Morphology and Ion Transport in Block-Copolymer Electrolytes

by

Scott Allen Mullin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Chemical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Nitash P. Balsara, Chair
Professor John Newman
Professor Rachel Segalman
Professor Andrew Minor

Fall 2011

Abstract

Morphology and Ion Transport in Block-Copolymer Electrolytes

by

Scott Allen Mullin

Doctor of Philosophy in Chemical Engineering

University of California, Berkeley

Professor Nitash P. Balsara, Chair

Lithium metal batteries have significantly higher energy densities than current state-of-the-art rechargeable lithium-ion batteries, but lithium metal reacts with conventional liquid electrolytes, causing electrolyte depletion, lithium capacity loss, and eventual cell failure. Solid polymer electrolytes offer long-term chemical stability against lithium metal and confer significant safety advantages because they contain no volatile chemical species. Despite decades of research, two major problems prevent their implementation: dendrite growth resulting in battery failure and low conductivity at room temperature. Theoretical work has predicted that dendrite growth can be completely suppressed only by an electrolyte with a shear modulus near 6 GPa.[1] Ionic conductivity is the highest in mobile, liquid-like polymers, which typically exhibit shear moduli well below 1 MPa. In polymer electrolytes, ions are transported via segmental motion of polymer chains[2] – precluding the use of rigid polymers as ion-transport media. Numerous attempts have been made to increase the modulus of polymer electrolytes, including cross-linking the conductive homopolymer[3, 4] and using inorganic fillers[5] – approaches which usually lead to conductive polymers with shear moduli no larger than 1 MPa in the melt state. In order to conduct ions while inhibiting dendrite growth, the shear modulus of solid polymer electrolytes must be increased by several orders of magnitude without significantly impairing ion transport.

The potential of block copolymers to resolve the conflicting demands of high conductivity and high modulus has been recognized in several previous studies.[6-26] Block copolymers self-assemble into well-defined microstructures such as alternating lamellae, cylinders arranged on a hexagonal lattice, etc. In this thesis, the block-copolymer electrolyte systems composed of polystyrene-*b*-poly(ethylene oxide) (SEO) block copolymers with lithium bis(trifluoromethanesulfone)imide (LiTFSI) salt are used as a model. In SEO/LiTFSI systems, the mechanical properties are dominated by the rigid polystyrene (PS) block, whereas the conductivity depends crucially on the connectivity of the poly(ethylene oxide) (PEO) domains where the LiTFSI salt resides. The molar ratio of LiTFSI to PEO ether linkages, r (r = [Li]/[EO]) is used as a convenient expression of the salt concentration. Electrolytes composed of high-molecular-weight lamellar SEO copolymers exhibit conductivities approximately 1/3 those

composed of PEO homopolymer, but they exhibit shear moduli that are in the range of 50 to 100 MPa.[19, 20]

Electrolytes composed of low-molecular-weight lamellar SEO copolymers, however, exhibit poor conductivities that increase with increasing molecular weight. This conductivity trend was first reported by Singh et al.[20] and was reproduced across a wide range of conditions by Panday et al.[19] The authors of both of these studies hypothesized that the molecular weight trend originated from reconfigurations of the PEO chains induced by the SEO microstructure that, in turn, affected the LiTFSI dissociation and ion-transport properties. Gomez et al. reported transmission electron microscopy (TEM) data that suggested that LiTFSI was concentrated more heavily in the centers of PEO microdomains, thus supporting this hypothesis.[9] The exact mechanism underlying the conductivity trend with molecular weight remains uncertain. The studies by Singh and Panday reported only bulk conductivity properties, and that of Gomez was suggestive, but it did not link the phenomena of PEO chain rearrangements with LiTFSI binding properties or ionic mobilities.

From previous studies of block-copolymer electrolytes it is clear that the copolymer morphologies strongly affect bulk ion transport properties.[6-26] However, only the Singh[20] and Panday[19] references systematically explored the molecular-weight trend while keeping the morphology constant, and only ionic conductivity had been used to probe bulk ion-transport properties. In SEO/LiTFSI mixtures, we assume that LiTFSI segregates completely into the PEO microdomains, thus giving three relevant species for defining electrolyte properties: the solvent (PEO), the cation (Li^+) and the anion ($TFSI^-$). This system is thus a binary electrolyte, and the ion-transport properties can be completely described by three parameters. These parameters can be chosen as the conductivity, the salt diffusion coefficient, and the cation transference number, as detailed by Ma et al.[27] No previous studies have measured salt diffusion coefficients or transference numbers in block-copolymer electrolytes. Since complete sets of ion transport properties are known for PEO/LiTFSI systems,[28] complete sets of ion transport properties for SEO/LiTFSI systems having various morphologies could provide insight into the effects of block-copolymer electrolyte morphologies on ion-transport properties. The ion-transport measurement approach used in this work closely follows that outlined by Ma et al., which was developed from concentrated solution theory.[27] This is the only theoretically rigorous approach for measuring the salt diffusion coefficient and transference number for polymer electrolytes. Concentrated solution theory requires knowledge of the number of species present in the electrolyte, but knowledge of microscopic details such as ion pairing are not necessary. However, concentrated solution theory was developed with the assumption of a homogeneous electrolyte, and correction factors are occasionally needed to understand or describe effects arising from the microstructures of SEO-based electrolytes. These correction factors include the volume fraction of the PEO phase, ϕ_{PEO}, and a tortuosity factor to account for the geometry of the PEO phase, f. The value of f depends only on the specific microstructure (i.e., lamellar or cylindrical) and can be used to predict salt diffusion coefficients in block-copolymer electrolytes based on those of the corresponding homopolymer electrolytes, as outlined by Sax and Ottino.[29] The value of ϕ_{PEO} varies from sample to sample, and can be used in combination with f to predict block-copolymer electrolyte conductivities, as outlined by Singh et al.[20] Details on the usage of these parameters are discussed as needed in Chapters 2, 3 and 5.

The central goal of this thesis is to understand the relationship between ion transport properties and morphology in block-copolymer electrolytes. The bulk of the work focuses on the systematic measurement of ion transport properties across various morphologies. However, in the course of that work it became clear that the effects of ionic currents on block-copolymer morphologies were unknown. To address this, a study was undertaken in which block-copolymer electrolyte morphologies were determined *in situ* while ionic currents were passed, and the result was that ionic currents can dramatically alter block-copolymer electrolyte morphologies.

The outline of this thesis is described here. Chapter 1 describes common methods for preparing and characterizing block-copolymer electrolytes. In the study detailed in Chapter 2, the restricted-diffusion technique was used to measure salt diffusion coefficients for lamellar block-copolymer electrolytes. Lamellar microstructures were found to alter the diffusion/relaxation behavior of the polymers – a result that was attributed to the presence of grain boundaries. In the study detailed in Chapter 3, conductivities were measured as a function of temperature and preparation history for cylinder-forming block-copolymer electrolytes wherein the PEO phase was the minor component inside of a PS matrix. Unannealed samples with cylindrical morphologies were found to exhibit normalized conductivities similar to those with lamellar morphologies. Upon annealing, however, the conductivities dropped substantially for all samples except high-molecular-weight copolymers with lamellar morphologies. These results were attributed to poor connections of the PEO phase across grain boundaries that develop during annealing, and indicate that the molecular-weight dependent conductivity trend in lamellar block-copolymer electrolytes arises, at least in part, due to grain-boundary effects. The study detailed in Chapter 4 outlines the simplicity and efficacy of the resonant soft X-ray scattering (RSOXS) technique for determining the morphologies of high-molecular-weight block-copolymer electrolytes with domain spacings larger than 100 nm. The study in Chapter 5 measured lithium transference numbers for a high-molecular-weight lamellar block-copolymer electrolyte. The goal was to determine whether the transference numbers were the same as those of analogous PEO electrolytes. Confirmation of SEO and PEO electrolytes having the same transference numbers would indicate that the observed molecular weight trends in lamellar SEOs arise due to geometric and connectivity effects rather than intrinsic differences between bulk PEO and confined PEO channels in SEO microstructures. The comparison between PEO and SEO transference numbers could not be accurately made. In the study detailed in Chapter 6, ionic currents were used to induce morphological changes in SEO-based electrolytes, including a previously unseen structure which we termed a "gradient crystal". The observed gradient crystals were composed of the gyroid morphology and exhibited position-dependent domain spacings. Finally, the Appendix sections outline some of the key details underpinning the experiments in the Chapters.

Table of Contents

List of Figures

List of Tables

List of Symbols

A_0	power-law prefactor, nm
A	electrode area, cm^2
a, b	chosen integration limits for Contin fit, 1/s
c	electrolyte concentration, mol/L
c_{ref}	concentration of reference electrolyte, mol/L
c_{T}	total molar salt concentration, mol/L
c_0	molar concentration of solvent in the solution, mol/L
d	domain spacing, nm
D	mutual diffusion coefficient, cm^2/s
$\mathfrak{D}$	self-diffusion coefficient, cm^2/s
$D_{1,\mathrm{avg}}$	$\Gamma_{1,\mathrm{avg}}L^2/\pi^2$, cm^2/s
D_{avg}	average of $D_{1,\mathrm{avg}}$ obtained from multiple relaxation profiles, cm^2/s
D_{SEO}	D_{avg} for a given SEO sample, cm^2/s
D_{PEO}	D_{avg} for a given PEO sample, cm^2/s
f	morphology factor
$f()$	distribution function fit to decay profiles, V
$f_n()$	normalized distribution function
F	Faraday's constant. 96,487 C/equivalent
i	current density, mA/cm^2
I_0	incident X-ray beam intensity from ion gauge reading, arbitrary units
Im	imaginary component of area-specific impedance, Ω-cm^2
k_0	baseline offset due to thermocouple effect, V
k_1	exponential prefactor, V
L	sample thickness, mm
m	molality of solution, mol/kg (Chapter 2)
m	slope parameter from current-interrupt experiments (Chapter 5)
M_{PEO}	number-averaged molecular weight of PEO block, kg/mol
M_{PS}	number-averaged molecular weights of PS block, kg/mol
M_{salt}	molar mass of LiTFSI
PDI	polydispersity index
q	magnitude of X-ray scattering wave vector, 1/nm
q^*	scattering vector of peak corresponding to {100} scattering planes
q'	scattering vector of peak at lowest-observed q-value
Q	constant-phase-angle element for circuit fitting
r	molar ratio of LiTFSI to ether oxygens in PEO
r_{m}	mass ratio of LiTFSI to ether oxygens in PEO
r_0	equilibrium value of r
R	electrolyte resistance
Re	real component of area-specific impedance, Ω-cm^2
t	time (various units)
t_{j}	elapsed time at point of current interrupt
t_+^0	cationic transference number with solvent velocity as reference
U_{j}	concentration overpotential at time t_{j} for current-interrupt experiments
$U_{\Delta\mathrm{c}}$	open-circuit potential measured from concentration-cell experiments
$V(t)$	open-circuit potential with initial fast-decay period removed, V

$V_{\text{concentration-cell}}(t)$	open-circuit potential for concentration-cell experiments
$V_{\text{interrupt}}(t)$	open-circuit potential for current-interrupt experiments
x	power-law exponent

Greek

α	dimensionless time (Dt/L^2)
σ	conductivity, S/cm
λ	X-ray wavelength, nm
ϕ_{PEO}	volume fraction of PEO in a given SEO
θ	X-ray scattering angle
γ_{+-}	mean molal activity coefficient
$\mathit{\Gamma}$	inverse time constant, 1/s
$\mathit{\Gamma}_{1,\text{avg}}$	1st moment / 0th moment
$\mathit{\Gamma}_{2,\text{avg}}$	2nd moment / 1st moment
ρ	electrolyte density, g/cm^3
τ	dimensionless time for current interrupt experiment
ω_{e}	mass fraction of salt in the electrolyte solution

Acknowledgements

I have been influenced by a lot of people during my time at Berkeley. Many people have contributed to my work and research, and many more have provided moral support and mental well-being. My advisor, Nitash Balsara, has perhaps had the strongest influence on me. Nitash provided an interesting starting point, and then he gave me the freedom and ample resources to follow interesting leads, try new experiments and build new equipment. I had my fair share of snags and dead ends, and Nitash saw me through those with encouragement and the perspective of "lessons learned" rather than failures. For all of that, thank you Nitash.

Professor John Newman was a patient, helpful mentor. I owe most of my electrochemical knowledge to long conversations with him. He taught me that I did not really know what a potential was, and then he provided me with the tools to figure out what that meant. Thank you, Professor Newman.

I thank Professor Rachel Segalman for providing me with thoughtful scientific and career advice over the past few years. She also served as a sharp overseer of my qualifying exam, and as a gracious member of my thesis committee.

I also thank Professor Andrew Minor for his kind review of my thesis work, and his post-doctoral student, Dr. Frances Allen, for her contributions to our understanding about SEO systems.

My labmates over the years have contributed substantially to my research and to my development as a scientist. I had a lot of fun working with everyone in the Balsara lab. Drs. Nisita Wanakule and Alisyn Nedoma taught me anionic synthesis, showed me the ropes, and kept the lab cheerful. Greg Stone and I started at the same time and weathered many storms together, often at neighboring desks. More than a great friend, Greg was a great collaborator. He made substantial contributions to the diffusion study in Chapter 2 and the structural study in Chapter 6. Alex Teran has been another great friend and labmate. He synthesized the "magic" polymer we used to great effect in Chapter 6, and he was always a great sounding board. Alex, and his undergraduate assistant Rodger Yuan, also contributed to the conductivity results in Chapter 3. I worked extensively with Dr. Daniel Hallinan to build, equip and manage the labs used for the measurements in this work. Dan is always cheerful, and he always challenges ideas I take for granted. Dan made some wonderful contributions to the impulse-sealing technique that was crucial to the work in Chapter 6. I also had the pleasure of learning about separators from David Wong, and I benefitted from his sense of humor. Shrayesh Patel (part of the regular burrito contingency) and Dr. Anna Javier (a "real" chemist) taught me about electronically conducting polymers. Keith Beers was always a good friend, and he taught me about single-ion conductors. Keith also provided the excellent TEM images in Chapter 4. Dr. Guillaume Sudre brought great humor to the lab, and Professor Moon Park brought great intensity and focus. I also had a lot of fun working with Nicholas Young. Drs. Evren Ozcam, and Sebnem Inceoglu brought a lot of personality and flair to the lab. I am proud to have worked with such great colleagues.

Dr. Paul Albertus, Nathan Craig and Maureen Tang from the Newman lab were all great friends, and they gave me a lot of advice and support over the years. A fruitful collaboration based on thin film cathodes designed and built by Drs. Wyatt Tenhaeff and Nancy Dudney from Oak Ridge National Laboratory taught me a great deal about electrodes.

Dr. Joel Stettler and his undergraduate assistant Yuping Li performed some interesting NMR-based lithium diffusion measurements on some of the block-copolymer systems. Those measurements gave us some insight into the nature of LiTFSI diffusion through SEO systems. I enjoyed working with Joel, and I appreciated the support from his advisor, Professor Jeff Reimer.

I had the pleasure of working with several undergraduates. Sharmishtha Pal developed a lot of the LabVIEW modules I used, and she and Monica Castañeda helped with the story in Chapter 3. Vincent Chen worked with me on PEO conductivity measurements. All of them were wonderful colleagues.

Jim Breen fixed all of my broken glass; Eric Granlund machined most of my experimental setups; and Henry Chan and Steve Hoagland did most of the electronics work my experiments required. Susan Lauer, Cindy Stevenson, Micah Liedeker and Charlotte Standish at LBL made sure I got paid and helped me navigate the LBL bureaucracy. At UCB, Aileen Harris, Rocio Sanchez and Fred Deakin did the same.

At the Advanced Light Source, I was supported extensively by the efforts of Dr. Alex Hexemer, Dr. Cheng Wang, Dr. Anthony Young, Eric Schaible, Steven Alvarez, and Peter Reichert. They made my ALS experiments successful and fun.

Finally, I thank Victoria Tran and Autumn Trullin for keeping my focus on happiness and the real world. To mom, dad and Shawn: thanks for supporting my crazy idea to be a scientist, even though it kept me away from you all these years. The same goes for the rest of the Mullin, Baker, Bennett, Bittle and Shamblin families. To all of my other Berkeley friends: thanks for keeping my years here exciting and light-hearted. Thank you all!

This work was conducted under the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy's Batteries for Advanced Transportation Technologies (BATT) Program, under Contract No. DE-AC02-05CH11231. I was also supported by fellowships from Arkema and Tyco Electronics. Some experiments were conducted at the L.B.N.L Advanced Light Source and National Center for Electron Microscopy, which are supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Chapter 1 – Block-Copolymer Electrolytes

1.1 Introduction

The focus of this dissertation is to understand the properties of electrolytes based on polystyrene-*b*-poly(ethylene oxide) (SEO) block-copolymer mixtures with lithium bis(trifluoromethanesulfone)imide (LiTFSI) salt. The poly(ethylene oxide) (PEO) block solvates LiTFSI and provides ion conduction whereas the polystyrene (PS) block provides mechanical strength. The interplay between block-copolymer microstructure and ion transport is a key theme of this work; thus it is necessary to understand the methods used to create and characterize the block-copolymer electrolytes. First, the methods used to synthesize, purify, and characterize SEO diblock copolymers are described. Next, the methods used to prepare SEO/LiTFSI electrolyte mixtures are described. Finally, routine SEO/LiTFSI characterization methods are described, including small-angle X-ray scattering (SAXS), and conductivity measurements.

1.2 Block-Copolymer Synthesis and Characterization

SEO block copolymers were synthesized in benzene on a vacuum line using sequential anionic polymerization as described extensively in previous work[19, 20, 24, 30]. The PS block of each copolymer was synthesized first using *sec*-butyllithium as the initiator and purified styrene as the monomer, and an aliquot was taken and analyzed via gel permeation chromatography (GPC) to determine its polydispersity index (PDI_{polymer}) and number-averaged molecular weight using a Viscotek OmniSEC separations module and triple-detector system calibrated using PS standards with tetrahydrofuran (THF) as the eluent. The PDI_{polymer} of the PS blocks of all of the SEO copolymers in this study were less than 1.10, with the exceptions of SEO(247-116), SEO(352-166) and SEO(240 – 269) which had PS-block PDI_{polymer} values of 1.19, 1.43 and 1.20, respectively. The PEO block is subsequently grown from the living PS chain using P4 *tert*-butylphosphazene base as a promoter. The volume fractions of each block and the number-average molecular weight of the PEO block (M_{PEO}) were determined using ^{1}H nuclear magnetic resonance (NMR) spectroscopy. The PDI_{polymer} of each resulting SEO copolymer was determined from a Viscotek OmniSEC GPC calibrated with polystyrene standards and using dimethylformamide as the eluent. The PDI_{polymer} values for SEO(247-116) and SEO(352-166) could not be conclusively determined due to poor chromatographic separation on the GPC that indicated unrealistic values significantly less than the PDI_{polymer} values of the constituent PS blocks.

The copolymers are designated by the number-averaged molecular weights of the PS and PEO blocks in kg/mol as SEO($M_{\mathrm{PS}} - M_{\mathrm{PEO}}$). PEO homopolymers are designated by the number-averaged molecular weight of the PEO as PEO(M_{PEO}). The PEO(27) homopolymer used in this study was purchased from Polymer Source and used as-received.

Table 1 – Polymers used in this Thesis

Lab Name	Publication Name	M_{PS} (kg/mol)	M_{PEO} (kg/mol)	ϕ_{EO}	$PDI_{polymer}$	Morphology	Domain spacing (nm)
SEO-AAT-2	SEO(1.7-1.4)	1.7	1.4	0.44	1.04	LAM	7.6
SEO1	SEO(36-24)	36	24	0.38	1.10	LAM	51.8
SEO4	SEO(40-54)	40	54	0.55	1.20	LAM	82.4
SEO5	SEO(74-98)	74	98	0.55	1.14	LAM	113.7
SEO6	SEO(16-16)	16	16	0.48	1.09	LAM	34.9
SEO7	SEO(53-68)	53	68	0.54	1.05	LAM	101.9
SEO8	SEO(37-25)	37	25	0.39	1.04	LAM	52.7
SEO9	SEO(54-23)	54	23	0.28	1.04	CYL	47.8
SEO11	SEO(6-7)	6.4	7.2	0.52	1.02	LAM	20.5
SEO15	SEO(247-116)	247	116	0.30	-	CYL	117
SEO16	SEO(352-166)	352	166	0.30	-	CYL	124
SEO17	SEO(240-269)	240	260	0.50	1.26	LAM	242

1.2.1 Purification

Polymers were precipitated in hexanes and then decanted to remove residual lithium salts that originated from the synthesis process. The polymers were then re-dissolved in benzene and precipitated two times. The *tert*-butyl phosphazene base is difficult to remove, particularly for low-molecular-weight samples that include a larger initial quantity of it. Its presence is indicated by discoloration of polymer samples (yellow, black, brown, etc.) from a pure white color. Phosphazene-contaminated polymers were dissolved in benzene, heated to 50 °C on a hotplate, and then passed through a coarse filter packed with neutral alumina three times. Finally, the polymers were vacuum filtered through a 0.2 μm filter and then freeze-dried in a lyophilizer (Millrock LD85).

