))-
(CamraPositionX*sin(Image1Alpha));




       WhiskerCountArrayXYZ[b5][1]=(WhiskerCountArrayOrign[b5][1]*cos(Image1Alpha)*cos
(Image1Beta))+(z1*sin(Image1Alpha)*cos(Image1Beta))-
(WhiskerCountArrayOrign[b5][2]*sin(Image1Beta));

       WhiskerCountArrayXYZ[b5][2]=(WhiskerCountArrayOrign[b5][2]*cos(Image1Beta))+(Wh
iskerCountArrayOrign[b5][1]*cos(Image1Alpha)*sin(Image1Beta))+(z1*sin(Image1Alpha)*sin
(Image1Beta));
                            WhiskerCountArrayXYZ[b5][3]=(z1*cos(Image1Alpha))-
(WhiskerCountArrayOrign[b5][1]*sin(Image1Alpha));


       WhiskerCountArrayXYZImaginary[b5][1]=(CamraPositionX*cos(Image1Alpha)*cos(Image
1Beta))+(CamraPositionZ*sin(Image1Alpha)*cos(Image1Beta))-
(CamraPositionY*sin(Image1Beta));

       WhiskerCountArrayXYZImaginary[b5][2]=(CamraPositionY*cos(Image1Beta))+(CamraPos
itionX*cos(Image1Alpha)*sin(Image1Beta))+(CamraPositionZ*sin(Image1Alpha)*sin(Image1Be
ta));

       WhiskerCountArrayXYZImaginary[b5][3]=(CamraPositionZ*cos(Image1Alpha))-
(CamraPositionX*sin(Image1Alpha));




                            b5 +=1;
                     }
                     }
                     catch(FormatException ^)
                            {
                                   MessageBox::Show("Please  Enter  angles  and  working
distance of images", "data Warning");
                            }


                     int W3=1;
                     while (W3 <699)
                            {
                                   int b3=1;
                                   while (b3 < 29)
                                   {
                                          WhiskerArray3d[W3][b3][1]=Cross(1,
```

```
              WhiskerArrayImage1RealXYZ[W3][b3][1],
WhiskerArrayImage1RealXYZ[W3][b3][2], WhiskerArrayImage1RealXYZ[W3][b3][3],

              WhiskerArrayImage1RealXYZImaginary[W3][b3][1],
WhiskerArrayImage1RealXYZImaginary[W3][b3][2],
WhiskerArrayImage1RealXYZImaginary[W3][b3][3],

              WhiskerArrayImage2RealXYZ[W3][b3][1],
WhiskerArrayImage2RealXYZ[W3][b3][2], WhiskerArrayImage2RealXYZ[W3][b3][3],

              WhiskerArrayImage2RealXYZImaginary[W3][b3][1],
WhiskerArrayImage2RealXYZImaginary[W3][b3][2],
WhiskerArrayImage2RealXYZImaginary[W3][b3][3]);
                                          WhiskerArray3d[W3][b3][2]=Cross(2,

       WhiskerArrayImage1RealXYZ[W3][b3][1],     WhiskerArrayImage1RealXYZ[W3][b3][2],
WhiskerArrayImage1RealXYZ[W3][b3][3],

       WhiskerArrayImage1RealXYZImaginary[W3][b3][1],
WhiskerArrayImage1RealXYZImaginary[W3][b3][2],
WhiskerArrayImage1RealXYZImaginary[W3][b3][3],

       WhiskerArrayImage2RealXYZ[W3][b3][1],     WhiskerArrayImage2RealXYZ[W3][b3][2],
WhiskerArrayImage2RealXYZ[W3][b3][3],

       WhiskerArrayImage2RealXYZImaginary[W3][b3][1],
WhiskerArrayImage2RealXYZImaginary[W3][b3][2],
WhiskerArrayImage2RealXYZImaginary[W3][b3][3]);
                                          WhiskerArray3d[W3][b3][3]=Cross(3,

       WhiskerArrayImage1RealXYZ[W3][b3][1],     WhiskerArrayImage1RealXYZ[W3][b3][2],
WhiskerArrayImage1RealXYZ[W3][b3][3],