1.2.2 Freeze-drying

Freeze-drying (synonymous with lyophilizing) was regularly used in this work to prepare purified polymers and polymer electrolytes for further handling. The samples are first dissolved in an appropriate solvent and then frozen. The frozen polymer/solvent mixture is then placed on a cold surface and held under vacuum. The frozen surface is slowly heated, and the solvent is removed via sublimation under vacuum. This process results in a polymer "fluff" that has a very large void fraction in the range of 50 to 80%. This is particularly advantageous for polymers such as SEO that are composed of large fractions of a glassy block. In this case, the large void fraction allows for complete solvent removal and the "fluff" is easy to handle with tweezers for sample preparation.

The freeze-drying process was originally performed on a vacuum line, and the cold plates were generated by placing metal plates in liquid nitrogen prior to freeze-drying. This procedure became problematic, particularly for large samples in which the solvent volume would readily clog the solvent traps and the cooling plate would warm too rapidly, such that the sample would liquefy and re-dissolve before freeze-drying could complete. To address this problem, a lyophilizer (Millrock LD85) was procured. The lyophilizer was later modified with a valve assembly such that desiccators could be loaded into the unit and electrolyte samples could be freeze-dried without exposure to air.

In a typical procedure for freeze-drying purified SEO samples after synthesis, SEO samples were placed in small jars or vials and dissolved to a concentration of 10 to 20 mg/mL in purified benzene. The sample containers were then loosely covered with aluminum foil and the entire jar was frozen. Sample jars were frozen by placing them into dry ice/isopropanol slurries for a few hours. The jars were then placed on the shelves of the lyophilizer and held at a shelf surface temperature of -70 °C overnight with a slight vacuum to seal the unit. The condenser coils were held constant at a temperature of -83 °C during the entire unit operation. The shelf temperature is then increased at a rate of 20 °C per day until a temperature of 10 °C is reached. For the final step, the shelf temperature is held at 60 °C to remove residual solvent.

SEO/LiTFSI samples were freeze-dried similarly, except that LiTFSI solutions in tetrahydrofuran (THF) or methylethylketone (MEK) were mixed into the SEO/benzene solutions at the desired ratio via a micropipette. The solutions were stirred overnight and were then placed into a desiccator. The desiccator was sealed and then placed into the lyophilizer. The desiccator valve connected to a feedthrough that routed through an external Swagelok® valve and then back into the lyophilizer vacuum chamber. The lyophilizer shelf temperature was held at -70 °C overnight, during which time the entire unit was held under vacuum, including the portion of tubing up to the desiccator. Next, the external valve was closed, and the lyophilizer was vented to atmosphere. The piping between the external valve and the desiccator was still under vacuum. The desiccator valve was then opened, and the unit was re-sealed and placed back under vacuum. The unit was held until full vacuum level was achieved (10 mbar), and then the external valve was opened, and the desiccator contents were then exposed to the active vacuum of the lyophilizer chamber. The freeze-drying process then continued as usual. Once the freeze-drying process completed, the external valve was closed to isolate the desiccator, and the chamber was vented to atmosphere. The desiccator valve was then closed, and the desiccator was transferred back into a glovebox.

It is important to choose an appropriate solvent for freeze-drying. The polymer/solvent solution must remain frozen during the entire freeze-drying process, but the solvent must also sublimate readily under vacuum. Good freeze-drying solvents typically have melting points between 0 and 25 °C and boiling points between 70 and 100 °C. Examples of good freeze-drying solvents include benzene, cyclohexane, *tert*-butanol, and water. Examples of poor freeze-drying solvents include hexanes, most alcohols, THF, and MEK due to their low melting points and N-methyl pyrrolidone (NMP), dimethyl sulfoxide (DMSO), and dimethyl formamide (DMF) due to their high boiling points. These constraints on solvent choice are imposed upon the problem of solvent choice for dissolving two unlike polymers and a salt such that a single solvent for freeze-drying SEO/LiTFSI has not been found. Freeze-drying is often successful when small amounts of THF or other volatile species are used as cosolvents, although they bypass the condenser coils and collect in the pump oil, thus necessitating regular pump-oil changes. PEO samples are typically insoluble in benzene/THF mixtures with LiTFSI, thus requiring a different solvent choice. Empirical tests showed that *tert*-butanol/MEK mixtures provided good freeze-drying results for PEO/LiTFSI, although most PEO/LiTFSI samples are prepared by solvent evaporation and heating without freeze-drying.

1.3 Electrolyte Sample Preparation

Argon gloveboxes (MBraun and Vacuum Atmospheres Company) with oxygen and water at sub-ppm levels were used for all sample preparation and testing steps. Freeze-dried neat (salt-free) polymer samples were placed in clean, pre-weighed vials and dried at 90 °C under vacuum in a glovebox antechamber for three days. Neat polymers were weighed after drying. LiTFSI (Novolyte) was taken from its as-received, air-free packaging, transferred into a vial inside of a glovebox, and then dried at 120 °C under vacuum in a glovebox antechamber for three days. Benzene and THF were purified using an MBraun solvent purification system to remove water and were subsequently vacuum distilled. Dry LiTFSI was weighed into a volumetric flask, and then THF was added to the fill line to create a 100 mg/mL solution. Each neat polymer sample was dissolved in benzene to a concentration of 15 mg/mL, and THF/LiTFSI solution was added using a micropipette to obtain the desired salt concentration. Polymer solutions containing LiTFSI were placed inside an air-tight desiccator and transferred into the lyophilizer. Samples were then freeze-dried without exposure to air.

1.3.1 Sample Pressing (Hot-Press and Hand-Press Methods)

Electrolyte samples are typically prepared from a freeze-dried "fluff" unless otherwise specified. In order to get the electrolyte into a usable form for electrochemical measurements, it is necessary to compact the fluff into a void-free form. Void-free films are obtained by loading the electrolyte fluff into a plastic fixture containing a cylindrical bore. The back end of the fixture is blocked by loading a plastic rod and placing the fixture upright on a clean surface in the glovebox; the fluff is tamped down from the top with a separate plastic rod to create a pellet. Low-molecular-weight SEO copolymers are too sticky to be formed into pellets by this method. Instead, they are either formed into balls with a spatula or they are melted at 100 °C on a hotplate and then scraped into a spacer. Regardless of the preparation procedure, this type of pellet is typically opaque and is presumed to contain a relatively high void fraction. Next, the pellet is placed into a spacer between two sheets of clean material, and the entire assembly is pressed together. The spacer material depends on the experiment of interest. For example, electrochemical measurements will typically use plastic spacers such as Garolite G-10 or Kapton, whereas SAXS measurements may use metal spacers or washers that seat into a specific sample holder design. Typical samples for electrochemical measurements are pressed between sheets of pouch material or low-stick surfaces such as Teflon or other fluoropolymers.

Sample preparation procedures evolved over time as better equipment was purchased. The oldest method is to use a hand press that consists of two metal blocks with a large screw handle. This type of press was typically pre-heated on a hot plate and then the sample assembly was placed inside. Then the press was tightened by hand, and allowed to anneal on the hot plate for a period of time. This method poses difficulties such as poor thermal control, low applied pressure on the sample, and frequent breakdown of the press due to over-tightening. These difficulties prevent its effective implementation for high-molecular-weight copolymers, but optically clear and visually void-free samples of low-molecular-weight samples can be prepared by this method. High-molecular-weight samples require the use of a hot press. A Carver Mini-C hot press was used for this purpose in the restricted diffusion study in Chapter 2. The Carver press is pumped up by hand and includes a manual pressure gauge. The platens on this press are feedback-controlled, and the temperature at the sample position is highly reproducible once calibrated (usually 10 °C hotter than the platen temperature for thin samples). This press was later supplanted by a larger pneumatic press manufactured by Janesville Tool and Mfg. Co. Samples in Chapter 3 were prepared with this press. The Janesville press is advantageous for reproducible sample preparation because it exhibits better temperature control, the platens are fixed at each corner to a post for stability, the pneumatic pressure provides a consistent sample pressure, and the dwell time can be programmed from 0.01 s to many hours. The force between the platens (in lb_f) can be approximated as 54.1 × the inlet gas pressure (in psi) per the manufacturer's specifications. Most copolymers pressed in this manner will be optically clear, with the exception of high-molecular-weight samples.

1.4 Small-Angle X-Ray Scattering (SAXS)

Small-angle X-ray scattering (SAXS) results are commonly used in this work to determine the morphology and domain spacings of block copolymers and block-copolymer electrolytes. Typical SAXS experimental details are described in this section. The reader is referred to the literature for advanced details.[31, 32] SAXS experiments were performed at various synchrotron X-ray sources, including the Advanced Light Source (ALS), the Advanced Photon Source (APS), and the Stanford Synchrotron Radiation Lightsource (SSRL). All of the SAXS experiments presented in this dissertation were performed at the ALS beamline 7.3.3. A SAXS experimental setup is shown schematically in Figure 1. This setup describes the essential elements of any SAXS experiment, although the composition and mounting of the sample changes across experiments.

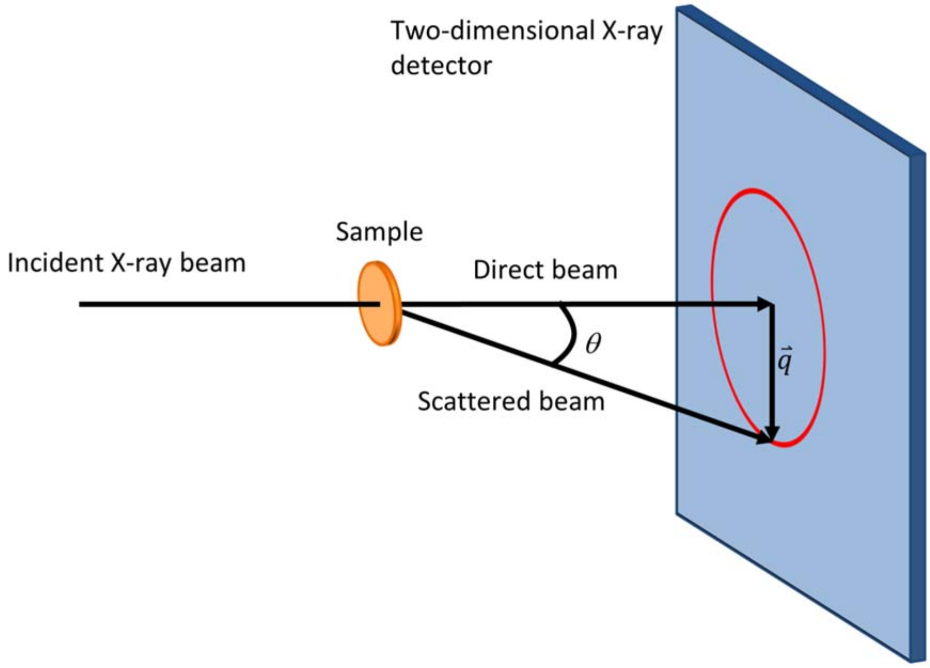


Figure 1 – Schematic of generic SAXS experimental setup

In this schematic, an incident synchrotron X-ray beam hits the sample. The beam energy, 10 keV, is in the hard X-ray range and few of the X-rays are absorbed. Most of the X-rays transmit directly through the sample or are scattered due to the polymer microstructure. The X-rays pass through an evacuated flight tube with Kapton windows (not shown on Figure 1) and strike a two-dimensional (2D) detector. A beamstop (not shown in Figure 1) is used to absorb the intense, direct beam in order to protect the detector from damage. The scattered beam at any point on the detector diverges from the direct beam with a scattering angle θ. The scattering vector, $\vec{q}$, is the difference between the scattered beam and direct beam vectors. One-dimensional (1D) profiles are obtained from the 2D SAXS images by azimuthal integration and the results are displayed as integrated intensity (in arbitrary units) versus the magnitude of the scattering vector, q, as shown in Equation 1.

$$q = 4\pi\lambda^{-1}\sin\left(\theta/2\right) \qquad [1]$$

where λ is the wavelength of the X-rays (0.124 nm for a 10 keV beam). SAXS data were reduced from 2D to 1D form using various versions of the Nika macro for Igor Pro, which is written and freely distributed by Jan Ilavsky, a beamline scientist at the Advanced Photon Source.[33] Defects in the 2D image are masked off before the 1D integration is performed. Common defects include gaps between the detector quadrants, the beamstop or other obstructing objects, bad pixels, bad detector quadrants, and streaks emanating from the beamstop (usually direct beam reflections from the sample, sample stage, or beamstop). Masking a 2D image has little effect on the intensity of the resultant 1D profile because the integrated intensity is normalized to account for the masked portions, although it can adversely affect the signal/noise ratio at a given q value. Overly aggressive masking can, however, reduce the visible q-range in the data, particularly at regions near the beamstop and near the detector corners.

A particularly important q value is that of the primary scattering vector, q^*, which corresponds to the {100} family of reflection planes for a given morphology. The domain spacing, d, is calculated from the scattering peak at the lowest observed q value, q', as $d = {}^{2\pi}\!/_{q'}$. The q' peak corresponds to the correlation length in the disordered morphology, the {100} planes for the lamellar morphology, and the {211} planes for the gyroid morphology. Thus, the q^* and q' peaks are the same for most morphologies. This distinction is made because the {100} reflection is forbidden in the gyroid morphology and the scattering peak in the disordered morphology arises from correlations between the polymer blocks arising from the fact that they are covalently bonded – there are no true "domains" or reflection planes in the disordered morphology since all of the components are mixed on a molecular level. 1D scattering profiles are often normalized by the incident ion gauge reading (I_0, an arbitrary measure of the incident beam intensity) in order to account for time-dependent variations in the incident beam intensity.

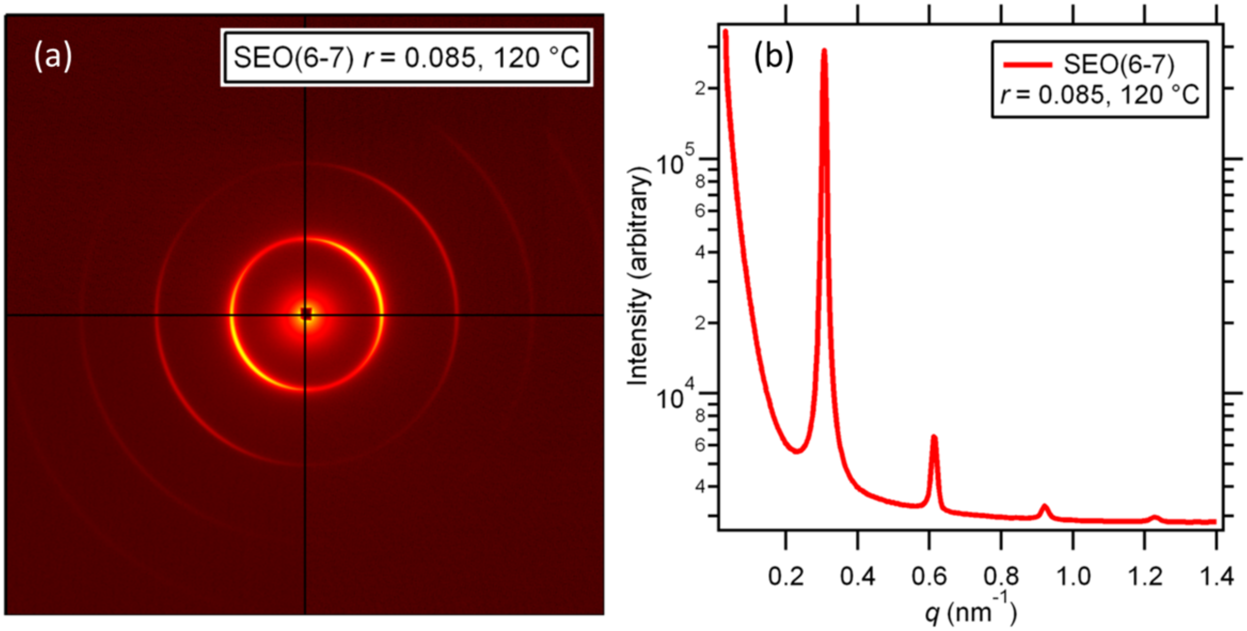


Figure 2 – Example SAXS data for SEO(6-7) *r* = 0.085 at 120 °C.
(a) 2D detector image showing evenly-spaced rings. The black lines are gaps between the detector quadrants. The dark square in the center is the beamstop. The top 20% of the image has lower intensity due to interference with the post-sample ionization chamber. (b) 1D profile of the 2D image in (a), obtained after masking off the defects mentioned. The integer spacing between the peaks is characteristic of the lamellar morphology. The q* peak is located at 0.307 nm-1 and corresponds to a domain spacing of 20.5 nm.

Diblock-copolymer morphologies are identified by their characteristic SAXS patterns. In the simplest block-copolymer cases, the morphologies can be identified by the spacing between the scattering peaks. For instance, the lamellar morphology gives scattering peaks at integer ratios relative to q^*, as shown in Figure 2. The first few scattering peaks for the morphologies identified in this work are shown in Table 2. Note that the first allowed peak for the gyroid morphology is found at $q/q^* = \sqrt{6}$, such that the peak locations relative to the first observed peak are located at q/q' = 1, $\sqrt{4/3}$, $\sqrt{7/3}$, etc. The 2D scattering profiles contain additional information such as the sample alignment or orientation with respect to the beam. This information is discussed in further detail when relevant.

Table 2 – Scattering peak locations for common block-copolymer morphologies

Structure	Abbreviation	Scattering-peak locations, q/q*
Lamellae	LAM	1, 2, 3, 4, 5, 6 …
Cylinders	CYL	1, $\sqrt{3}$, 2, $\sqrt{7}$, 3, $\sqrt{12}$ …
Gyroid	GYR	$\sqrt{6}$, $\sqrt{8}$, $\sqrt{14}$, $\sqrt{16}$, $\sqrt{20}$, $\sqrt{22}$ …
Disordered	DIS	1 (broad, single peak)

1.5 Conductivity Measurements

Alternating-current (ac) impedance spectroscopy is among the simplest electrochemical characterization technique for electrolytes, the results of which can be used to calculate electrolyte conductivities. Impedance spectroscopy experiments were typically performed using 300 µm-thick ion-blocking stainless steel electrodes. Conductivity experiments were set up by assembling electrolyte samples inside Swagelok cells or pouch cells with one metal shim on each side. Samples were removed after the measurements were taken to measure their thicknesses with a micrometer. All of the ac impedance measurements in this work were performed on using a potentiostat (Bio-Logic VMP3), and the measurement frequency was swept from 1 MHz to 1 Hz with logarithmically spaced points. Measurements were performed on a given sample at peak-to-peak voltages of 10, 50, and 100 mV to ensure that the electrolyte response was within the linear regime. If this hypothesis holds, then the Nyquist impedance plots (-Im vs. Re, where Im is the magnitude of the imaginary component and Re is the magnitude of the real component of the impedance response) have the same shape and magnitude at each applied potential. This was true for all of the samples in this study, and reported data were obtained from experiments with 100 mV peak-to-peak voltages due to the higher signal/noise ratio that is obtained. Figure 3 shows a Nyquist plot of SEO(53-68), which is typical of data obtained from SEO samples. The impedances of leads and cell fixtures are more than three orders of magnitude lower than those of the samples.

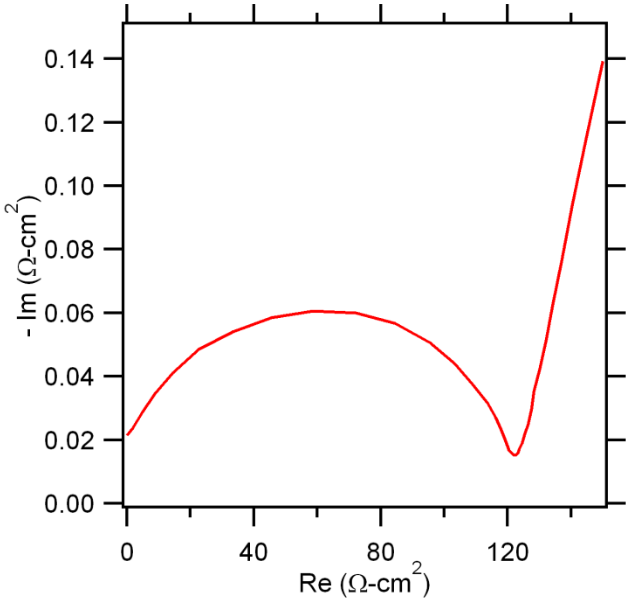


Figure 3 – Typical Nyquist impedance plot.
The trace is the negative of the imaginary area-specific resistance (-Im) versus real area-specific resistance (Re) in units of Ω-cm^2 for SEO(53-68) at 90 °C and r = 0.085. The range from 1 MHz to 300 Hz is visible, and the low-frequency minimum occurs near 5 kHz (Re = 123 Ω-cm^2). The sample area is 0.117 cm^2, and the sample thickness is 589 µm, giving a conductivity of 0.48 mS/cm. The frequency at the top of the semicircle is around 150 kHz.

The Nyquist impedance plot in Figure 3 shows typical behavior for the simple case of an electrolyte between blocking electrodes. The measurement frequency decreases from left to right along the trace. At high frequencies, the signal is dominated by capacitance from the instrumentation, particularly the cables. At low frequencies, the signal is dominated by capacitance that results from charging at the electrode/electrolyte interface due to the blocking nature of the electrodes – visible in Figure 3 as the tail. At intermediate frequencies, the signal arises from the electrolyte – visible in Figure 3 as the semi-circle. The electrolyte resistance, R, can be taken as the low-frequency minimum of the semicircle, which occurs at about 123 Ω-cm^2 in Figure 3. Alternatively, R can be extracted from a fit to the semicircle, which is well-described in most cases as a circuit composed R in parallel with a constant-phase-angle element, Q. It is also possible to capture the tail in the fit by modeling the same Q/R circuit with an additional constant-phase element in series ($Q_1/R + Q_2$). These fits can be readily performed using the built-in fitting functionality in the VMP3 software, EC-Lab. The R values that result from each of these methods are typically within a few percent of one another. Once R is obtained, the electrolyte conductivity, σ, can be calculated from Equation 2,

$$\sigma = \frac{L}{RA} \qquad [2]$$

where L is the sample thickness and A is the area of the sample. A is taken as the spacer area unless otherwise specified.

Chapter 2 – Salt Diffusion Coefficients in Block-Copolymer Electrolytes

ABSTRACT

The salt diffusion coefficient in a series of nanostructured block-copolymer electrolytes was measured in a symmetric lithium/polymer electrolyte/lithium cell using the restricted-diffusion technique. The decay of the open-circuit potential as measured by this technique was analyzed by a Laplace inversion algorithm to give the distribution of relaxation processes characteristic of the electrolytes. The distribution function was characterized by two parameters, an average diffusion coefficient, D_{avg}, and a polydispersity index, $PDI_{\text{diffusion}}$, which is a measure of the width of the distribution. We compare these parameters obtained from a series of nearly symmetric polystyrene-*b*-poly(ethylene oxide) (SEO) block-copolymer electrolytes containing lithium bis(trifluoromethanesulfone)imide salt (LiTFSI) with those obtained from a homogeneous poly(ethylene oxide) (PEO)/LiTFSI mixture. D_{avg} of the SEO/LiTFSI mixtures increases with increasing molecular weight of the PEO block, M_{PEO}, and reaches a plateau of (2/3)D_{PEO} when M_{PEO} exceeds 50 kg/mol (D_{PEO} is the average salt diffusion coefficient in PEO homopolymer). The $PDI_{\text{diffusion}}$ values obtained for SEO copolymers are significantly higher than those obtained in PEO homopolymer.

2.1 Introduction

To our knowledge, all previous experimental characterization studies of ion transport in block-copolymer electrolytes[6-26] are restricted to the determination of ionic conductivity. In the case of systems wherein only one of the microphases contributes to ionic conduction, the conductivity, σ, is proportional to the volume fraction of the conducting block, ϕ_{PEO}. We thus write

$$\sigma = \sigma_{PEO}\phi_{PEO}f \qquad [3]$$

where σ_{PEO} is the intrinsic conductivity of the conducting microphase and f is a tortuosity factor. In theory, the upper bound for f in the case of randomly oriented lamellar grains is 2/3.[20] This bound is obtained in theory when the additional resistance or conductance due to the presence of grain boundaries is negligible. Experiments thus far suggest that the f = 2/3 limit is realized in high-molecular-weight block-copolymer electrolytes.[19, 20]

It is well-known, however, that the performance of electrolytes in batteries depends on several other properties such as the salt diffusion coefficient, the activity coefficient in the electrolyte, and the cation transference number, as outlined by Newman and coworkers.[27, 34]

There are no papers that address the effect of block-copolymer electrolyte morphology on these parameters. The purpose of this Chapter is to present experimental data on the effect of morphology on salt diffusion coefficients in block-copolymer electrolytes. This parameter is important for predicting the concentration gradients that develop in cells during charge-discharge cycles. We have measured the salt diffusion coefficient using the restricted-diffusion method, as outlined by Newman and Thompson.[35, 36]

Restricted-diffusion data analysis does not require knowledge of salt speciation, interfacial phenomena, or other ion transport parameters that affect the salt diffusion coefficient determined by ac impedance.[37] The restricted-diffusion method measures the mutual diffusion coefficient which is directly applicable in cell modeling, instead of the self-diffusion coefficient measured by techniques such as NMR.[38, 39] The restricted diffusion technique has theoretical advantages for optical cells because the concentration can be tracked at arbitrary points along the length of the diffusive pathway.[35, 40-42] In this study, however, the concentration profile was determined indirectly by measuring the open-circuit voltage. This approach has been successfully applied to a number of electrolyte systems, including molten polysulfides[36] and polymer electrolytes.[27, 28, 43-47] Several studies have applied the restricted-diffusion method to microstructured media, including a separator-containing system,[46] a porous lithium-ion anode,[48] and a system containing two separators and a microporous cathode.[49] Thorat et al.[49] used the restricted-diffusion method to estimate tortuosity in the separators and the cathode. Onishi and Newman established the importance of contributions from polymer relaxation modes on the measured conductivity of Nafion.[50] The present paper is the first report wherein the presence of a distribution of relaxation modes is established by using a Laplace inversion algorithm to analyze the restricted-diffusion data.

2.2 Methods

The molar ratio of lithium to ethylene oxide moieties, *r*, was held fixed at 0.085 for all SEO and PEO(27) samples. This is the salt concentration that maximizes ionic conductivity in SEO electrolytes.[19]

Electrolyte morphologies and domain spacings of the SEO/LiTFSI mixtures were determined using small-angle X-ray scattering (SAXS) performed at the Advanced Light Source (ALS) at Lawrence Berkeley National Laboratory in Berkeley, CA, using beamline 7.3.3. SAXS samples were prepared by hot-pressing dry polymer electrolytes into 200 μm thick spacers (Garolite G-10) at 90 °C, and the spacers were then assembled into custom-built, airtight aluminum sample holders with Kapton windows. Samples were annealed in the sample holders at 120 °C for at least four days and were then transported immediately from the glovebox to the beamline for SAXS acquisition. All SAXS analysis was performed using the Nika program written for Igor Pro.[51] Beam-centers and sample-to-detector distances were calibrated using silver behenate, a common SAXS standard with a primary scattering peak at 58.38 nm. The measured two-dimensional scattering data were averaged azimuthally to obtain intensity versus magnitude of the scattering wave vector *q,* as described in Chapter 1. The SAXS data from all of the samples were obtained at 120 °C (above the glass-transition temperature of PS). A few samples were examined at both 90 and 120 °C, and no change in morphology was observed. All of the profiles except SEO(240 – 269) contained a primary peak at scattering vector q^*. This enables determination of the domain spacing, $d = 2\pi/q^*$. The presence of higher-order SAXS peaks confirmed that the morphologies of all samples are lamellar, as given in Table 3. The domain spacing of SEO(240 – 269) could not be determined via SAXS because its primary scattering peak was outside of the detectable range at the maximum sample-to-detector distance of 4.4 meters. However, resonant soft X-ray scattering (RSOXS) was used to determine that SEO(240 – 269) exhibits a lamellar morphology with a domain spacing of 241 nm, as described in Chapter 4[52]. The domain spacings of the SEO samples except SEO(240 – 269) given in Table 1 were fit to a power-law of the form $d = A_0 M_{\mathrm{SEO}}^x$, which gave $A_0 = 3.08 \pm 0.95$ nm (mol/kg)^x and $x = 0.71 \pm 0.07$. The exponent x is in reasonable agreement with theoretical predictions of block copolymers in the strong-segregation limit which give $x = 0.67$.[53, 54]

Table 3 – Characteristics of polymers used in Chapter 2

Polymer	M_{PS} (kg/mol)	M_{PEO} (kg/mol)	ϕ_{PEO}	PDI_{polymer}	Domain spacing (nm)[a]	Morphology[a]
PEO(27)	–	27	1	1.10	–	–
SEO(6 – 7)	6	7	0.52	1.02	21.7	Lamellar
SEO(16 – 16)	16	16	0.48	1.09	34.9	Lamellar
SEO(36 – 24)	36	24	0.38	1.10	51.8	Lamellar
SEO(37 – 25)	37	25	0.39	1.04	52.7	Lamellar
SEO(40 – 54)	40	54	0.55	1.20	82.4	Lamellar
SEO(53 – 68)	53	68	0.54	1.05	101.9	Lamellar
SEO(74 – 98)	74	98	0.55	1.14	113.7	Lamellar
SEO(240 – 269)	240	269	0.50	1.26	241[b]	Lamellar[b]

[a]Parameters correspond to SEO/LiTFSI mixtures at 90 °C with $r = 0.085$.

[b]The morphology of SEO(240-269) was determined via RSOXS for a spun-cast film with $r = 0.085$ at room temperature.

SEO samples for electrochemical measurements were prepared by first placing freeze-dried electrolytes in Garolite G-10 molds and hand-pressing them into pellets. Pellets were then placed into G-10 spacers with inner diameters of 3.86 mm and with thicknesses ranging from 125 µm to 800 µm. Samples were hot-pressed between polytetrafluoroethylene sheets for 1 second at 90 °C with a force of 1000 $\mathrm{lb_f}$ (Carver Mini-C) to create uniform, nonporous films. The sample geometry was not constrained during pressing; thus the pressure on the sample was not well-defined. Based on the spacer area, the pressure on the sample is 3800 bar. PEO(27) samples were prepared by melting the electrolyte in a vial on a hot plate and then using a spatula to place the molten PEO(27) into a spacer.

Restricted-diffusion samples were prepared by assembling the polymer electrolytes with one piece of 150 µm-thick lithium metal foil (FMC Lithium) on each side, and with one 18 µm thick nickel shim on each side. Lithium metal shims were cut from the foil with a 0.635 cm (0.25 inch) gasket punch and then carefully removed with tweezers. Ni/Li/polymer/Li/Ni samples were then assembled in stainless-steel Swagelok cells. The Ni foils prevent lithium corrosion at the Swagelok electrode surfaces. The bottom electrodes in the Swagelok cells were fixed, whereas the top electrodes consisted of spring-loaded plungers that maintained pressure on the polymer samples. A home-made temperature controlled box was used to house the cells during the electrochemical experiments, and a proportional-integral-derivative-feedback (PID) temperature controller was used to regulate the box temperature. The sample temperatures were calibrated prior to experimentation by placing wire thermocouples into the sample position of each Swagelok cell. Temperatures of all samples corresponded to the control thermocouple better than ± 1 °C from room temperature up to 90 °C, the maximum temperature used in this study.

Restricted-diffusion measurements were performed on a galvanostat (Solartron 1470 E) with a sampling rate of six points/min with voltage resolution on the order of 5 μV. Samples were polarized at a current density of 0.13 mA/cm^2 for 10 minutes, and then allowed to relax for 12 hours with the current turned off. The open-circuit voltage, V, was measured as a function of time, t. This polarization and relaxation process was repeated with the negative applied current density to complete a cycle. Further relaxation profiles were collected at other current/charging time pairings for a total of four or five relaxation cycles. Current densities ranging from 0.043 to 0.171 mA/cm^2 were used. Total sample thicknesses were measured at the end of each experiment using a micrometer, and the lithium and nickel foil thicknesses were subtracted to determine the electrolyte thickness for each polymer electrolyte.

In previous studies,[35, 36] it is generally assumed that after a sufficiently long period (which we specify below) the open-circuit voltage relaxation, $V(t)$, decays with a single exponential as in Equation 4

$$V(t) = k_0 + k_1 e^{-\Gamma t} \qquad [4]$$

where k_0 is a baseline offset and k_1 is a constant related to the initial concentration profile. The k_0 term arises from the potential generated by uncontrolled effects such as small but finite thermal gradients across the electrodes (the "thermocouple effect") and differences in electrochemical potential of the electrodes due to handling during their preparation. In this study k_0 values are typically in the range of 0.1 to 0.5 mV. The parameter Γ is of primary interest as it relates to the diffusion coefficient, D

$$D = \frac{\Gamma L^2}{\pi^2} \qquad [5]$$

where L is the thickness of the electrolyte. For simple unstructured electrolytes, Equation 4 is predicted to apply regardless of the initial salt concentration profile when the dimensionless quantity $\alpha = Dt/L^2 > 0.05$.[36]

The salt diffusion coefficient measured by the restricted diffusion method is obtained in the presence of concentration gradients. It is related to the salt self-diffusion coefficient, $\mathfrak{D}$, by Equation 6[34]

$$D = \mathfrak{D}\frac{\mathrm{c_T}}{\mathrm{c_0}}\left(1 + \frac{\mathrm{dln}(\gamma_\pm)}{\mathrm{dln(m)}}\right) \qquad [6]$$

where c_T is the total molar salt concentration, c_0 is the molar concentration of the solvent in the solution, $\gamma_\pm$ is the mean molal activity coefficient, and m is the molality of the solution (moles of electrolyte per kilogram of solvent). In the case of a salt with univalent ions, $\mathfrak{D}$ is given by

$$\mathfrak{D} = \frac{2\mathfrak{D}_{0+}\mathfrak{D}_{0-}}{\mathfrak{D}_{0+}+\mathfrak{D}_{0-}} \quad [7]$$

where $\mathfrak{D}_{0+}$ and $\mathfrak{D}_{0-}$ are the self-diffusion coefficients for the cation and anion, respectively. Equations 4 and 5 have been applied only to homogeneous binary electrolytes.[27, 43-47] In the case of block-copolymer electrolytes, it is likely that these equations are only approximately valid with the assumption that the "solvent" is the block that solvates the salt. The validity of these equations can be tested by combining restricted-diffusion measurements with pulsed field gradient NMR measurements. We will report on this in future publications. In this Chapter, we focus on the explicit measurement of *D* without attempting to determine how it is related to self-diffusion coefficients of the ions.

We found that *V*(*t*) for homopolymer electrolytes was well-described by Equation 4 when $\alpha > 0.05$. However *V*(*t*) for block-copolymer electrolytes could not be described by Equation 4. In other words, the restricted diffusion experiments indicated the presence of multiple relaxation modes. This is expressed by an extension of Equation 4

$$V(t) = k_0 + \int_a^b e^{-\Gamma_c t} f(\Gamma_c) d\Gamma_c \quad [8]$$

where $f(\Gamma_c)$ is a continuous distribution of time constants, and *a* and *b* are chosen integration limits. The term $[V(t) - k_0]$ is the Laplace transform of $f(\Gamma_c)$. Equation 4 may be considered as a special case of Equation 8 in which $f(\Gamma_c) = k_1\delta(\Gamma_c - \Gamma)$, where δ is the Dirac delta function. In practice, one solves Equation 8 numerically and the integral in Equation 8 is thus replaced by a sum of discrete exponential terms with time constants $1/\Gamma_i$ and prefactors $f(\Gamma_i)$. We use the subscripts c and i to distinguish between continuous and discrete approximations of $f(\Gamma)$.

We focus on two averages of Γ, defined in Equations 9 and 10

$$\Gamma_{1,\text{avg}} = \frac{1^{\text{st}} \text{ moment}}{0^{\text{th}} \text{ moment}} = \frac{\sum_{i=a}^{b} f(\Gamma_i)\Gamma_i}{\sum_{i=a}^{b} f(\Gamma_i)} \quad [9]$$

$$\Gamma_{2,\text{avg}} = \frac{2^{\text{nd}} \text{ moment}}{1^{\text{st}} \text{ moment}} = \frac{\sum_{i=a}^{b} f(\Gamma_i)\Gamma_i^2}{\sum_{i=a}^{b} f(\Gamma_i)\Gamma_i} \quad [10]$$

These averages are recast in terms of two parameters that characterize the salt diffusion processes: an average diffusion coefficient, $D_{1,\text{avg}}$, defined in Equation 11

$$D_{1,\text{avg}} = \frac{\Gamma_{1,\text{avg}} L^2}{\pi^2} \quad [11]$$

and a polydispersity index, $PDI_{\text{diffusion}}$, defined in Equation 12.

$$PDI_{\text{diffusion}} = \frac{\Gamma_{2,\text{avg}}}{\Gamma_{1,\text{avg}}} \qquad [12]$$

$PDI_{\text{diffusion}}$ is a measure of the width of the $f(\Gamma_i)$ distribution. In simple liquids, we expect $f(\Gamma_i)$ to be a delta function, in which case $PDI_{\text{diffusion}} = 1.0$. Our framework is similar to that used to characterize the molecular-weight distribution of polymers in Table 3. The term D_{avg} is defined as the average of a collection of $D_{1,\text{avg}}$ values obtained for multiple relaxation profiles obtained from a set of samples with a given composition.

It is useful to use D_i, defined as $D_i = \Gamma_i L^2/\pi^2$, as the independent variable, rather than Γ_i, particularly when examining distribution functions obtained from cells with different values of L. Furthermore, $f(D_i)$ curves (units of V) are divided by their 0th moments (units of V) to give normalized distributions, $f_n(D_i)$ (dimensionless). The 0th moment of $f(D_i)$ equals $V(t = 0)$, and the 0th moment of each $f_n(D_i)$ distribution is unity. We define $t = 0$ as the start of the fitting region of the measured relaxation profile ($\alpha > 0.05$).

To perform the analysis described above (Equations 8 through 12), the experimental data files of the open-circuit voltage, $V(t)$, versus time, t, were loaded into Igor Pro. Laplace inversion fits were performed using the Contin program, which is freely available on the Internet.[55] The Laplace inversion integral is an ill-posed problem, and in the presence of finite instrument noise, fit solutions can contain arbitrarily large deviations from the true distribution function.[56] Contin imposes the following physically-reasoned constraints in order to achieve a well-defined, reproducible fit to the data: 1) all values of Γ_i are positive, 2) $f(\Gamma_i)$ is smooth and 3) the most parsimonious distribution is chosen – that is, the simplest fit which adequately describes the decay profile.[57] Our custom Igor Pro code saved data from a selected portion of the $V(t)$ data set into a Fortran-readable text format and then executed Contin. The Contin program then generated a text file output containing the fit solutions which were subsequently read back into Igor Pro. The first 10 to 15 minutes of each $V(t)$ were cropped in order to remove the contributions of higher-order terms in accordance with the constraint that $\alpha > 0.05$.[36] A logarithmically-spaced grid of 500 points was used to represent $f(\Gamma_i)$ with integration limits $a = 10^{-6}$ 1/s and $b = 10^{-2}$ 1/s.

Contin attempts to fit the experimental data across the entire input range (from tens of mV to μV), resulting in sensitivity to experimental noise, particularly to fluctuations in the baseline. The 12 h data acquisition times were necessary to obtain accurate values of the baseline k_0 [see Equation 8]. This important fact has been recognized in a previous publication.[36] The inclusion of long periods of baseline data results in the appearance of small-amplitude peaks in $f(\Gamma_i)$ at both high and low values of Γ_i. We do not believe that these peaks have any physical significance; thus $\Gamma_{1,\text{avg}}$ and $PDI_{\text{diffusion}}$ values are calculated from the distribution in the range 1×10^{-8} cm^2/s $< D_i < 5 \times 10^{-7}$ cm^2/s.

A global Laplace inversion fitting procedure known as GRIP[58] was also used to analyze decay profiles. The D_{avg} values obtained from GRIP deviated less than 1% from those obtained by Contin. Global fitting algorithms such as GRIP should be considered in further detail, but the similarity to Contin results and the computational expense of the GRIP algorithm led us to focus on Contin for this report.

2.3 Results and Discussion

In Figure 4 we show a typical voltage-decay curve and the corresponding Contin fit for a PEO(27) sample, and Figure 5 shows the corresponding distribution function $f_n(D_i)$ obtained from the fit in Figure 4. The y-axis in Figure 4 is $-\ln[V(t)-k_0]$ wherein the baseline k_0 is obtained from the Contin fit. The distribution in Figure 5 is relatively sharp with $PDI_{\text{diffusion}} = 1.002$, indicating that $V(t)$ for PEO(27) is well-described by a single exponential. In fact, the profile in Figure 4 can be fit with Equation 4 across the same region to obtain close fits to the experimental data across all time scales and similar values of k_0 as that obtained from the Contin fit. $D_{1,\text{avg}}$ for the distribution in Figure 5 is 1.20×10^{-7} cm^2/s, which compares favorably to the value of 1.14×10^{-7} cm^2/s derived from a single-exponential fit to the same region of $V(t)$. D_{avg} for PEO(27) at 90 °C and $r = 0.085$, as calculated from Contin fits to 28 relaxation curves from 7 different samples, is $(1.23 \pm 0.19) \times 10^{-7}$ cm^2/s.

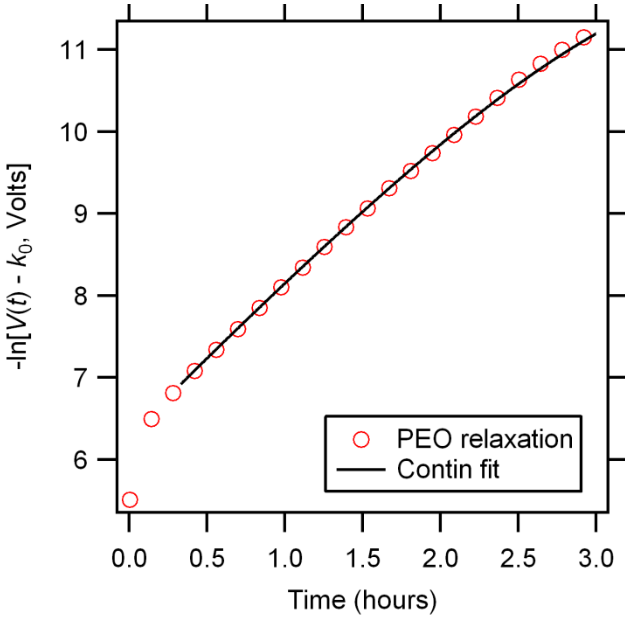


Figure 4 – Relaxation curve for PEO(27) at *r* = 0.085 and 90 °C.