       WhiskerArrayImage1RealXYZImaginary[W3][b3][1],
WhiskerArrayImage1RealXYZImaginary[W3][b3][2],
WhiskerArrayImage1RealXYZImaginary[W3][b3][3],

       WhiskerArrayImage2RealXYZ[W3][b3][1],     WhiskerArrayImage2RealXYZ[W3][b3][2],
WhiskerArrayImage2RealXYZ[W3][b3][3],

       WhiskerArrayImage2RealXYZImaginary[W3][b3][1],
WhiskerArrayImage2RealXYZImaginary[W3][b3][2],
WhiskerArrayImage2RealXYZImaginary[W3][b3][3]);


                                          b3+=1;
                                   }
                                   W3+=1;
                            }



                     int b6=1;
                                   while (b6 < 399)
                                   {
                                          StangeShapeEruptionArray3d[b6][1]=eruption(1,

                            StangeShapeEruptionArrayXYZ[b6][1],

                            StangeShapeEruptionArrayXYZ[b6][2],
```

```
        StangeShapeEruptionArrayXYZ[b6][3],

        StangeShapeEruptionArrayXYZImaginary[b6][1],

        StangeShapeEruptionArrayXYZImaginary[b6][2],

        StangeShapeEruptionArrayXYZImaginary[b6][3]);
                        StangeShapeEruptionArray3d[b6][2]=eruption(2,

        StangeShapeEruptionArrayXYZ[b6][1],

        StangeShapeEruptionArrayXYZ[b6][2],

        StangeShapeEruptionArrayXYZ[b6][3],

        StangeShapeEruptionArrayXYZImaginary[b6][1],

        StangeShapeEruptionArrayXYZImaginary[b6][2],

        StangeShapeEruptionArrayXYZImaginary[b6][3]);
                        StangeShapeEruptionArray3d[b6][3]=eruption(3,

        StangeShapeEruptionArrayXYZ[b6][1],

        StangeShapeEruptionArrayXYZ[b6][2],

        StangeShapeEruptionArrayXYZ[b6][3],

        StangeShapeEruptionArrayXYZImaginary[b6][1],

        StangeShapeEruptionArrayXYZImaginary[b6][2],

        StangeShapeEruptionArrayXYZImaginary[b6][3]);


                        WhiskerCountArray3d[b6][1]=eruption(1,

WhiskerCountArrayXYZ[b6][1],

WhiskerCountArrayXYZ[b6][2],

WhiskerCountArrayXYZ[b6][3],

WhiskerCountArrayXYZImaginary[b6][1],

WhiskerCountArrayXYZImaginary[b6][2],

WhiskerCountArrayXYZImaginary[b6][3]);

                        WhiskerCountArray3d[b6][2]=eruption(2,

WhiskerCountArrayXYZ[b6][1],

WhiskerCountArrayXYZ[b6][2],

WhiskerCountArrayXYZ[b6][3],

WhiskerCountArrayXYZImaginary[b6][1],

WhiskerCountArrayXYZImaginary[b6][2],
```

```
                        WhiskerCountArrayXYZImaginary[b6][3]);
                                            WhiskerCountArray3d[b6][3]=eruption(3,
                        WhiskerCountArrayXYZ[b6][1],
                        WhiskerCountArrayXYZ[b6][2],
                        WhiskerCountArrayXYZ[b6][3],
                        WhiskerCountArrayXYZImaginary[b6][1],
                        WhiskerCountArrayXYZImaginary[b6][2],
                        WhiskerCountArrayXYZImaginary[b6][3]);



                                            b6+=1;
                                    }

                                    this->CalculateLight->BackColor                    =
System::Drawing::Color::LawnGreen;
                        }
                        else
                        {
                                MessageBox::Show("Please  Set  Reference  Points  to  set  an
origin", "data Warning");
                        }
                        }
    private:    System::Void    toolStripMenuItem2_Click(System::Object^        sender,
System::EventArgs^  e)
                {
                         if (CalculateComplete == true)
                         {
                        SaveFileDialog^ saveFileDlg = gcnew SaveFileDialog;
                        saveFileDlg->Filter = "txt files|*.txt";
                        saveFileDlg->FilterIndex = 2;
                        saveFileDlg->RestoreDirectory = true;
                        if            (           saveFileDlg->ShowDialog()           ==
Windows::Forms::DialogResult::OK )
                        {
                                System::IO::File::WriteAllText(saveFileDlg->FileName,
String::Empty);