Typical trace of baseline subtracted open-circuit voltage, V-k_0, versus time, t, after turning off the current in a symmetric lithium/homopolymer electrolyte/lithium cell on a semi-log plot for PEO(27) (open circles, plotted sparsely) at 90 °C and $r = 0.085$. The curve represents the Laplace inversion fit using Contin.

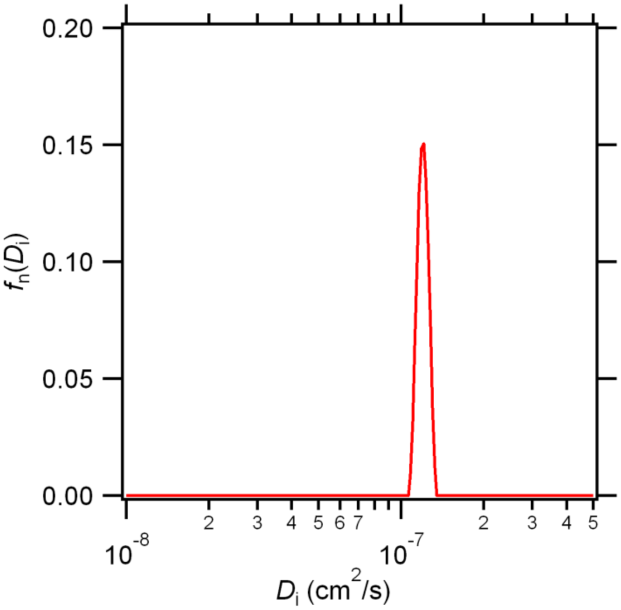


Figure 5 – Normalized distribution function for the PEO(27) relaxation profile in Figure 4. Relaxation modes, $f_n(D_i)$, obtained from the Contin fit as a function of effective diffusion coefficient, D_i. 90 °C and r = 0.085. The independent variable was converted from the raw output of Γ_i to D_i using Equation 5. The distribution is relatively sharp with a PDIdiffusion of 1.002, indicating that the decay profile in Figure 4 is well-described by a single-exponential.

We are aware of two restricted diffusion reports on PEO/LiTFSI. Geiculescu et al.[47] report a value of (4.2 ± 0.8) × 10^{-8} cm^2/s at 90 °C and r = 0.033, which is similar to the value of 4.65 × 10^{-8} cm^2/s reported by Edman and coworkers[28] at 85 °C and r = 0.083. Edman and coworkers[38] used NMR to study diffusion in PEO/LiTFSI and report values for $\mathfrak{D}_{0+}$ and $\mathfrak{D}_{0-}$ at 85 °C and $r \approx 0.085$ of approximately 2 × 10^{-7} cm^2/s and 4 × 10^{-8} cm^2/s, respectively, resulting in $\mathfrak{D} \approx 6.8 \times 10^{-8}$ cm^2/s according to Equation 7. The information in the right side of Equation 6 is unknown for PEO/LiTFSI, so D_{avg} cannot be computed from the calculated value of D. We do not offer any explanation for the deviations between our measurements and those reported in the literature except to note that these deviations are larger than those expected due to differences in temperature and sample composition.

Only some of the PEO(27) distribution functions are sharp and monodisperse, although all of the PEO(27) relaxation profiles can be fit adequately with Equation 4. We noted that $V(t)$ curves that contained large amounts of baseline noise had significantly larger $PDI_{\text{diffusion}}$ values. Of the twenty eight PEO(27) Contin fits, ten have $PDI_{\text{diffusion}}$ values less than 1.10. The average $PDI_{\text{diffusion}}$ value for all PEO(27) samples is 1.16 ± 0.13.

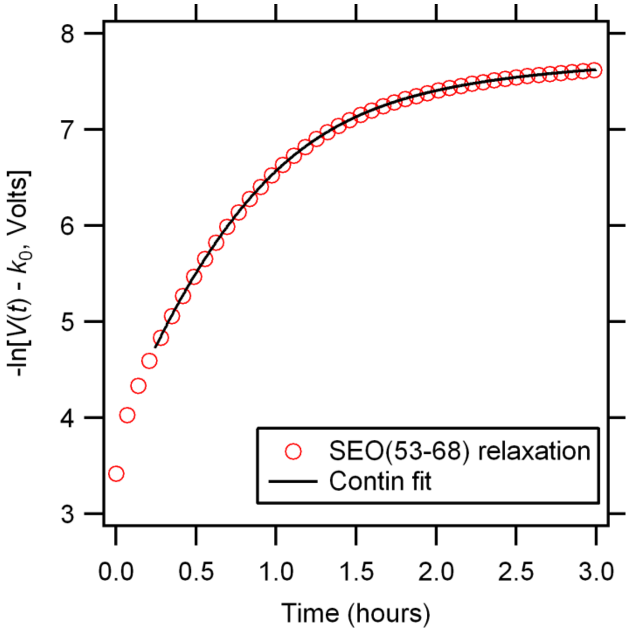


Figure 6 – Example relaxation profile of SEO(53-68).

This is a typical trace of baseline subtracted open-circuit voltage, V-k_0, versus time, t, after turning off the current in a symmetric lithium | block-copolymer electrolyte | lithium cell on a semi-log plot for SEO(53-68) (open circles, plotted sparsely) at 90 ° and r = 0.085. The curve represents the Laplace inversion fit using Contin.

In Figure 6 we show a typical –ln[$V(t)$-k_0] versus time curve obtained from SEO(53-68) for the first 3 hours, wherein the baseline k_0 is obtained from the Contin fit. It is clear that the –ln[$V(t)$-k_0] vs. time plot is nonlinear over a very large time window. In contrast, –ln[$V(t)$-k_0] vs. time for the PEO(27) samples were approximately linear after α = 0.05 (Figure 4). Estimates of the diffusion coefficient in SEO(53-68), determined below, give α = 0.05 at t = 15 minutes. The fact that the data in Figure 6 are non-linear on much larger time scales indicates that the non-linearity of data obtained from block-copolymer electrolytes arises from factors that are fundamentally different from those affecting homogeneous electrolytes. The Contin fit shown in Figure 6 matches $V(t)$ from the beginning of the fit region through the end of the relaxation measurement, whereas a single-exponential fit to the same region fails to capture the shape of the curve across all time scales and fails to give a realistic estimate of k_0. It is worth noting that the range of [$V(t)$-k_0] in Figure 6 spans a factor of e^5.

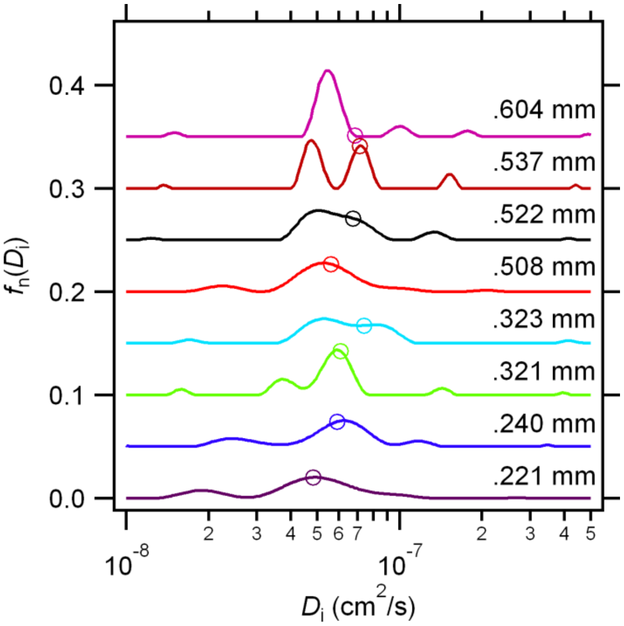


Figure 7 – Normalized distribution functions obtained from Contin fits.
Each distribution is plotted as a function of effective diffusion coefficient, D_i. Data obtained from eight SEO(53-68) (r = 0.085, 90 °C) samples with different thicknesses are shown. Curves are vertically offset by 0.05 units for clarity. $D_{1,avg}$ is indicated by an open circle for each curve. $PDI_{diffusion}$ values for the curves, from top to bottom, are 1.422, 1.197, 1.158, 1.218, 1.099, 1.092, 1.163, and 1.179, respectively.

Figure 7 shows Contin distribution functions $f_n(D_i)$ for SEO(53-68) across a wide range of sample thicknesses. The $D_{1,avg}$ values obtained from the Contin distributions are also shown in Figure 7 (open circles). The individual distribution functions in Figure 7 vary widely while the $D_{1,avg}$ values do not. The values of $PDI_{diffusion}$ for the distribution functions in Figure 7 vary from 1.092 to 1.422. It is known that Contin is effective in resolving different relaxation modes only when the time scales of the modes differ by orders of magnitude.[56, 57] This is not the case for the $f_n(D_i)$ functions in Figure 7. We thus do not ascribe any particular significance to the locations of the peaks seen in Figure 7. It is clear, however, that salt diffusion in SEO(53-68) at 90 °C is characterized by a broad distribution function with a $PDI_{diffusion}$ of about 1.36.

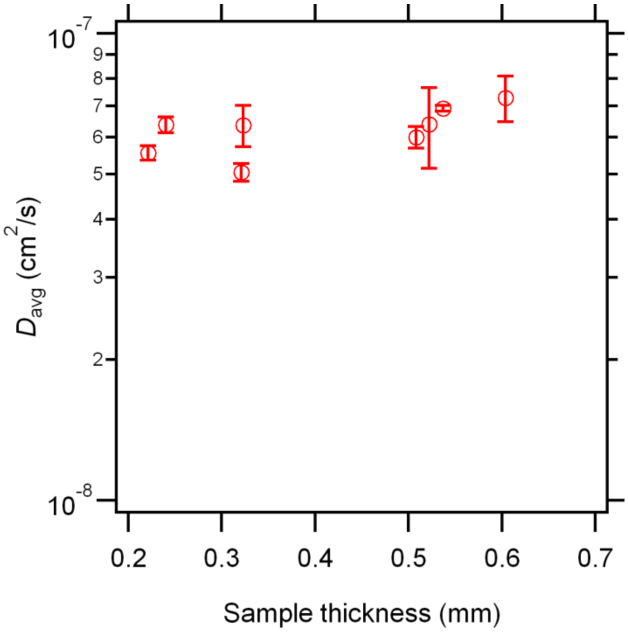


Figure 8 – A plot of the average diffusion coefficient, D_{avg}, versus sample thickness.
All data is for the SEO(53-68) (r = 0.085, 90 °C) samples. D_{avg} is independent of sample thickness – an important null test.

Figure 8 shows D_{avg} as a function of sample thickness for SEO(53-68). The error bars were calculated from the standard deviation of $D_{1,avg}$ values taken for four charge/discharge relaxation cycles per sample. The fact that D_{avg} is independent of sample thickness is consistent with the notion that the process being probed by our experiments is diffusive.

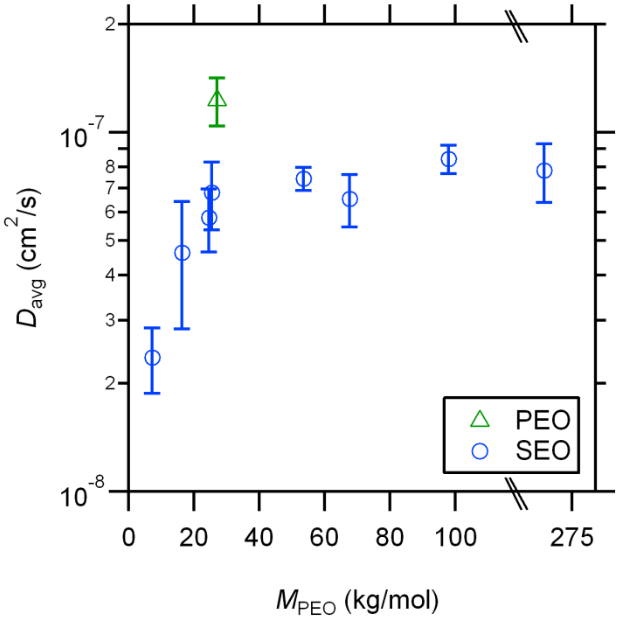


Figure 9 – D_{avg} for a series of nearly symmetric SEO copolymers and PEO(27) homopolymer.
All data is at r = 0.085 and 90 °C and is plotted as a function of the molecular weight of the PEO chains, M_{PEO} (split axis). D_{avg} for SEO copolymers increases with M_{PEO} and plateaus in the vicinity of 8 x 10^{-8} cm^2/s in the vicinity of M_{PEO} = 50 kg/mol and approaches D_{avg} for PEO(27) homopolymer.

Data obtained from all of the SEO-based electrolytes was qualitatively similar to those reported in Figures 4 and 5 in that their respective relaxations could not be fit with Equation 4 and that their Contin distributions were generally very broad. These results are summarized in Figure 9 which shows D_{avg} as a function of M_{PEO}. Each datum in Figure 9 is based on twelve to seventy two relaxation curves obtained from three to twelve samples for a given M_{PEO}. The error bars represent the standard deviations of D_{avg}. The main trend seen in Figure 9 is that D_{avg} increases with M_{PEO} and plateaus in the vicinity of 8×10^{-8} cm^2/s when M_{PEO} exceeds 50 kg/mol. This trend is very similar to the conductivity versus M_{PEO} trend reported by Singh et al.[20] and Panday et al.[19] It is convenient to normalize D_{avg} for SEO samples by D_{avg} for PEO(27), $D_{\mathrm{PEO}} = (1.23 \pm 0.19) \times 10^{-7}$ cm^2/s, to focus on the effect of copolymer morphology on salt diffusion. Our measured value of D_{PEO} is shown in Figure 9. Following our analysis of ionic conductivity given by Equation 3, we define a normalized diffusion coefficient, D_{n}, as

$$D_{\mathrm{n}} = \frac{D_{\mathrm{SEO}}}{D_{\mathrm{PEO}} f} \qquad [13]$$

where D_{SEO} is D_{avg} for the SEO sample of interest and f is the same tortuosity factor introduced in Equation 3. We have used $f = 2/3$ to normalize all of our data. Following previous work[19], we define a normalized conductivity, σ_{N},

$$\sigma_N = \frac{\sigma_{\mathrm{SEO}}}{\sigma_{\mathrm{PEO}} \phi_{\mathrm{PEO}} f} \qquad [14]$$

where σ_{SEO} is the measured conductivity of SEO, σ_{PEO} is the measured conductivity of PEO, and $f = 2/3$. We used the value of 1.87×10^{-3} S/cm from Lascaud et al.[59] for σ_{PEO}. In the absence of tortuosity, i.e. if the sample were a single grain with lamellae oriented perpendicular to the electrodes, we expect the conductivity and diffusion coefficient to be $\sigma_{\mathrm{PEO}}\phi_{\mathrm{PEO}}$ and D_{PEO}, respectively. This is the reason for the major difference between Equations 13 and 14, namely the absence of ϕ_{PEO} in the denominator of Equation 13. $D_{\mathrm{n}} = 1$ and $\sigma_{\mathrm{N}} = 1$ represent ideal cases when lamellar grains are well-connected and the intrinsic transport properties within the PEO nanostructures are the same as those in bulk PEO. Values of D_{n} and σ_{N} less than 1 are attributable to poor connectivity between grains or poor intrinsic transport within the channels. The characteristic grain size in salt-free block copolymers is on the order of 1 to 10 μm,[60, 61] compared to L ranging between 125 to 800 μm.

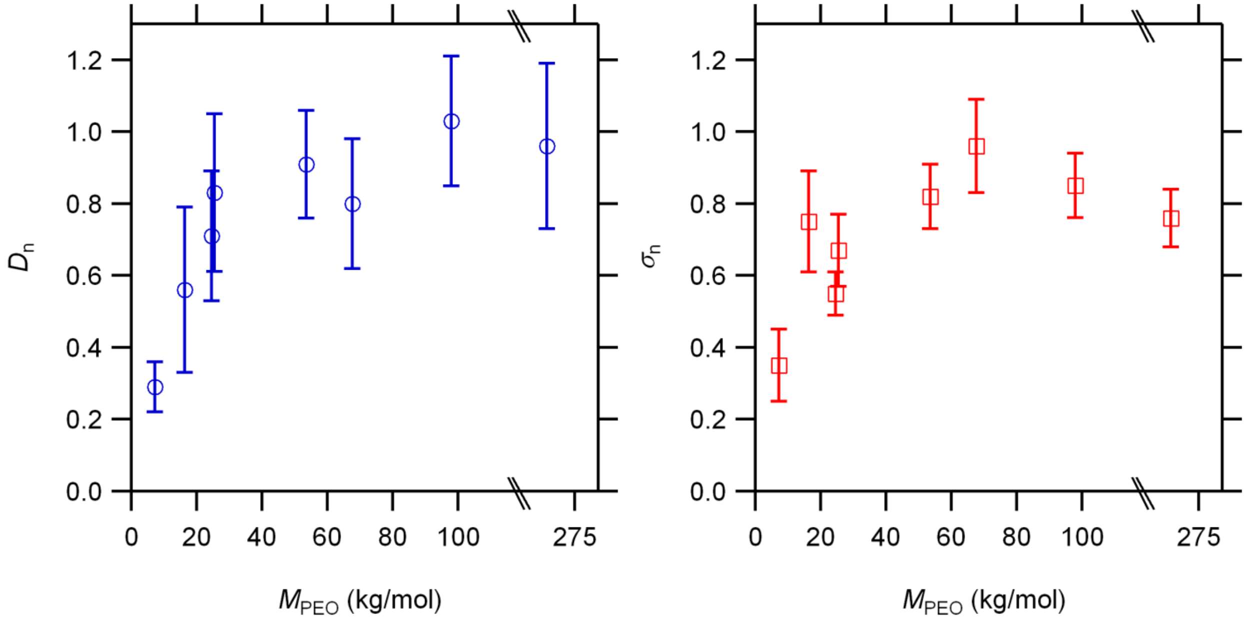


Figure 10 – Comparison of normalized salt diffusion coefficients and conductivities.
(a) Normalized salt diffusion, D_n, and (b) normalized conductivities, σ_n, for a series of nearly symmetric SEO copolymers at $r = 0.085$ and 90 °C as a function of the molecular weight of the PEO block, M_{PEO} (split axis). D_n and σ_n exhibit similar dependences on M_{PEO} – approaching the ideal value of unity at high values of M_{PEO}.

In Figure 10a and Figure 10b we show D_n and σ_n, respectively, as functions of M_{PEO} for the samples listed in Table 3. It is evident that D_n and σ_N both plateau at values around unity when M_{PEO} exceeds 50 kg/mol. σ_N values vary slightly from those reported by Panday et al.[19] due to differences in the sample preparation procedure. Samples in this study were not annealed above 90 °C, whereas samples reported by Panday et al.[19] were annealed at 130 °C prior to measurement. We found that annealing the samples extensively at 130 °C in the presence of two lithium electrodes and then cooling them to 90 °C gave irreproducible relaxation results.

Table 4 – Average diffusion coefficient, D_{avg}, and $PDI_{diffusion}$ values.
All listed values are at 90 °C and r = 0.085.

Polymer	D_{avg} (cm^2/s)	Average $PDI_{diffusion}$
PEO(27)	$(1.23 \pm 0.12) \times 10^{-7}$	1.16 ± 0.13
SEO(6 – 7)	$(2.36 \pm 0.49) \times 10^{-8}$	1.67 ± 0.32
SEO(16 – 16)	$(4.62 \pm 0.18) \times 10^{-8}$	1.59 ± 0.26
SEO(36 – 24)	$(5.79 \pm 1.2) \times 10^{-8}$	1.41 ± 0.18
SEO(37 – 25)	$(6.80 \pm 1.5) \times 10^{-8}$	1.64 ± 0.35
SEO(40 – 54)	$(7.44 \pm 0.6) \times 10^{-8}$	1.31 ± 0.15
SEO(53 – 68)	$(6.54 \pm 1.1) \times 10^{-8}$	1.36 ± 0.27
SEO(74 – 98)	$(8.45 \pm 0.8) \times 10^{-8}$	1.38 ± 0.21
SEO(240 – 269)	$(7.83 \pm 1.5) \times 10^{-8}$	1.35 ± 0.26

Our diffusion measurements are summarized in Table 4 where D_{avg} and $PDI_{diffusion}$ of each of the polymers listed in Table 3 are reported. The $PDI_{diffusion}$ values of salt diffusion distribution functions of block-copolymer electrolytes are significantly larger than that of homopolymer electrolytes. A possible explanation for this observation is illustrated in Figure 11 where we show a set of randomly oriented lamellar domains through which the salt molecules diffuse. It is clear that such a structure will lead to a distribution of path lengths, which in turn will lead to a distribution of Γ. Such a distribution is not expected in homogeneous electrolytes. Further work is needed to establish the underpinnings of the observed values of $PDI_{\mathrm{diffusion}}$.

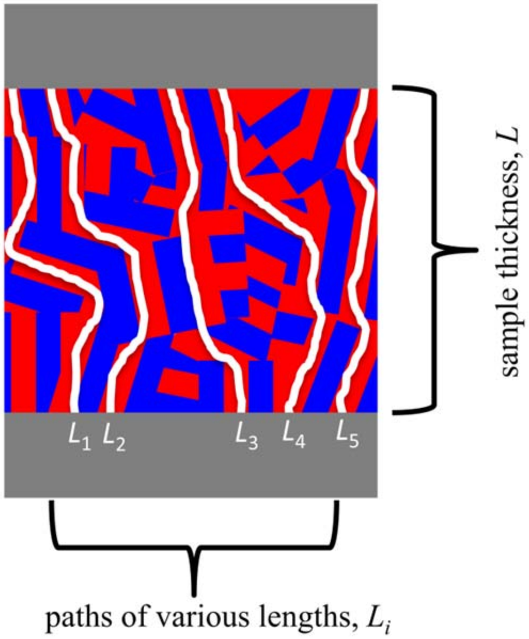

sample thickness, L
L_1 L_2 L_3 L_4 L_5
paths of various lengths, L_i

Figure 11 – Illustration of the proposed source of large $PDI_{\mathrm{diffusion}}$ values in SEO electrolytes. Lamellar block-copolymer grains (PEO lamellae are red and the PS lamellae are blue) are drawn between lithium metal electrodes. The distance between the electrodes is the measured sample thickness (L). Paths of various lengths (L_{i}) arise due to the random orientation of the lamellae. The diagram is not to scale; in reality, individual grains are much smaller than the sample thickness.

2.4 Conclusions

The salt diffusion coefficient, D_{avg}, of a series of lamellar SEO/LiTFSI electrolytes was measured using the restricted-diffusion technique in combination with a Laplace inversion fitting analysis. We found that D_{avg} increases as the molecular weight of the PEO block, $M_{\mathrm{PEO,}}$ increases, and reaches a plateau when M_{PEO} exceeds 50 kg/mol. This is similar to previously reported trends that indicate that the ionic conductivity of block-copolymer electrolytes, σ, also increases with increasing M_{PEO} and reaches a plateau when M_{PEO} exceeds 50 kg/mol. Equations 13 and 14 provide a simple framework for quantifying the effect of morphology on ion transport. The values of the normalized diffusion coefficient and conductivity based on Equations 13 and 14, D_{n} and σ_{n}, are unity in the high-molecular-weight limit. The diffusion of salt in block copolymers is characterized by a broad distribution function with a polydispersity index, $PDI_{\mathrm{diffusion}}$, which is significantly higher than that of homogeneous electrolytes. The broad distribution functions that we have obtained may be applicable to diffusion in other microstructured membranes such as those used in gas and liquid separation.[62, 63] The ability of potentiostats to resolve small electrochemical potential gradients arising from small concentration gradients, however, provides a unique avenue for quantifying mass transport in complex systems.

Chapter 3 – Conductivity of Block-Copolymer Electrolytes with Minor Conducting Phase

ABSTRACT

Conductivities of asymmetric block-copolymer electrolytes with minor-phase-conductive-cylindrical morphologies were measured as a function of molecular weight. Un-annealed samples exhibited conductivities above 1×10^{-4} S/cm above 100 °C, but exhibited large, irreversible conductivity drops upon annealing past 110 °C. This annealing effect was also observed for lamellar samples, for which large, irreversible conductivity drops were also observed. Conductivities of as-prepared lamellar samples exhibited weak dependence on molecular weight, but those of annealed samples exhibited a strong molecular-weight dependence that matched previous observations. These results are explained in the context of inter-grain connectivity of the conducting phase. We propose that the onset of long-range structural order, as is expected to develop during annealing processes, may result in disconnection of the conductive phase across grain boundaries.

3.1 Introduction

This work is part of a series of studies on the conductivity of block-copolymer electrolytes based on polystyrene-*b*-poly(ethylene oxide) (SEO) and lithium bis(trifluoromethanesulfone) imide (LiTFSI) salt. Previous conductivity studies only reported conductivity results taken during cooling runs after heating samples to 120 °C, since that was found to give the most reproducible results.[9, 19, 20, 24, 30] In addition, previous studies were restricted to nearly symmetric SEO copolymers that formed lamellar morphologies. In this work, we report on the conductivity of asymmetric SEO copolymers having poly(ethylene oxide) (PEO) volume fractions (ϕ_{PEO}) of 0.28 to 0.30. At these compositions, the SEO morphologies consist of hexagonally-packed, ionically-conductive PEO/LiTFSI domains with a cylindrical shape surrounded by a continuous, insulative polystyrene (PS) matrix. The mechanical properties are dominated by those of PS, and it was unclear whether the PEO domains would form continuously connected pathways across bulk sample distances of order 100 μm. Previous work on analogous systems found extremely poor conductivities in such highly asymmetric block-copolymer electrolytes.[64-67] It has been suggested that bulk alignment of the microstructure is required in order to achieve percolation of the conductive phase across bulk length-scales.[64, 65] In this work we report that conductivities above 10^{-4} S/cm are attainable at 90 °C in unaligned, asymmetric SEO/LiTFSI samples composed primarily of the insulating PS phase. Furthermore, we found that processing history plays an important role in the conductivities of all SEO samples. Asymmetric SEOs and low-molecular-weight lamellar SEOs exhibited the highest conductivities when prepared at room temperature, and experienced large conductivity drops after heating above the glass-transition temperature (T_g) of PS. We attribute this annealing effect to rearrangements of microstructural motifs such as grain boundaries.

3.2 Methods

SEO block copolymers were synthesized, purified, and characterized as described in Chapter 1 of this work. SEO/LiTFSI electrolyte mixtures were prepared and freeze-dried, also described in Chapter 1. Freeze-dried electrolyte samples were prepared for conductivity measurements by first forming pellets with a hand press, and then pressing the pellets into Garolite G-10 spacers with a 3.86 mm diameter using a pneumatic press (Janesville Tool and Mfg. Co). As-pressed samples of SEO(54-23) were optically clear, whereas those of SEO(247-116) and SEO(352-166) were opaque. After pressing, sample thicknesses were measured with a micrometer (Mitutoyo #500-196-20) directly on the sample surfaces. 300 μm-thick stainless steel electrodes were placed on each side of the samples and taped to the spacer surface with Kapton tape. Aluminum tabs (Showa-Denko) were affixed to the electrodes with Kapton tape, and the entire cell assemblies were placed into rectangular sections of pouch material (Showa-Denko). Each piece of pouch material was cut from a template so that the finished, pouched cells would fit neatly into the heating stage. Each edge of the pouches was sealed on both sides using a vacuum sealer (AudionVac VMS 53) using 3 s sealing pulses and at least 0.5 atm of vacuum pressure to drive the seal. The unit was fully evacuated before sealing the final edge on each sample in order to create a consistent vacuum level. The pressure difference between the pouch interiors and ambient atmosphere compresses the cell assemblies together. All preparation steps were performed inside of an argon glovebox with sub-ppm oxygen and water levels.

The pouched samples were then removed from the glovebox and transferred to the pouch cell heating stage. The heating stage consists of an aluminum plate with screw-holes tapped for the purpose of affixing insulation. Four samples were placed on each side of the stage, allowing for up to eight samples to be run simultaneously. The back side of each tab was electrically isolated from the heating stage with Kapton tape and the samples were affixed onto the stage by sheets of Teflon insulation. The entire stage was insulated with Teflon. Five 100 W miniature cartridge heaters (Watlow FireRod) were potted inside the heating stage and a K-type thermocouple with fluoropolymer insulation and a coated tip was placed in the center of the stage. A resettable thermal switch with an actuating temperature of 225 °C was affixed to the stage and served as a failsafe. The thermal switch was connected to a relay that connected the heating rods to the power output of a temperature controller. The thermal switch would open if actuated, which in turn would open the relay and prohibit power from reaching the heating rods. Temperature control was provided by a feedback temperature controller (Watlow EZ-Zone PM) that was operated by a custom control program written in LabVIEW. The LabVIEW program was used to control and monitor the temperature of the sample stage. The LabVIEW program accepted a temperature profile as input, and would hold the sample stage within 1 °C of the setpoint for at least one hour before conductivity measurements were taken. In practice, the control temperature was within 0.1 °C of the setpoint for at least 45 minutes before conductivity measurements were taken. The sample temperatures were calibrated by placing fine gauge wire thermocouples in the sample position of test cells designed for that purpose. Sample temperatures were found to correspond to the setpoint temperature within 1 °C. Once the stage was stabilized under these conditions, the Lab VIEW program signaled the potentiostat at each step by means of transistor-transistor logic (TTL) signals. A dedicated digital logic signaler (National Instruments USB 6210) under Lab VIEW control was used to send and receive TTL signals. The TTL signaler was connected to the VMP3 via 9-pin D-subminiature (DB-9) connectors. The VMP3 was programmed to wait for a TTL signal, at which point ac impedance sweeps were performed at 10, 50 and 100 mV. At the end of each conductivity run, the VMP3 was programmed to return a TTL signal to the signaler. The Lab VIEW program would run in parallel, waiting for the TTL signal. Once the return signal was received, the program would advance to the next temperature step. This protocol enabled consistent temperature history control across multiple experiments.

Pouch cells were transferred back into a glovebox at the conclusion of each conductivity experiment. The pouches were carefully cut open with scissors so as to leave the sample undisturbed. The stainless steel shims did not adhere to electrolyte samples with the cylindrical morphology and loosely fell away upon disassembly. The thicknesses of these samples were measured directly from the sample surfaces. Since lamellar samples adhered to the shims, the total thicknesses of the cell sandwiches were measured, and the sample thicknesses were determined by subtraction of the shim thicknesses. The final sample thicknesses were typically 20 to 50% lower than those of the initial sample thicknesses. The details of which sample thicknesses were used to calculate sample conductivities are discussed in the Results and Discussion section.

Sample morphologies were determined via X-ray scattering techniques as described in Chapter 1. Small-angle X-ray scattering (SAXS) was used to identify SEO(54-23) as having a cylindrical morphology and a domain spacing of 47.8 nm at a salt concentration of r = 0.085. SEO(247-116) and SEO(352-166) exhibited domain spacings that were well outside of the q-range accessible via SAXS, and their morphologies were instead determined from drop-cast and spun-coat films as described in Chapter 4. They were both confirmed to have cylindrical morphologies at salt concentrations of r = 0.085. SEO(247-116) and SEO(352-166) have domain spacings of 116 and 123 nm, respectively. Several lamellar SEO samples are referenced in this study. Their characteristics can be found in Table 1.

3.3 Results and Discussion

Figure 12 shows a semi-logarithmic plot of σ vs. 1000/T for SEO(352-166) with r = 0.085. The samples were made by pressing freeze-dried electrolyte that were not annealed prior to this experiment. From their initial states, the samples were heated from 30 to 130 °C in 10 °C steps (denoted as “First heating run”). The sample was held at each specified temperature for at least 1 hour. During the first heating run, the conductivity between 30 and 110 °C increased monotonically with increasing temperature, as expected. Between 110 and 130 °C, σ decreases with increasing T. Next, the sample was cooled from 130 to 30 °C in 10 °C steps (denoted as “First cooling run”), during which σ decreases monotonically with T across the entire scan. The heating and cooling runs are then repeated up to a maximum of 120 °C, (denoted as “Second heating run” and “Second cooling run). The average conductivity values from the “First cooling run”, the “Second heating run”, and the “Second cooling run” agree within 0.3 to 1.3% from 40 to 110 °C, and within 5% at 30 °C. The deviation between the “First heating run” and subsequent runs is maximized in the vicinity of 100 °C. It is interesting to note that deviations between the “First heating run” and subsequent runs at 30 °C are within experimental error due to the change in slope of the temperature dependence.

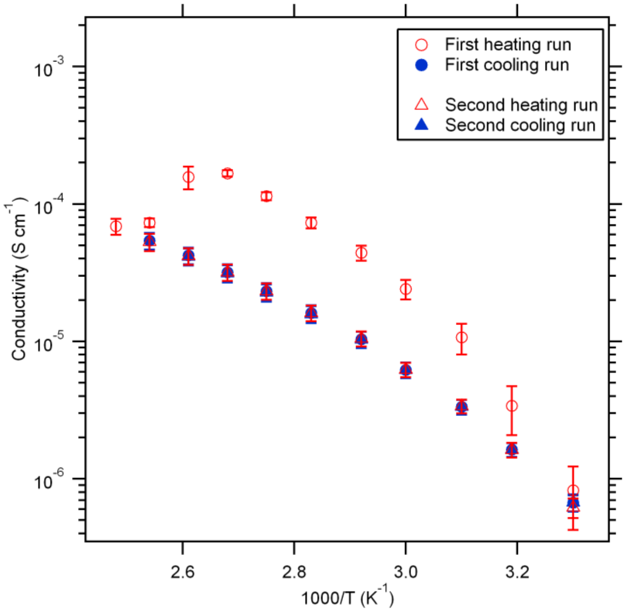


Figure 12 – SEO(352-166) *r* = 0.085 conductivity across temperature scans.
The conductivity drops during the initial heating scan, but remains consistent during subsequent heating and cooling scans.

The sensitivity of the samples to preparation temperature was evaluated by preparing samples of SEO(352-166) at r = 0.085 with a hot press platen temperature of 90 °C. This temperature is below the T_g of the PS block, but it is well above the melting point of PEO/LiTFSI domains, which occurs between 50 and 60 °C, depending on the salt concentration.[68] The results of this test are shown in Figure 13 and are compared with the data from the first heating and cooling runs shown in Figure 12. The results are in close agreement – reaching a maximum value at 100 °C and then experiencing an irreversible conductivity drop with further heating. This test demonstrates that SEO(352-166) conductivities are reproducible for sample preparation temperatures less than the T_g of PS.

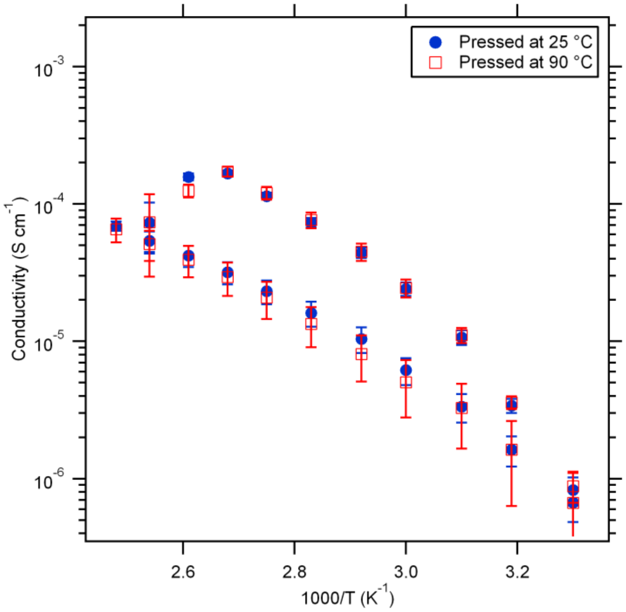


Figure 13 – Conductivity comparison of SEO(352-166) *r* = 0.085 hot pressed at 25 and 90 °C.

In most cases the final sample thicknesses were significantly smaller than the initial sample thicknesses. In previous studies, this aspect was handled by using the final sample thickness for all measurements and reporting only cooling run data. In this study, the initial thickness is used to calculate the conductivities for the first heating run from 30 to 110 °C, after which point the final thicknesses are used for all subsequent measurements. This is a rigorous approach for the initial heating run. The initial sample thicknesses are stable up to 110 °C at which the T_g of PS is reached and the sample deforms under the static pressure of the sample pouch. The error in the conductivity measurements at 110 and 120 °C are significantly higher than those at other temperatures because the sample thicknesses are changing and are not well defined. The sample thicknesses stabilize by 130 °C, as indicated by the reproducibility on subsequent heating and cooling runs. Using the final sample thicknesses for all points would under-estimate the conductivities of the first heating run up to a factor of two.

Statistically significant annealing effects were observed for all samples in this study, including three SEOs with minor-conducting-phase cylindrical morphologies and three SEOs with lamellar morphologies. Arrhenius plots of the conductivities of these samples are shown in Figure 14. Each figure compares the average conductivity of four to seven samples during heating and cooling runs from the range 30 to 130 °C. The error bars represent the standard deviations of the measured conductivities. All data in Figure 14 was acquired from six or seven samples prepared at room temperature with $r = 0.085$.

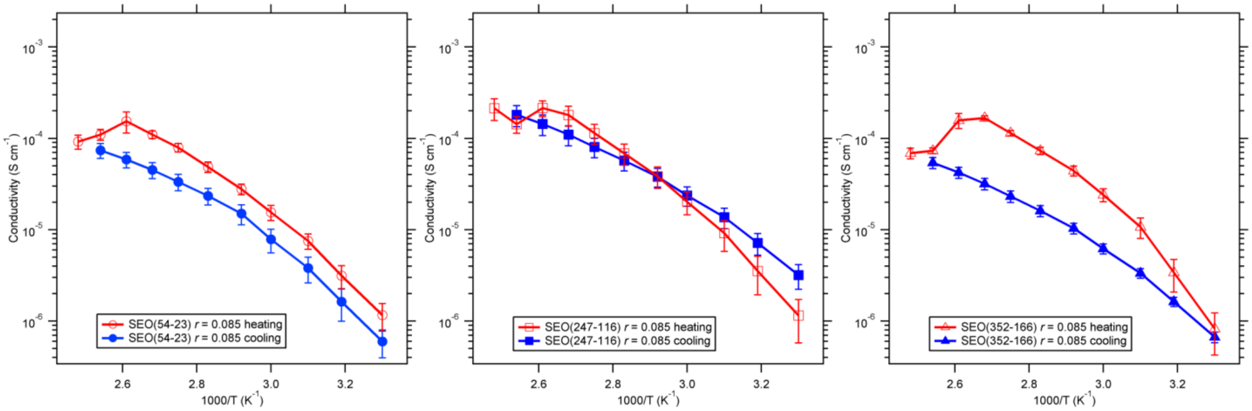


Figure 14 – Conductivity temperature scans of cylindrical SEO samples.
(a) SEO(54-23) shows a maximum conductivity of 1.5 E-4 S/cm at 110 °C. (b) SEO(247-116) shows a maximum conductivity of 2.2 × 10^{-4} S/cm at 110 °C. (c) SEO(352-166) shows a maximum conductivity of 1.7 × 10^{-4} S/cm at 100 °C.

All of the samples in Figure 14 have cylindrical morphologies, and all exhibit conductivity changes upon annealing. SEO(54-23) and SEO(352-166) exhibit irreversible conductivity drops. SEO(247-116) exhibits a significant change in slope such that the conductivities in the range of 80 to 110 °C are smaller on the cooling run, whereas the conductivities in the range of 30 to 60 °C are greater on the cooling run. Deviations from VTF behavior always occur during heating runs with the onset of conductivity drops occurring near the T_g of polystyrene. The block copolymers become mobile above this temperature. This strongly suggests that the conductivity drops are due to changes in the polymer microstructure. The microstructures of as-prepared samples are expected to be highly disorganized, reflecting the structure of the freeze-dried material. Pressing the freeze-dried source material is expected to remove void volume from the samples, but because it is performed at room temperature the glassy PS domains prevent the block copolymers from annealing towards an equilibrium structure. Therefore the as-prepared samples must exhibit kinetically trapped microstructures, and the changes that occur upon annealing must arise due to microstructural rearrangements. The exact microstructural changes that occur are unclear at this time, but the effects they have on SEO/LiTFSI conductivities are significant. If we assume that the only effect of annealing is to rearrange the connectivity between the PEO domains – that is, the PS and PEO domains are always well-defined, there are no mixing/demixing effects between PS and PEO domains, and the intrinsic conductivity within PEO domains is constant – then bulk conductivity changes can be considered to arise from changes in the connectivity of PEO domains. Connectivity between specific polymer domains within block-copolymer microstructures is poorly understood. In an ideal scenario, the microstructure contains no defects and well-defined domains repeat at regular intervals throughout the entire sample volume. Since no efforts are made to impose alignment upon the samples, a more realistic conceptualization is to consider the sample to consist of a large number of domains that exhibit regular, periodic structures on micrometer length-scales, but in which the bulk structure is composed of isotropically distributed grains. Because the as-prepared samples originate from a freeze-dried state, the initial morphologies may not have well-defined grains of any size. If this conception of the initial, disorganized microstructure is correct, then it will be driven towards an isotropically distributed, random grain microstructure upon annealing. In most cases presented in this work the conductivity decreases upon annealing, indicating that having a well-defined microstructure may be detrimental for bulk ion transport.

We posit that the initial, disorganized microstructure has relatively well-connected PEO domains that develop poor inter-grain connectivity during annealing. It is through this lens that we interpret our results.

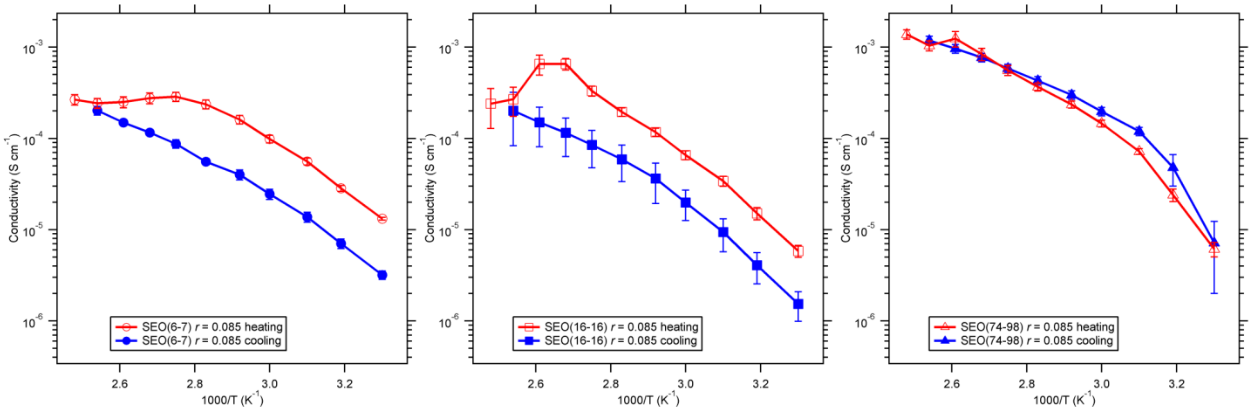


Figure 15 – Conductivity temperature scans of lamellar SEO samples.
(a) SEO(6-7) shows a maximum conductivity of 2.9 × 10^{-4} S/cm at 90 °C. (b) SEO(16-16) shows a maximum conductivity of 6.6 × 10^{-4} S/cm at 100 °C. (c) SEO(74-98) shows a maximum conductivity of 1.4 × 10^{-3} S/cm at 130 °C.