                                try
                                {
                                System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Name: , " + txtSampleInfoName->Text + Environment::NewLine));
                                System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image1 Name: , " + txtImage1Name->Text + Environment::NewLine));
                                System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image2 Name: , " + txtImage2Name->Text + Environment::NewLine));

                                System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Substrate:      ,      "      +      txtSampleInfoSubstrate->Text      +
Environment::NewLine));
```

```
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Coating Material: , " + txtSampleInfoCoating->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Coating Thickness: , " + txtSampleInfoCoatingThickness->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Electrolyte: , " + txtSampleInfoElectrolyte->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Voltage Pattern: , " + txtSampleInfoVoltage->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Current Density Pattern: , " + txtSampleInfoCurrent->Text +
Environment::NewLine));



                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Date of Creation: , " + txtSampleInfoDateCreation->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Time of Creation: , " + txtSampleInfoTimeCreation->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Date of Image: , " + txtSampleInfoDateImage->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Time of Image: , " + txtSampleInfoTimeImage->Text +
Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image1 Tilt: , " + Image1Alpha/(PI/180) + Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image1 Rotation: , " + Image1Beta/(PI/180) + Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image2 tilt: , " + Image2Alpha/(PI/180) + Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Image2 Rotation: , " + Image2Beta/(PI/180) + Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Comments: , " + txtSampleInfoComments->Text + Environment::NewLine +
Environment::NewLine));
                            }
                            catch(FormatException ^)
                            {
                                    MessageBox::Show("Please fill in sample info
boxes");
                            }

                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Number of Whiskers: , " + (w-1) + Environment::NewLine ));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Number of Strange Shape Eruptions: , " + (j-1) + Environment::NewLine
));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Number Whiskers Counted with fast Count: , " + (k-1) +
Environment::NewLine + Environment::NewLine ));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Length Per Pixel: , " + LengthPerPixel + Environment::NewLine +
Environment::NewLine + Environment::NewLine));

                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Strange Shape eruption " + Environment::NewLine));
```

```
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format("x:,"));
                            int bxs =1;
                            while(bxs<399)
                            {
                                   if(StangeShapeEruptionArray3d[bxs][1] != 0 )
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(StangeShapeEruptionArray3d[bxs][1] + ", " ));
                                   }
                                   bxs+=1;
                            }
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format(Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format("y:,"));
                            int bys =1;
                            while(bys<399)
                            {
                                   if(StangeShapeEruptionArray3d[bys][2] != 0)
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(StangeShapeEruptionArray3d[bys][2] + "," ));
                                   }
                                   bys+=1;
                            }
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format(Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format("z:,"));
                            int bzs =1;
                            while(bzs<399)
                            {
                                   if(StangeShapeEruptionArray3d[bzs][1] != 0)
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(StangeShapeEruptionArray3d[bzs][3] + "," ));
                                   }
                                   bzs+=1;
                            }
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format(Environment::NewLine + Environment::NewLine));




                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format("Whisker Count " + Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName, 
String::Format("x:,"));
                            int bxw =1;
                            while(bxw<399)
                            {
                                   if(WhiskerCountArray3d[bxw][1] != 0 )
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(WhiskerCountArray3d[bxw][1] + ", " ));
                                   }
                                   bxw+=1;
                            }
```

```
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("y:,"));
                            int byw =1;
                            while(byw<399)
                            {
                                   if(WhiskerCountArray3d[byw][2] != 0)
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(WhiskerCountArray3d[byw][2] + "," ));
                                   }
                                   byw+=1;
                            }
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(Environment::NewLine));
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("z:,"));
                            int bzw =1;
                            while(bzw<399)
                            {
                                   if(WhiskerCountArray3d[bzw][1] != 0)
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(WhiskerCountArray3d[bzw][3] + "," ));
                                   }
                                   bzw+=1;
                            }
                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(Environment::NewLine + Environment::NewLine));