All of the samples in Figure 15 have lamellar morphologies, and all exhibit conductivity changes upon annealing. SEO(6-7) and SEO(16-16) both exhibit significant conductivity drops upon annealing, whereas SEO(74-98) post-annealing conductivities are unaffected in the range 90 and 130 °C and slightly elevated in the range 30 to 80 °C. It is clear that SEO morphology and molecular weight are both important factors in both the pre and post-annealing conductivities. Further insight is gained by evaluating the measured conductivities relative to the theoretically anticipated values if the sample microstructures are isotropically distributed random grains with perfect inter-grain connectivity. Following the arguments in Chapter 2, the theoretical conductivity is given by Equation 14,

$$\sigma = \sigma_{PEO}\phi_{PEO}f \qquad [15]$$

where σ is the predicted conductivity, σ_{PEO} is the intrinsic conductivity of the conducting microphase, ϕ_{PEO} is the volume fraction of the conducting block, and f is a tortuosity factor. Randomly oriented lamellar grains are predicted to have f = 2/3, and randomly oriented cylindrical grains (with an insulating matrix) are predicted to have f = 1/3 [29]. Thus, in addition to having a smaller volume fraction of the conducting phase, samples with confined cylindrical channels are expected to pay an additional penalty for the tortuosity that is naturally required to form inter-grain connections across random grains. Based on the conductivity prediction in Equation 15, a normalized conductivity, σ_{N}, can be calculated according to Equation 16

$$\sigma_N = \frac{\sigma_{\mathrm{SEO}}}{\sigma_{\mathrm{PEO}}\phi_{\mathrm{PEO}}f} \qquad [16]$$

where σ_{SEO} is the measured conductivity of the block copolymer. It is well known that σ_{PEO} decreases with M_{PEO} and plateuas above M_{PEO} = 4 kg/mol.[2, 69] Since M_{PEO} is significantly greater than 4 kg/mol for this study, we presume that σ_{PEO} is constant with a value of 1.12 ± 0.15 × 10^{-3} S/cm based on a PEO homopolymer sample with M_{PEO} = 20 kg/mol with r = 0.085 at 80 °C.[69] If all of our assumptions about the SEO microstructure are valid (isotropically distributed grains, perfect inter-grain connectivity, confined PEO channels having the same intrinsic conductivity as bulk PEO), then we expect $\sigma_{\mathrm{N}} = 1$. Deviations of σ_{N} arise due to a combination of experimental error and the breakdown of one or more of the assumptions underlying Equation 16. We note in passing that σ_{N} values up to 1.5 are possible for lamellar samples and 3.0 for cylindrical samples in the extreme scenario in which the sample is composed of a single grain in which all of the conductive domains are oriented orthogonal to the electrodes.

Figure 16 shows normalized conductivities of cylindrical SEO samples at 80 °C during heating and cooling runs as a function of M_{PEO}. The error bars are calculated using error propagation of σ_{SEO} and σ_{PEO} and assuming no covariance between the terms. The average values of σ_{N} are weakly dependent on M_{PEO} during the heating run, although the trend is not statistically significant. σ_{N} for SEO(352-166) is 0.65 ± 0.11, which is a substantial fraction of the predicted conductivity. This indicates that PEO domains in the as-prepared samples must be relatively well-connected and not overly tortuous throughout the sample. After annealing, however, σ_{N} drops to 0.14 ± 0.03, which indicates that the majority of the PEO domains either lose connectivity or that the conductive pathways become overly tortuous. The cooling-run data show a strong dependence on M_{PEO}, although there is no apparent trend. We conjecture that this result may be due to the specific nature of the initial, kinetically trapped structure and also to the specific nature of its development into a new structure during annealing. If this assumption is correct, then SEO conductivities are expected to depend on other, uncontrolled aspects of the processing history, such as the exact structure of the freeze-dried electrolyte.

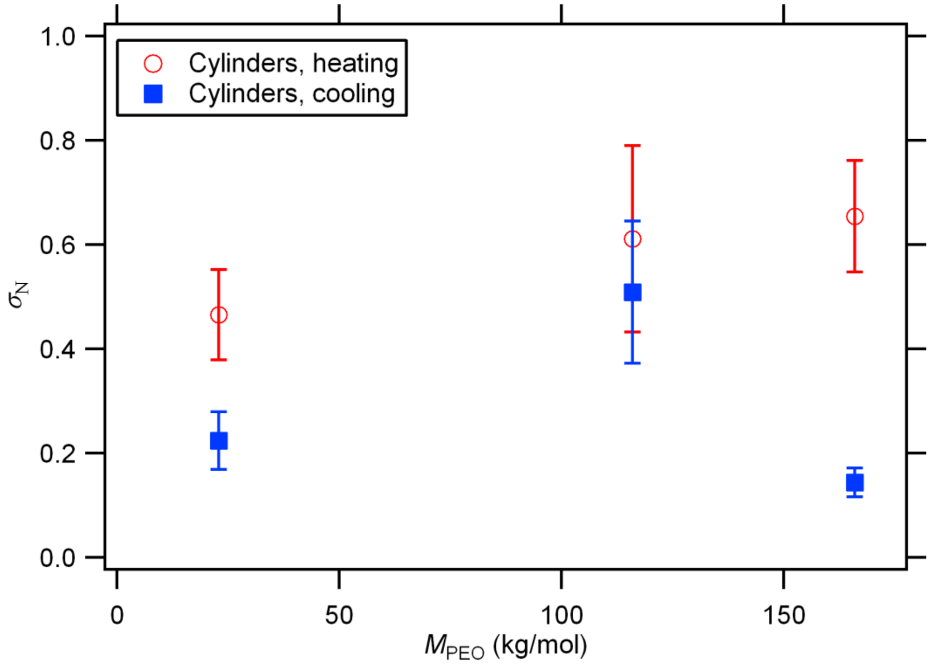


Figure 16 – Normalized conductivities of cylindrical SEO samples.
All data are at 80 °C, and all samples have r = 0.085. As-prepared, average conductivities are 45 to 65 % of the theoretically anticipated maximum. After annealing, average conductivities range from 15 to 50% of the theoretical prediction. σ_{N} is calculated with f = 1/3.

Figure 17 shows normalized conductivities of cylindrical SEO samples at 80 °C during heating and cooling runs as a function of M_{PEO}. Previous reports found that σ_{N} increased with M_{PEO} and reached a plateua in the vicinity of M_{PEO} = 50 kg/mol.[19, 20] σ_{N} values from literature are calculated from σ_{SEO} cooling-run data reported by Panday et al.[19] and σ_{PEO} as described above. The cooling-run data from the present study are in excellent agreement with the literature reports. However, the heating-run data exhibit a weak temperature dependence. This result suggests that PEO-phase connectivities and tortuosities are qualitatively similar before annealing, regardless of M_{PEO}. After annealing, however, the M_{PEO} dependence is profound. SEO(74-98) exhibits near-unity σ_{N} values, which indicates that the PEO domains must be well-connected across grain boundaries. Low-molecular-weight SEOs, however, are an order of magnitude less conductive, which indicates that the PEO domains must be poorly connected across grain boundaries. It is clear from these results that the dependence of σ_{SEO} on M_{PEO} can be attributed, at least in part, to annealing effects.

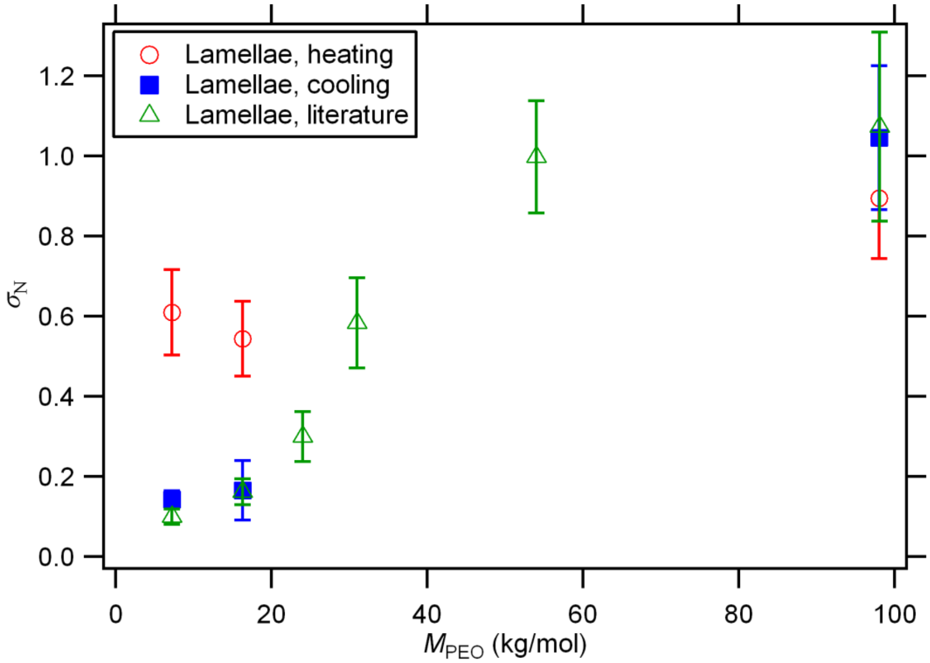


Figure 17 – Normalized conductivities of lamellar SEO samples.
Literature results derived from Panday et al.[19] for the same SEOs are included for comparison. All data is at 80 °C and all samples have r = 0.085. As-prepared samples exhibit conductivities that are 50 to 100 % of the theoretically anticipated maximum. After annealing, conductivities range from 10 to 120% of the theoretical prediction. σ_{N} is calculated with f = 2/3.

3.4 Conclusions

Conductivities of SEO/LiTFSI electrolytes, σ_{SEO}, were measured as a function of temperature. Insight was gained by evaluating normalized conductivity values, σ_{N}, which were calculated by Equation 16. Samples were prepared by pressing freeze-dried electrolytes into a spacer and are expected to exhibit disorganized morphologies. Upon heating, σ_{SEO} follows typical temperature dependences up to the range 90 to 110 °C, at which point most SEOs undergo large, irreversible conductivity drops. This annealing phenomenon occurs near the T_g of polystyrene, and it most likely reflects microstructural rearrangements that affect the connectivity and tortuosity of the conductive PEO/LiTFSI domains across SEO grain boundaries. Before annealing, both the lamellar and cylindrical samples exhibit weak dependences on M_{PEO}, and σ_N ranges from 0.5 to 0.75 for most samples – indicating that the PEO/LiTFSI domains are well connected throughout the sample. After annealing, σ_{SEO} values for samples with cylindrical morphologies exhibited no clear trend with M_{PEO}, whereas those for samples with lamellar morphologies increase with M_{PEO} in agreement with previously reports.[19, 20] The exact morphological changes that occur during annealing are unclear. However, block-copolymer annealing processes are generally anticipated to produce well-defined grains, which contrasts the poorly defined microstructures that originate from freeze-dried source material. The results of this study suggest that the presence of well-defined block-copolymer microstructures is insufficient to provide good conductivity, and that well-defined microstructures may be detrimental to conductivity in some cases.

Chapter 4 – Characterization of Large Microstructures by Resonant Soft X-ray Scattering

ABSTRACT

Resonant soft X-ray scattering (RSOXS) is used to determine the morphologies and domain spacings of high-molecular-weight block-copolymer electrolytes. The sample domain spacings ranged from 116 to 242 nm, which are well outside the accessible range of typical small-angle X-ray scattering. Sample morphologies were determined to be lamellar or cylindrical. Transmission electron microscopy was used to verify the morphologies. We demonstrated that the RSOXS technique is a simple but effective method for determining morphologies of large polymers.

4.1 Introduction

Resonant soft X-ray scattering (RSOXS) is an X-ray scattering technique that is analogous to small-angle X-ray scattering. The key difference between the two techniques is the operating X-ray wavelength, λ. SAXS beamlines typically operate with hard X-rays in the energy range of 10 to 15 keV (λ = 0.12 to 0.08 nm), whereas RSOXS operates with soft X-rays in the energy range of 0.2 to 1.0 keV (λ = 6.2 to 1.2 nm). RSOXS is typically employed as a method to adjust the X-ray scattering contrast between nanostructured domains. This technique is particularly useful for block copolymers because the energy range encompasses the X-ray absorption edges of light elements that comprise typical polymers, such as carbon, nitrogen and oxygen.[70-72] In this work we demonstrate the utility of RSOXS for a different purpose: resolving block-copolymer structures with domain spacings greater than 100 nm. Block-copolymer morphologies are routinely determined via small-angle X-ray scattering (SAXS). However, the accessible range of scattering vectors, q, for most SAXS instruments is limited to samples with domain spacings less than 100 nm. Larger block-copolymer structures are typically determined via transmission electron microscopy (TEM), ultra-small-angle X-ray scattering (USAXS) or ultra-small-angle neutron scattering (USANS). RSOXS provides an alternative route to resolve large block-copolymer microstructures due to its use of longer wavelength X-rays, which allows access to lower q-values, according to Equation 17

$$q = 4\pi\lambda^{-1}\sin(\theta/_2) \qquad [17]$$

where θ is the scattering angle. The RSOXS instrument configuration is qualitatively similar to the SAXS configuration drawn schematically in Figure 1. However, because RSOXS utilizes soft X-rays, the entire scattering experiment is performed in a high-vacuum chamber to prevent beam attenuation via absorption in air. RSOXS experiments were performed at the Advanced

Light Source beamline 11.0.1.2, which has a maximum sample-detector distance of 0.167 m. SAXS experiments were performed at the Advanced Light Source beamline 7.3.3 which has a significantly longer sample-detector distance of 4.4 m. Despite this limited sample-detector distance, the long wavelength of the RSOXS instrument enabled the determination of the morphologies of three high-molecular-weight block-copolymer electrolytes that could not be determined by SAXS. TEM was used to confirm sample morphologies independently.

4.2 Methods

Morphological information at various length-scales was compared with SAXS, RSOXS, and TEM experiments. Sample preparation, experimental methods and data analysis for each of these morphological-determination methods are compared in the following sub sections.

4.2.1 Resonant Soft X-ray Scattering (RSOXS) Experiments

Polystyrene-*b*-poly(ethylene oxide) (SEO) block copolymers were synthesized, characterized, purified, and freeze-dried as reported previously.[19, 20, 24, 30] Dry (solvent-free) electrolyte mixtures of SEO and lithium bis(trifluoromethanesulfone)imide (LiTFSI) were prepared by dissolving approximately 0.3 g of SEO in 20 mL of benzene and adding a mixture of LiTFSI/tetrahydrofuran with a micropipette to obtain the desired SEO/LiTFSI ratio. These steps were performed inside of an argon glovebox (MBraun) with sub-ppm levels of oxygen and water. The electrolyte mixtures were then placed in a desiccator and freeze-dried using an "air free" procedure in which the samples were never exposed to air. Further details on this procedure have previously been published.[30]

Freeze-dried electrolyte samples of SEO(240-269), SEO(352-166), and SEO(247-116) were dissolved in *N*-methyl pyrrolidone (NMP) to a concentration of 5 mg/mL. The samples all had the same salt concentration*,* r = [Li]/[EO] = 0.085. One drop of each electrolyte solution was placed on the center of a 100 nm thick, 1.5 × 1.5 mm silicon nitride membrane supported by silicon (Silson). NMP was evaporated by placing each membrane on a hot plate at 130 °C for 1 hour. Samples were further dried and annealed under vacuum at 130 °C for 12 hours. Sample-coated membranes were affixed to a sample stage with carbon tape, and the entire stage was heat-sealed inside of an air-tight pouch (Showa-Denko) and transferred to the Advanced Light Source. The RSOXS scattering chamber was vented with dry nitrogen. The sample stage was removed from the pouch, mounted into the scattering chamber, and the chamber was evacuated. These steps were performed as quickly as possible to minimize exposure of the samples to ambient air.

The RSOXS sample-detector distance was extended to the maximum value of 167 mm. The beamstop was adjusted such that the occluded region of the detector near the beam center was minimized. The beam center and sample-detector distance were calibrated with a polystyrene-*b*-poly(2-vinyl pyridine) block-copolymer sample with a known domain spacing of 39.1 nm. SEO/LiTFSI scattering profiles were obtained at an X-ray energy of 280 eV (λ = 4.4 nm) using 1 s exposures. These experiments were performed at room temperature. The 2D detector images were reduced to 1D profiles using the Nika software package for Igor Pro[33] and custom RSOXS-specific code developed by Eliot Gann at North Carolina State University.

4.2.2 Small-Angle X-ray Scattering (SAXS) Experiments

Freeze-dried samples of SEO(240-269), SEO(352-166), and SEO(247-116) at r = 0.085 were hot-pressed into Garolite G-10 spacers with 3.86 mm inner diameters. The samples were loaded into home-built air-tight aluminum sample holders composed of gasket seals, Viton O-rings, and Kapton windows. The sample holders were tightly screwed together, and were then placed in the glovebox antechamber heating tray and heated at 130 °C for 2 days under vacuum. Samples were transferred to the Advanced Light Source inside of a desiccator. The detector was positioned at the maximum distance of 4.4 m from the sample stage. A small, suspended beamstop was used to minimize the occluded detector area. The beamcenter and sample-detector distance were calibrated with a silver behenate standard. The sample holders were mounted in a stage, and scattering images were taken at room temperature using 1 s exposures. The 2D detector images were reduced to 1D profiles using the Nika software package for Igor Pro.[33]

4.2.3 Transmission Electron Microscopy (TEM) Experiments

Salt-free samples of SEO(240-269), SEO(352-166), and SEO(247-116) were dissolved in benzene and solvent annealed over a two-week period. Samples were further annealed under vacuum for 2 days at 130 °C. Annealed samples were cryo-microtomed at -120 °C to yield sections with thicknesses on the order of 100 nm using an RMC Boeckeler PT XL Ultramicrotome. The microtomed sections were then stained using ruthenium tetroxide (RuO_4) vapor for 25 minutes to enhance contrast between the polystyrene (PS) and poly(ethylene oxide) (PEO) domains. RuO_4 vapor reacts with both PS and PEO, but the reaction with PEO is faster. As a result, the PEO domains appear dark in the TEM images, whereas the PS phase appears light. Stained samples were imaged with a Zeiss LIBRA 200FE microscope operating at 200 kV and using an Omega energy filter. Images were recorded on a Gatan 2048 x 2048 pixel CCD camera. All images were acquired and analyzed using Digital Micrograph software (Gatan Inc.).

4.3 Results and Discussion

SAXS profiles for SEO(240-269), SEO(352-166) and SEO(247-116) at r = 0.085 are shown in Figure 18. The scattering profile was truncated in the range $q < 0.04$ nm^{-1} due to direct occlusion of the detector by the beamstop. The region $0.04 \leq q \leq 0.06$ nm^{-1} most likely contains interference from parasitic scattering around the beamstop that occurs when the beamstop is fully illuminated by samples with significant low-q scatter. This q region contains scattering contributions that likely arise from the block-copolymer morphology, but the beamstop interference obscures the signal. Each of the three SAXS profiles contains subtle humps and features that are indicative of scattering peaks, but none had a well-defined primary scattering peak, q^*. Using the theoretical block-copolymer phase diagram, the polymer morphologies can be predicted from the volume fraction of the PEO phase. From this prediction, SEO(240-269) is predicted to be lamellar and SEO(352-166) and SEO(247-116) are predicted to be cylindrical. In the absence of a primary scattering peak, the SAXS profiles cannot verify these predictions.

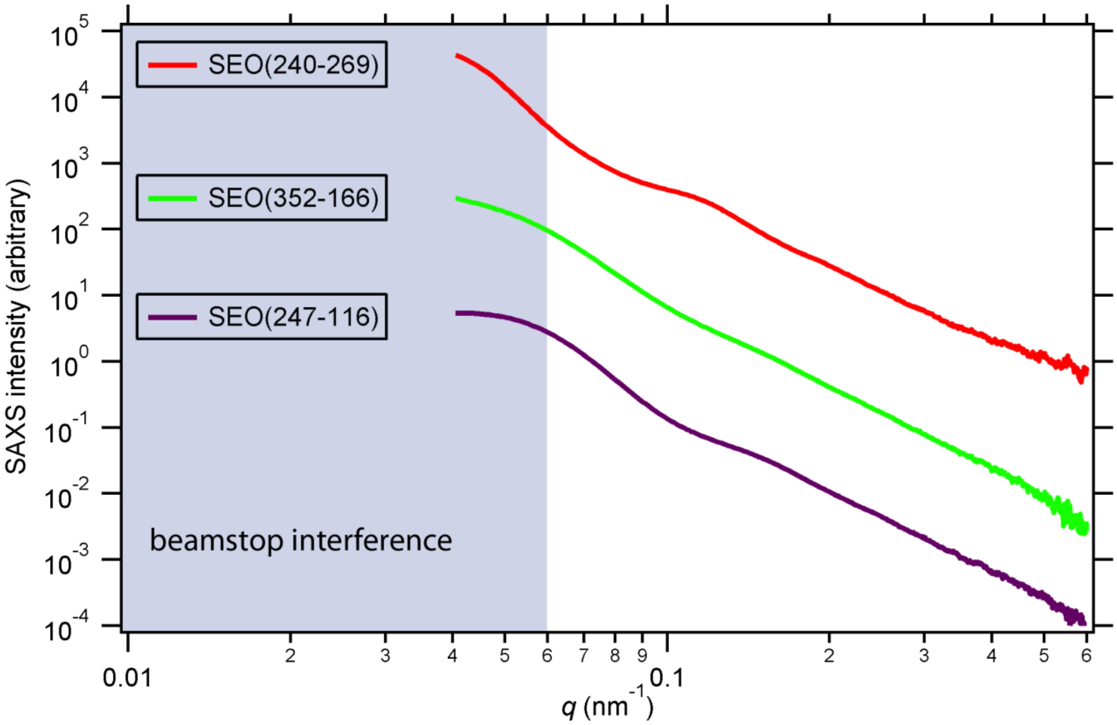


Figure 18 – SAXS profiles for high-molecular-weight SEO samples.
Beamstop interference below q = 0.06 nm^{-1} limits the observable range of domain spacings. The traces are vertically offset for clarity.