                            System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format("Whiskers " + Environment::NewLine));
                            int WN=1;
                            while (WN<699)
                            {
                                   if(WhiskerArray3d[WN][2][1] != 0 || WN == 1)
                                   {
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format("Whisker " + WN + Environment::NewLine));
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName,  String::Format("Diameter,  "  +  (sqrt(pow((DiameterMeasureArray[WN][2][1]-
DiameterMeasureArray[WN][1][1]),2)+pow((DiameterMeasureArray[WN][2][2]-
DiameterMeasureArray[WN][1][2]),2))) + Environment::NewLine));
                                          System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format("x:,"));
                                          int bx =1;
                                          while(bx<30)
                                          {
                                                 if(WhiskerArray3d[WN][bx][1]  !=  0  ||
bx < 2)
                                                 {

       System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(WhiskerArray3d[WN][bx][1] + ", " ));
                                                 }
                                                 bx+=1;
                                          }
```

```
                                        System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(Environment::NewLine));
                                        System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format("y:,"));
                                        int by =1;
                                        while(by<30)
                                        {
                                               if(WhiskerArray3d[WN][by][2]  !=  0  ||
by < 2)
                                               {

       System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(WhiskerArray3d[WN][by][2] + "," ));
                                               }
                                               by+=1;
                                        }
                                        System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(Environment::NewLine));
                                        System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format("z:,"));
                                        int bz =1;
                                        while(bz<30)
                                        {
                                               if(WhiskerArray3d[WN][bz][3]  !=  0  ||
bz < 2)
                                               {

       System::IO::File::AppendAllText(saveFileDlg->FileName,
String::Format(WhiskerArray3d[WN][bz][3] + "," ));
                                               }
                                               bz+=1;
                                        }
                                 }
                                 System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(Environment::NewLine));
                                 System::IO::File::AppendAllText(saveFileDlg-
>FileName, String::Format(Environment::NewLine));
                                 WN+=1;
                          }


                   }
                    }
                    else
                    {
                           MessageBox::Show("Please click Calculate first");

                    }
                    }

   private:    System::Void    txtSampleInfoDateCreation_MouseClick(System::Object^
sender, System::Windows::Forms::MouseEventArgs^  e)
               {
                      txtSampleInfoDateCreation->Text= String::Empty;
               }
   private:    System::Void    txtSampleInfoTimeCreation_MouseClick(System::Object^
sender, System::Windows::Forms::MouseEventArgs^  e)
               {
                      txtSampleInfoTimeCreation->Text= String::Empty;
               }
```

```
    private:  System::Void  txtSampleInfoDateImage_MouseClick(System::Object^    sender,
System::Windows::Forms::MouseEventArgs^  e)
               {
                      txtSampleInfoDateImage->Text= String::Empty;

               }
    private:  System::Void  txtSampleInfoTimeImage_MouseClick(System::Object^    sender,
System::Windows::Forms::MouseEventArgs^  e)
               {
                      txtSampleInfoTimeImage->Text= String::Empty;
               }
    private:    System::Void    btnSetRefPoint1_Click(System::Object^          sender,
System::EventArgs^  e)
               {
                      SetRefPoint1=true;
                      this->setPoint1Light->BackColor                                 =
System::Drawing::Color::Orange;

               }
    private:    System::Void    btnSetRefPoint2_Click(System::Object^          sender,
System::EventArgs^  e)
               {
                      SetRefPoint2=true;
                      this->setPoint2Light->BackColor                                 =
System::Drawing::Color::Orange;

               }
    private:    System::Void    btnDiameteMeasure_Click(System::Object^        sender,
System::EventArgs^  e)
               {
                      DiameterMeasure = !DiameterMeasure;
                      if (DiameterMeasure==true)
                      {
                            MessageBox::Show("Please  click  on  either  ends  of  a  line
that is a diameter of a whisker");
                            this->WhiskerSiameterLight->BackColor                     =
System::Drawing::Color::Orange;
                            }
                      if(DiameterMeasure==false)
                      {
                             this->WhiskerSiameterLight->BackColor                    =
System::Drawing::Color::Red;
                      }

               }
    private:    System::Void    btnDiameterRecord_Click(System::Object^        sender,
System::EventArgs^  e)
               {
                       if(ImageLoaded == true)
                      {
                            if(DiameterMeasure == true)
                            {
                                   w3 += 1;
                            }
                      }
                      else
                      {}
               }
    private:    System::Void    btnWhiskerDiameterUndo_Click(System::Object^      sender,
System::EventArgs^  e)
               {
```

```
                        if(ImageLoaded == true)
                        {
                        if (DiameterMeasure == true)
                        {
                                 DiameterMeasurePointNumber=1;
                                int A = 0;
                                while (A < 3)
                                {
                                        DiameterMeasureArray[w3][A][1] = 0;
                                        DiameterMeasureArray[w3][A][2] = 0;
                                        A += 1;
                                }
                                this->ImageBox->Size                                          =
System::Drawing::Size(ImageWidth*ImageZoom + 1, ImageHeight*ImageZoom + 1);
                                this->ImageBox->Size                                          =
System::Drawing::Size(ImageWidth*ImageZoom, ImageHeight*ImageZoom);
                        }
                        }
                        else
                        {}

                }
    private:    System::Void    toolStripMenuItem3_Click(System::Object^        sender,
System::EventArgs^  e)
                {
                        SaveFileDialog^ saveImageFileDlg = gcnew SaveFileDialog;
                        saveImageFileDlg->Filter = "Tiff files|*.Tiff";
                        saveImageFileDlg->FilterIndex = 2;
                        saveImageFileDlg->RestoreDirectory = true;
                        if          (          saveImageFileDlg->ShowDialog()          ==
Windows::Forms::DialogResult::OK )
                        {
                                Bitmap^   bmp   =   gcnew   Bitmap(ImageBox->Width,   ImageBox-
>Height);
                                this->ImageBox->DrawToBitmap(bmp,                 ImageBox-
>ClientRectangle);
                         bmp->Save(saveImageFileDlg->FileName,
System::Drawing::Imaging::ImageFormat::Tiff);
                        }
                }
    private:    System::Void    toolStripMenuItem4_Click(System::Object^        sender,
System::EventArgs^  e)
                {

                        SaveFileDialog^ saveImageFileDlg = gcnew SaveFileDialog;
                        saveImageFileDlg->Filter = "Tiff files|*.Tiff";
                        saveImageFileDlg->FilterIndex = 2;
                        saveImageFileDlg->RestoreDirectory = true;
                        if          (          saveImageFileDlg->ShowDialog()          ==
Windows::Forms::DialogResult::OK )
                        {
                                Bitmap^  bmp  =  gcnew  Bitmap(ImageBox2->Width,  ImageBox2-
>Height);
                                this->ImageBox2->DrawToBitmap(bmp,              ImageBox2-
>ClientRectangle);
                                bmp->Save(saveImageFileDlg->FileName,
System::Drawing::Imaging::ImageFormat::Tiff);
                        }

                }
    private:    System::Void    btnCountWhiskerStart_Click(System::Object^       sender,
System::EventArgs^  e)
```

```
            {
                 if(ImageLoaded == true)
                 {
                       MeasureWhiskerImage1 = false ;
                       this->MeasureWhiskerLight->BackColor                    =
System::Drawing::Color::Red;
                       CountStangeShapeEruption = false;
                       this->CountStrangeShapeEruptionLight->BackColor         =
System::Drawing::Color::Red;
                       CountWhisker = true;
                       this->CountWhiskerLight->BackColor                      =
System::Drawing::Color::Orange;
                 }
                 else
                 {}
            }
   private:    System::Void    btnCountWhiskerStop_Click(System::Object^       sender,
System::EventArgs^  e)
            {
                  CountWhisker = false;
                  this->CountWhiskerLight->BackColor                           =
System::Drawing::Color::Red;

            }
   private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^  e)
            {
                  Drawing::Icon ^ icoMain = gcnew Drawing::Icon(L"app.ico");
                 Icon = icoMain;
            }
   };
   }
```