RSOXS profiles for SEO(240-269), SEO(352-166), and SEO(247-116) at r = 0.085 are shown in Figure 19. Expected scattering peak locations are marked with triangles. The scattering images were occluded by the beamstop in the range q < 0.0112 and that region was truncated from the images. Similar to the SAXS profiles, a small portion of the RSOXS scattering profiles in the range 0.0112 ≤ q ≤ 0.0126 contains contributions that likely arise from parasitic scattering around the beamstop. Thus, the RSOXS scattering profiles are well resolved in the range 0.013 ≤ q ≤ 0.06, but that region is either occluded or noisy for the SAXS profiles. The morphologies of each of the SEO samples can be readily determined from these profiles. SEO(240-269) has a broad primary scattering peak at q^* = 0.026 nm^{-1} (corresponding to a 242 nm domain spacing) and weak higher-order scattering peaks. The higher-order peak positions are in agreement with those expected from the lamellar morphology. The first higher-order peak is at $2q^*$. The broad higher-order peak between 0.078 and 0.104 nm^{-1} is most likely an overlap of the $3q^*$ and $4q^*$ peaks, although neither peak is readily discernible. SEO(352-166) has a broad q^* peak at 0.051 nm^{-1} (corresponding to a 124 nm domain spacing), and the higher-order peak spacings are consistent with the cylindrical morphology. The first higher-order peak is consistent with an overlap of the $\sqrt{3}q^*$ and $2q^*$ peaks. The broad hump centered around 0.18 nm^{-1} is consistent with the $\sqrt{12}q^*$ peak. SEO(247-116) has a broad q^* peak at 0.054 nm^{-1} (corresponding to a 116 nm domain spacing), and exhibits higher-order scattering that is consistent with the cylindrical morphology. The broad peak centered around 0.1 nm^{-1} is consistent with the $\sqrt{3}q^*$ peak and may contain contributions from the $2q^*$ peak. The RSOXS data thus confirm the expected morphologies of the three polymers in this study. In comparison, the SAXS data are just outside of the range needed to capture the primary scattering peak of the smallest polymer, SEO(247-116), although hints of that peak and one higher order peak are apparent.

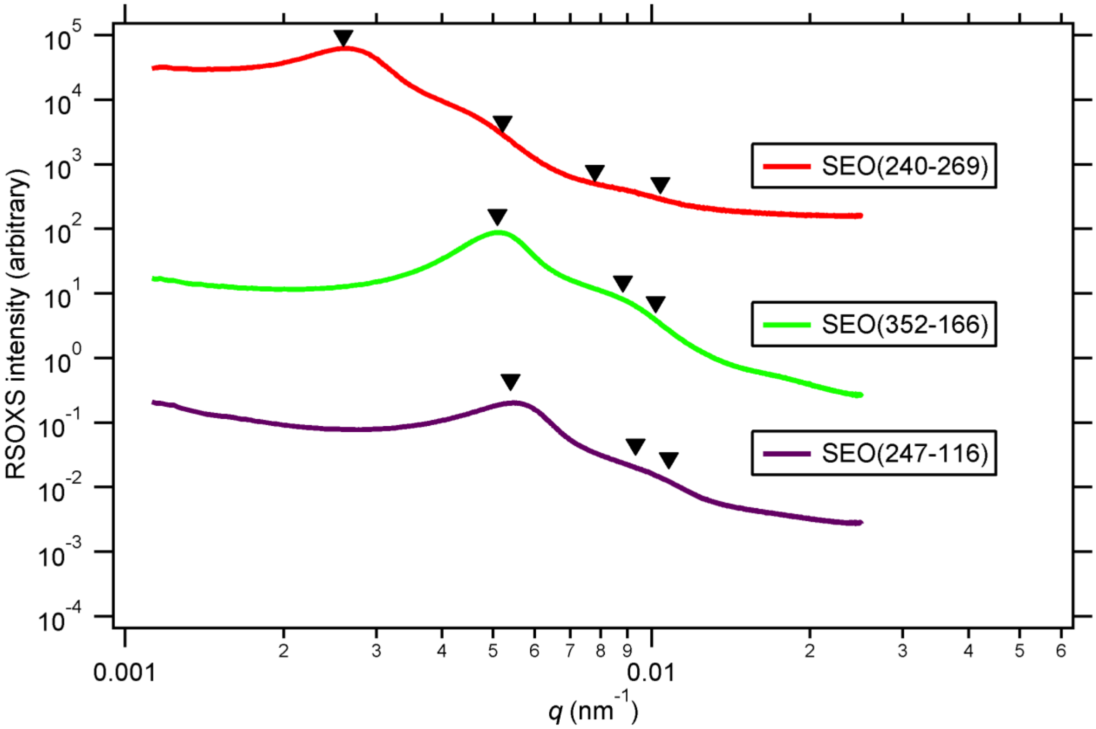


Figure 19 – RSOXS profiles of high-molecular-weight SEO samples.
The q range is unobstructed above 0.013 nm^{-1}; thus allowing determination of microstructures with large domain spacings. The symbols denote lamellar reflections at q/q^* locations of 1, 2, 3 and 4 for SEO(240-269), and cylindrical reflections at q/q^* locations of 1, $\sqrt{3}$, 2, and $\sqrt{12}$ for SEO(352-166) and SEO(247-116). The traces are vertically offset for clarity.

The RSOXS scattering profiles are very broad, most likely as a result of the high molecular-weights of the SEO copolymers. Improved sample annealing could potentially result in sharper scattering peaks, although the simplicity of the RSOXS sample preparation is advantageous. In order to confirm the morphologies independently, TEM of well-annealed samples was performed. Notably, both the solvent and thermal annealing steps for TEM sample preparation were significantly longer than those for RSOXS sample preparation. The TEM samples also differ in that they contain no salt, although previous work on SEO/LiTFSI mixtures indicates that morphologies of high-molecular-weight samples are insensitive to LiTFSI content. [19] TEM samples were also prepared under ambient conditions, and thus the water content of the PEO domains was not rigorously controlled. Furthermore, TEM staining and microtoming procedures can affect sample morphologies and domain spacings.[73, 74] Despite these caveats, the TEM results confirm the qualitative morphological determinations. Figure 20 shows a representative TEM image of SEO(240-269), which clearly exhibits the lamellar morphology. Figure 21 and Figure 22 show TEM images of SEO(247-116) and SEO(352-166), respectively. Both samples exhibit hexagonally packed cylinders of PEO in a PS matrix, as expected. The cylinders in Figure 21 and Figure 22 have faceted cross-sections. This is most likely due to crystallization of the PEO domains. The SEO(352-166) sample in Figure 22 exhibits a poorly organized microstructure and a distribution of sizes and shapes for the cylinders. This is possibly due to kinetic trapping of the structure that occurs due to the high molecular-weight and low mobility of the PS blocks.

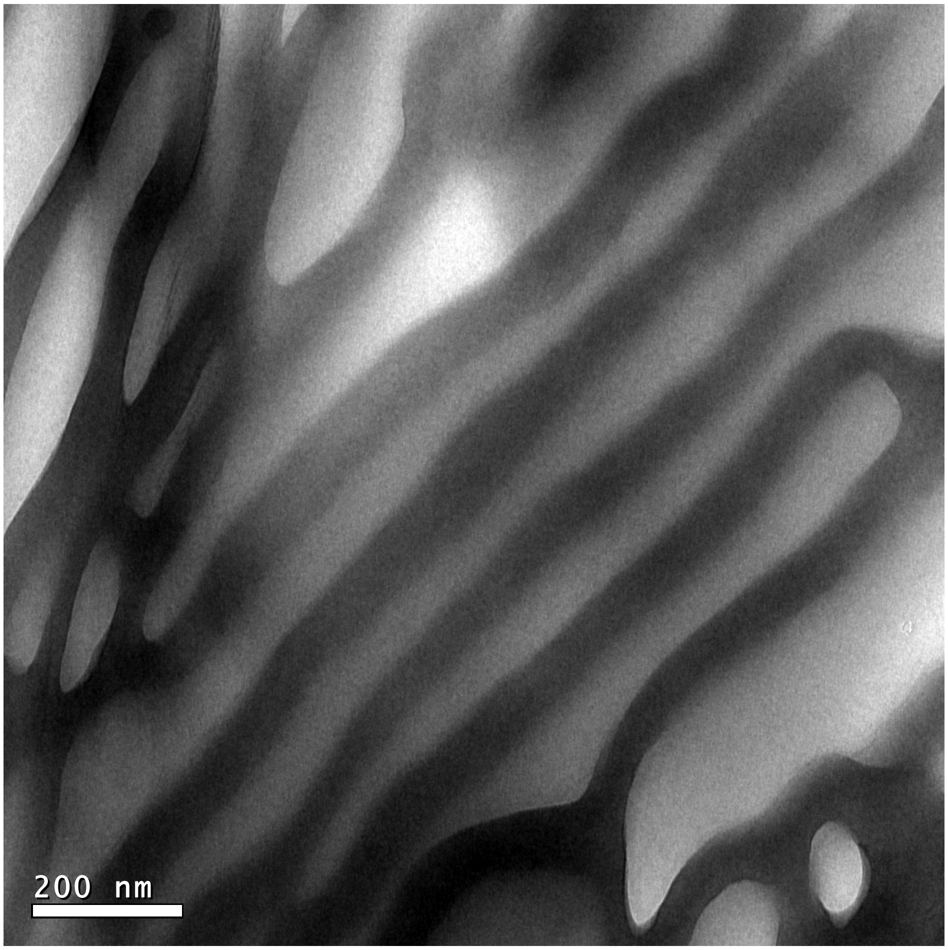


Figure 20 – TEM image of stained, neat SEO(240-269).
This image confirms the lamellar morphology suggested by RSOXS. The PEO regions are dark, whereas the PS regions are light.

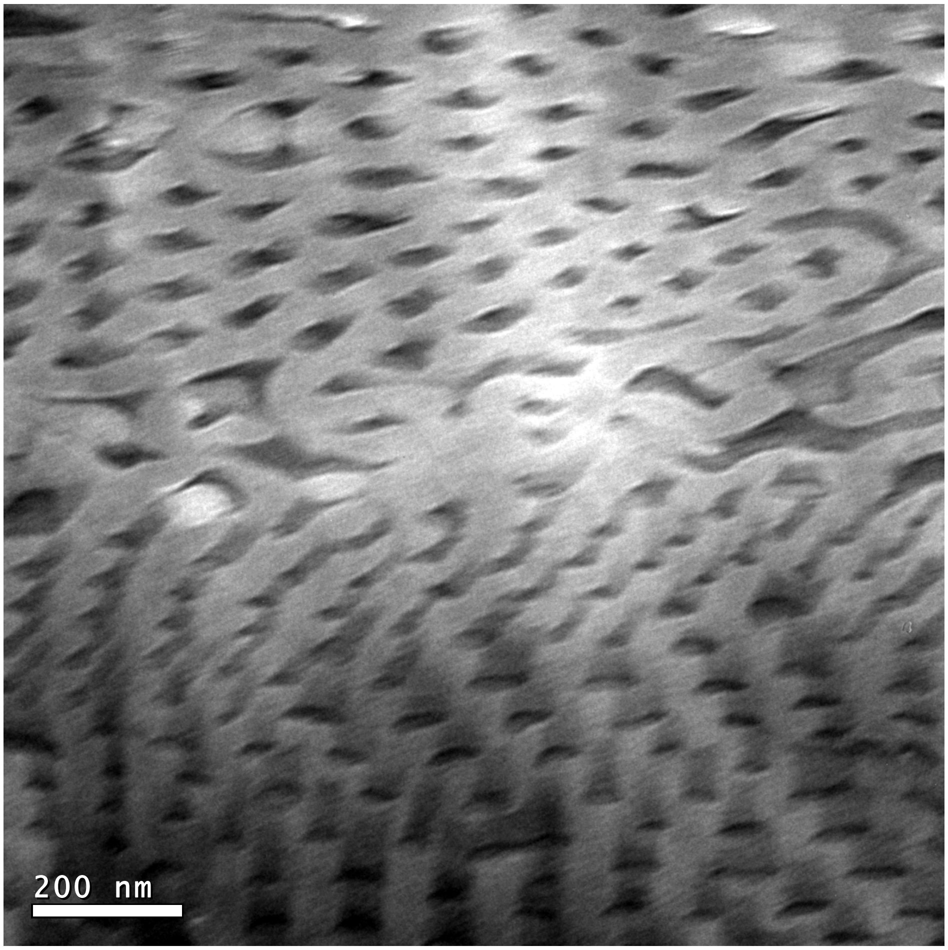


Figure 21 – TEM image of stained, neat SEO(247-116).
This image confirms the hexagonally-packed cylindrical morphology suggested by RSOXS. The PEO regions are dark, whereas the PS regions are light.

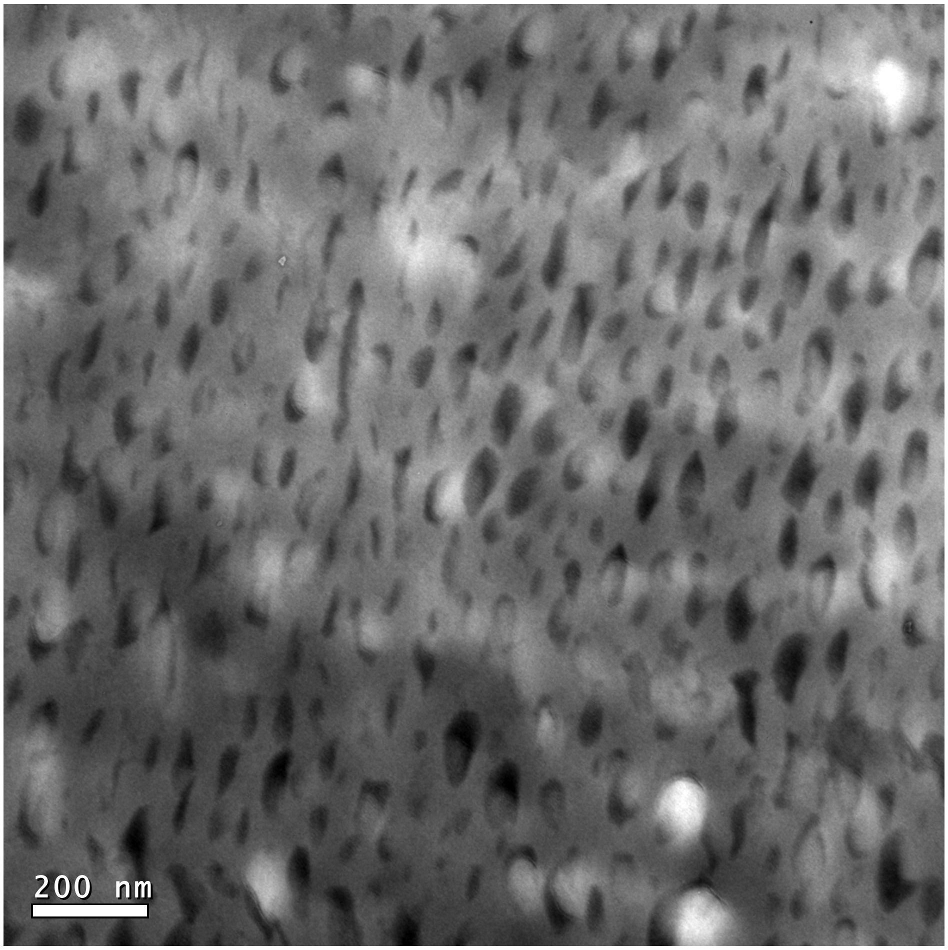


Figure 22 – TEM image of stained, neat SEO(352-166).
This image confirms the hexagonally-packed cylindrical morphology suggested by RSOXS. The PEO regions are dark, whereas the PS regions are light.

4.4 Conclusions

In this study, the morphologies of three high-molecular-weight block-copolymer electrolytes were resolved via RSOXS. The soft X-rays used by RSOXS enable the resolution of scattering features at significantly lower *q*-values than are attainable via SAXS, and RSOXS sample preparation, data acquisition and analysis are simpler and more rigorous than those of TEM. TEM images confirmed the morphological identification of the polymers, although differences and complications in the sample handling complicate direct comparisons between TEM and RSOXS. The primary limitation of RSOXS is that it must be performed in a vacuum chamber and relatively thin samples (less than 1 μm thick) must be used. The ease of sample preparation and data acquisition overcomes this limitation in many practical cases.

Chapter 5 – Transference Numbers in Block-Copolymer Electrolytes

ABSTRACT

Transference numbers for a high-molecular-weight block-copolymer electrolyte were measured using the rigorous method outlined by Ma et al.[27] The method requires the determination of three experimental parameters: the salt diffusion coefficient, a parameter determined from current-interrupt experiments, and a parameter determined from concentration-cell experiments. This method was developed for homogeneous electrolytes, and this work is the first report in which it has been applied to microstructured electrolytes. For a sample with a salt concentration, r = [Li]/[EO] = 0.085 at 90 °C, the best estimate of the lithium transference number is -0.75, although problems with the concentration-cell experiment give rise to large experimental noise. The results and interpretation of the concentration-cell and current-interrupt experiments are discussed at length.

5.1 Introduction

A complete description of electrolyte ion transport processes requires knowledge of $n(n - 1)/2$ parameters, where n is the number of independent species in the solution.[27] A binary electrolyte solution consists of one cation, one anion, and one solvent species, giving $n = 3$. The three necessary transport parameters can be chosen as the electrolyte conductivity, the salt diffusion coefficient, and the cation transference number. The limiting current in batteries is extremely sensitive to the transference number, as clearly illustrated by Thomas et al.[75] The literature on ion-transport measurements in block-copolymer electrolytes is limited almost exclusively to conductivity measurements.[6-26, 64-66] Fully parameterizing the ion transport properties of block-copolymer electrolytes provide insight as to how those parameters compare to homopolymer electrolyte analogues and will better inform the design of block-copolymer electrolytes with improved ion-transport characteristics. The restricted-diffusion study in Chapter 2 describes the measurement of salt diffusion coefficients, D, in electrolytes of polystyrene-*b*-poly(ethylene oxide) (SEO) mixed with lithium bis(trifluoromethanesulfone)imide (LiTFSI) salt. The key insight gained from that study is that the relaxation behavior of SEO/LiTFSI systems is best described by a distribution of time constants that is parameterized by an effective diffusion coefficient D_{avg} and a polydispersity index $PDI_{\text{diffusion}}$. Furthermore, high-molecular-weight SEO systems have conductivities that are approximately 1/3 those of poly(ethylene oxide) (PEO) homopolymer electrolytes and salt diffusion coefficients that are approximately 2/3 those of PEO homopolymer electrolytes. Transference-number measurements in SEO/LiTFSI were expected to highlight additional differences between SEO/LiTFSI and PEO/LiTFSI. In particular, the transference number is intimately related to the salt activity coefficient. If the transference number of SEO/LiTFSI matches that of PEO/LiTFSI, then it would support the premise that the PEO/LiTFSI channels in SEO are the same as bulk PEO homopolymer and that the ion transport properties of SEO are governed by geometric consideration such as the PEO channel geometry in the SEO microstructure and connectivity across grain boundaries. Significantly different transference numbers between PEO and SEO electrolytes would indicate that the SEO microstructure affects the local environment around Li^+ and $TFSI^-$, possibly due to conformational changes of the PEO chains imposed by SEO self-assembly, as proposed in previous work.[9, 19, 20]

In the present study, Li^+ transference numbers are measured using the method devised by Ma et al.[27] In this method, the transference number is calculated using parameters from three measurements: 1) restricted-diffusion experiments, 2) current-interrupt experiments, and 3) concentration-cell measurements. The restricted-diffusion method gives the salt diffusion coefficient, as described in Chapter 2. The current-interrupt measurement provides information about the concentration overpotential as a function of charge passed into the cell, and the concentration-cell experiment is used to relate the concentration overpotential to actual concentration gradients in the cell. All of these experiments are performed on symmetric cells with 1-dimensional geometry. The cation transference number, t_+^0 (the superscript 0 denotes the use of the solvent (PEO ether groups) velocity as a reference velocity) is calculated from these experimental parameters using Equation 18

$$t_+^0 = 1 - \frac{mc_\infty F(\pi D)^{1/2}}{4}\left(\frac{d\ln(c)}{dU_{\Delta c}}\right) \quad [18]$$

where m is obtained from the current-interrupt experiment, c_∞ is the bulk solution concentration, F is Faraday's constant (96,487 C/equivalent), and $\frac{d\ln(c)}{dU_{\Delta c}}$ is obtained from concentration-cell measurements. This method was developed using concentrated solution theory and was not designed to account for a microstructured electrolyte such as the SEO copolymers in this study. The general method for homopolymer electrolytes is described next. Adjustments to data analysis for each experiment to account for the block-copolymer structure are detailed in the Results and Discussion section.

This method of measuring the transference number is prone to error since the transference number is calculated from the product of three experimental parameters. In particular, it is difficult to get an accurate estimate of $\frac{d\ln(c)}{dU_{\Delta c}}$, as described below. Furthermore, the method is sensitive to experimental signal contributions that arise from side reactions.[76, 77] However, despite the fact that the accuracy of this method is limited by experimental noise, it is the only method that yields a rigorous value for t_+^0. A similar method exists for determining t_+^0 without requiring the concentration-cell experiment if the activity coefficient of the salt in the solution is known.[76] However if m and D are well known, then the activity coefficient can be determined from t_+^0 and vice versa. The concentration-cell experiment essentially provides information about the activity coefficient, which is apparent from Equation 19, as given by Ma et al.[27]

$$\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right) = \frac{-F}{2RT(1-t_+^0)}\left(\frac{dU_{\Delta c}}{d\ln(c)}\right) \quad [19]$$

where $f_\pm$ is the mean molar activity coefficient and R is the universal gas constant. Several studies have used the concentration-cell method on a variety of dry polymer electrolytes, including PEO/sodium triflate (NaTf),[27] PEO/NaTf and PEO/NaTFSI,[78] PEO/NaTFSI,[79] poly(propylene oxide)(PPO)/LiTf,[44] oxymethylene-linked PEO (PEMO)/LiTf and PEMO/LiTFSI,[80] PEO/LiTFSI,[28] and poly(ethylene oxide) dimethyl ether (PEODME) with triphenyloborane additive.[81] The galvanostatic polarization method, in which the activity coefficient is known from independent experiments, has been applied in a few studies,[70, 76] and generally appears to be a more accurate measurement of the transference number. Numerous studies have utilized the "steady-state current" method developed from dilute solution theory;[82] however that method has been proven to fail even in the limit of a dilute, ideal solution.[76, 83]

5.2 Methods

All preparation steps were performed in an argon-filled glovebox with sub-ppm water and oxygen levels. The current-interrupt and concentration-cell measurements are described separately.

5.2.1 Current-Interrupt Experiments

In the current-interrupt experiments, a symmetric cell comprised of lithium metal | polymer electrolyte | lithium metal is initially equilibrated at the temperature of interest. A fixed current density, i, is applied for a set period of time, t_j, after which the current is interrupted and the cell potential is monitored. The purpose of this experiment is to determine the concentration overpotential at time t_j. In order for Equation 18 to be valid, the diffusion layers must not extend far from the electrodes during the polarization step. This condition is met when Equation 20 is satisfied

$$(4Dt_\mathrm{j})^{1/2} << L/2 \qquad [20]$$

where L is the sample thickness. Once the current is interrupted, the electric double-layer at each electrode decays rapidly (in less than a second). The overall concentration gradient (and the associated concentration overpotential) also decays, but the timescale is on the order of seconds or minutes. Hafezi and Newman developed a method to extrapolate the open-circuit-potential, $V_\mathrm{interrupt}(t)$, to get an accurate estimate of U_j, the concentration overpotential at time t_j.[76] In this method, the dimensionless parameter τ is used as defined in Equation 21

$$\tau = \frac{\sqrt{t_\mathrm{j}}}{\sqrt{t}+\sqrt{t-t_\mathrm{j}}} \qquad [21]$$

where t is the elapsed time since the onset of current passage. The linear portion of $V_\mathrm{interrupt}(t)$ vs. τ is extrapolated to $\tau = 1$ (corresponding to $t = t_\mathrm{j}$) in order to determine U_j rigorously. The cell is allowed to relax until equilibrium is reached (corresponding to a constant baseline voltage that arises due to the thermocouple effect). The experiment is then repeated at multiple combinations of i and t_j. Finally, m is determined from the slope of a plot of U_j vs. $it_\mathrm{j}^{1/2}$.

Current-interrupt samples using SEO/LiTFSI were prepared as follows. A large sample (0.5 g) of SEO(74-98) at r = 0.085 was dissolved to a concentration of 10 mg/mL in N-methyl pyrrolidone (NMP). The solution was cast on the surface of an 18 μm-thick Ni foil on a home-built solvent caster with the doctor blade set at 500 μm to regulate the solution thickness. The film was dried on the solvent caster overnight at 60 °C. The edges of the dried film were lifted up with tweezers, but the film had a tendency to tear. The caster was cooled to room temperature, but the film was stuck tightly to the foil. Upon re-heating the caster to 50 °C, the free-standing film lifted readily off the foil. The free-standing film was placed between Ni foil sheets and transferred onto the heating tray inside the glovebox antechamber and heated at 120 °C overnight. The final film had a consistent thickness of 45 ± 1 μm. A 9/16 inch (1.43 cm) punch was used to make circular disks of the electrolyte film. A 7/16 inch (1.11 cm) punch was used to cut lithium metal electrodes from 300 μm thick foil (FMC Lithium), and lithium electrodes were placed on both sides of each electrolyte sample. Symmetric lithium | SEO/LiTFSI | lithium cells were pinched together between gloved fingers. Nickel tabs were affixed to the lithium metal with Kapton tape, and the entire cell assemblies were vacuum sealed into pouch material (Showa-Denko). The sealed pouch cells were transferred outside the glovebox to a heating stage consisting of an aluminum heating plate. This is the same heating stage described in Chapters 3 and 6. The heating stage was held at 90 °C, which corresponds better than 0.5 °C to the sample temperatures, as calibrated by fine-gauge wire thermocouples in mock sample cells. The samples were connected to a potentiostat (Solartron 1470E). The samples were allowed to equilibrate for several hours, and then a constant current was applied for time t_j, after which the open-circuit potential $V_{\text{interrupt}}(t)$ was monitored. A sampling rate of 10 measurements per second was used for all experiment steps. The current density, *i,* is found from the applied current and the electrode area, A. Since free-standing films were used, A is well defined (0.97 cm^2). Relaxation times ranged from 30 minutes to 6 hours in order to allow $V_{\text{interrupt}}(t)$ to decay back to the baseline voltage. In practice, these wait times were more than adequate since $V_{\text{interrupt}}(t)$ decayed back to the baseline voltage in less than 5 minutes for most experiments. The nonzero baseline voltage arises from small temperature gradients across the electrodes – known as the thermocouple effect. This contribution typically has a magnitude less than 1 mV for this experiment. The experiment was repeated at values of t_j ranging from 1 to 5 s, and i ranging from 0.05 to 10.3 mA/cm^2. Using a value of $D = 1 \times 10^{-7}$ cm^2/s, a conservative over-estimate of the salt diffusion coefficient, and the film thickness L = 45 μm, the left side of Equation 20 is 63% of the right side for t_j = 5 s, and 28% of the right side for t_j = 1 s. These values are significant, but the plot of U_j vs. $it_j^{1/2}$ is linear across all values of t_j and i used in this study, thus confirming that the experiments were performed within valid limits. In practice, i varies slightly with time due to instrument noise and current-control issues; thus i was determined from an average of all current readings during the short duration between t = 0 and t = t_j.

Figure 23 shows an example set of current-interrupt experiments for a sample of SEO(74-98) at r = 0.085 and 90 °C at various values of i at a fixed value of t_j = 1 s. The value of i was controlled from 2.06 (the bottom curve) to 7.22 mA/cm^2 (the top curve) in increments of 1.03 mA/cm^2. The abrupt jumps in $V_{\text{interrupt}}(t)$ that occur at the onset and interruption of current are due to rapidly-decaying contributions from the electric double layer and sample resistance.

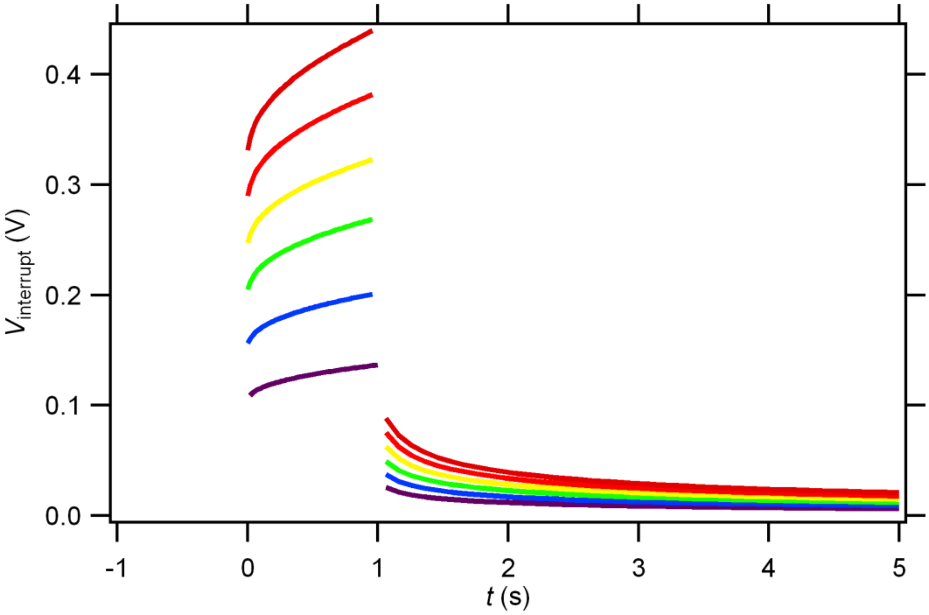


Figure 23 – Current-interrupt experiment for a sample of SEO(74-98).
The concentration was r = 0.085 for all samples, and the temperature was fixed at 90 °C. Each segment of the experiment consisted of a brief charging step (t_j = 1 s) and a long relaxation step (6 hours). The value of t was zeroed at the first point current was applied. The experiment was repeated with various values of i. Each experiment has a different color to provide visual distinction. The discontinuities in the voltage arise from a combination of ohmic losses from the electrolyte, charge-transfer reactions, and the formation of the electric double layer. These factors, along with the contribution from the thermal baseline voltage, prevent the rigorous determination of U_j from this graph, although it is apparent that $U_j < 0.1$ V for all samples.

Figure 24 shows a plot of $V_{interrupt}(t)$ vs. τ corresponding to the middle experiment in Figure 23. The experimental data are shown as open circles, and the solid lines are linear fits to the data. The linear fits are allowed to have a nonzero intercept to account for the nonzero baseline voltages. The fits are then extrapolated to $\tau = 1$, and the baseline offset is subtracted as a correction. The concentration overpotential thus obtained is equivalent to the value of the slope minus that of the intercept. The baseline offset, and thus the intercept, is usually on the order of 1 mV, whereas the slope, and thus the extrapolated value at $\tau = 1$, ranges from 1 to over 150 mV, depending on charging conditions. Finally, m is calculated from the slope of U_j vs. $it_j^{1/2}$.

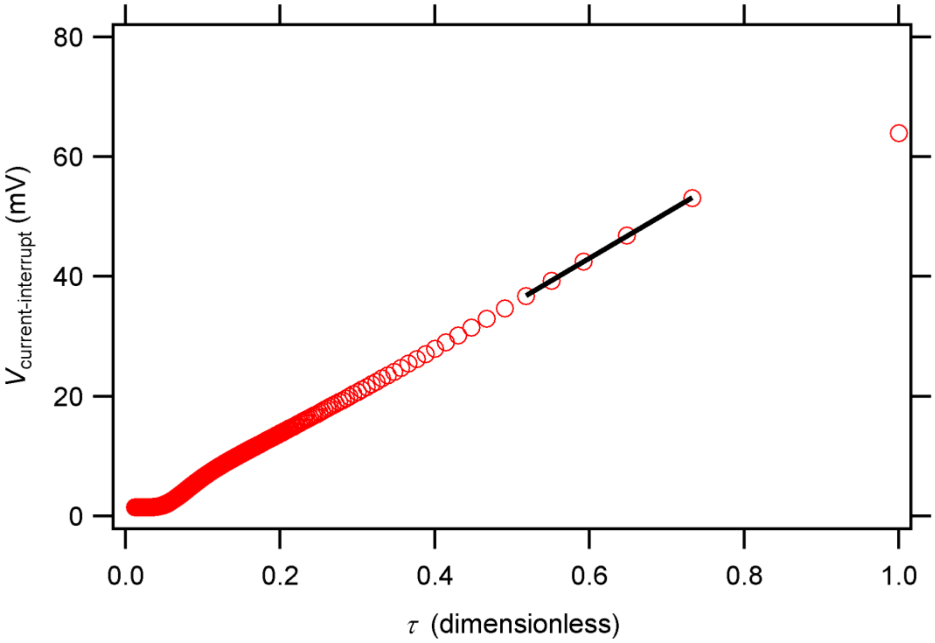


Figure 24 – Demonstration of τ fitting method for analyzing current-interrupt experiments. The symbols represent data points, and the line is a linear fit to the data in the range $0.53 \leq \tau \leq 0.73$. The sample was charged with the conditions $i = 4.12$ mA/cm^2 and $t_j = 1$ s and then allowed to relax for 6 hours. The current is interrupted at $\tau = 1$, the point to which the data are extrapolated. Infinite relaxation time corresponds to $\tau = 0$. The extrapolated value of U_j at $\tau = 1$ is 76 mV, which is significantly larger than the value of 64 mV obtained from the reading at $\tau = 1$. This deviation is due to the finite switching time of the instrument relative to the initial rapid decay of the concentration gradient and double-layer discharging.

Figure 25 shows the plots of U_j vs. $it_j^{1/2}$ for seven different samples. Each curve contains 45 data points. The slope of each trace gives a value of m. Each trace is linear over the entire range of $it_j^{1/2}$, indicating that the experimental parameters were performed within the valid regime. Linear fits were performed in Igor Pro using the built-in Levenberg-Marquardt algorithm. The slope and intercept were free parameters in the fit. In theory, the intercept should be zero for an ideal system. In practice, the magnitude of the intercepts ranged from -200 μV to 800 μV. These relatively small offsets arise due to asymmetries in the system, including contributions from the thermocouple effect, bias in the instrument reading, and inconsistencies in the cell geometry that occur during cell assembly. The offset values serve as an indicator of the quality of the experimental data, but for theoretical purposes they are irrelevant since only the slope is of interest for calculating transference numbers. Each line in Figure 25 has a different value of m, ranging from 11.9 to 19.5 $\frac{\text{mV cm}^2}{\text{mA s}^{1/2}}$ and averaging 15.5 ± 2.9 $\frac{\text{mV cm}^2}{\text{mA s}^{1/2}}$. The source of this spread is unclear.

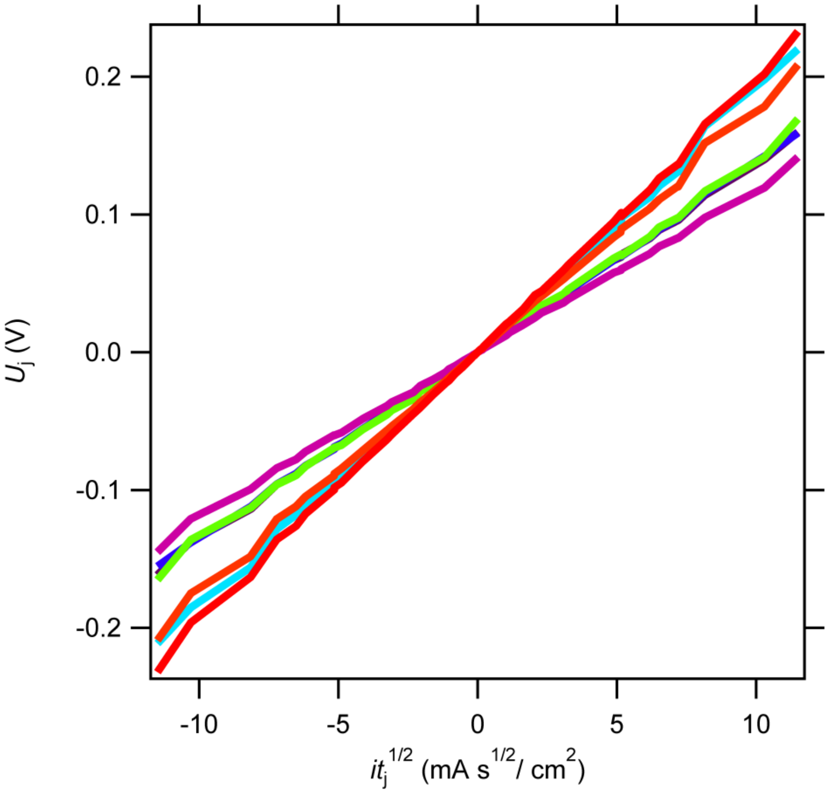


Figure 25 – Current interrupt results for 7 samples, each showing slightly different slope.
The current density in this figure is calculated from the electrode area, *A*. The parameter *m* is determined from the slope of each line.

Because each trace in Figure 25 is linear, the individual samples behaved consistently. All samples were prepared at the same time, and the sample thicknesses and electrode areas were identical within discernible experimental limits. The potentiostat was confirmed to be well-calibrated by measuring test resistors at various applied currents. One possibility is that the switching time in the instrument is not well defined, and that the variation from channel to channel could be significant, thus causing a channel-dependent, systematic deviation in t_j from the programmed setting. Another possibility is that slight temperature variations from sample to sample are enough to affect the response curves. However, the sample-to-sample temperature variations were calibrated and found to be less than 1 °C apart, and even a spread of several °C would not be expected to alter the magnitude of U_j by a factor of 2. It is also possible that each sample had different block-copolymer structure near the lithium interface. If one of the blocks preferentially wets the interface or if some PEO/LiTFSI regions are isolated near the interface, then the charging profiles could differ between samples, but the curves in Figure 25 would remain linear as long as the experiments were performed within the limit proscribed by Equation 21. This remains an unexplained observation, and *m* was calculated from a combination of all the data points, as described next.

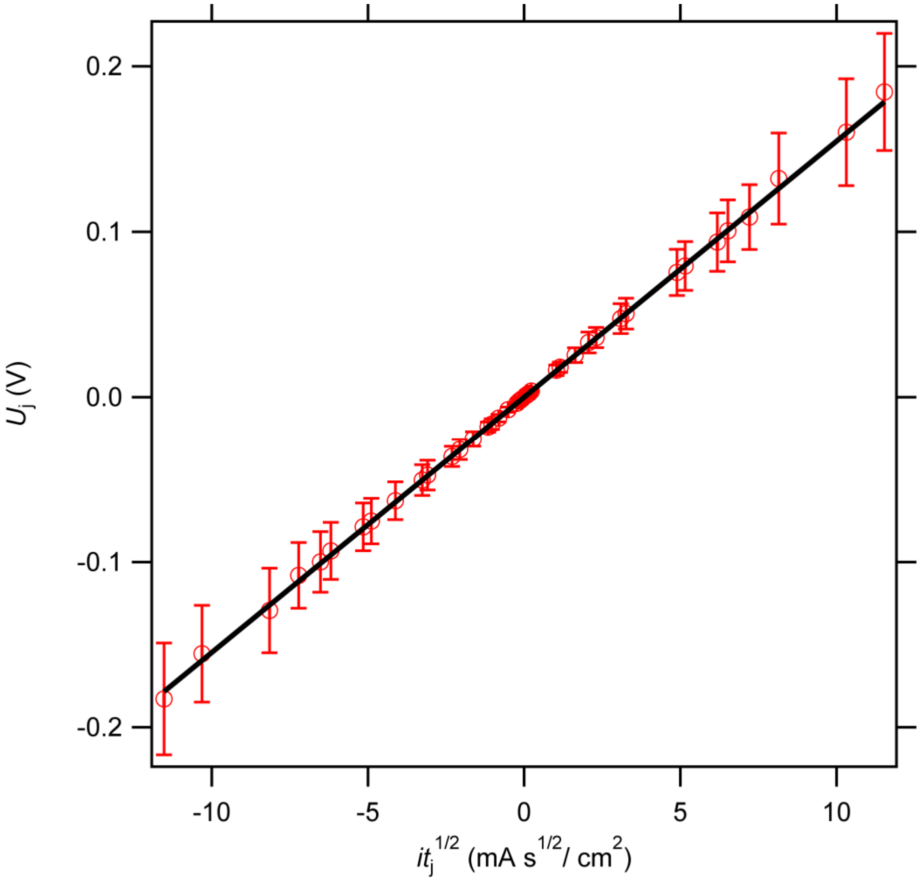


Figure 26 – Current-interrupt results averaged for the samples in Figure 25.
The line is the standard-deviation weighted best linear fit to the data across the entire dataset shown.

For a given current-interrupt experiment, the value of i was calculated from the time-averaged, recorded current from $t = 0$ to $t = t_{\mathrm{j}}$. The relative error for i values measured this way was on the order of 0.1%. Thus, for a set value of i, the value of U_{j} determined for each of the seven samples in Figure 25 can be averaged. This analysis is shown in Figure 26, which shows the current-interrupt results for the data from Figure 25 averaged at each value of i. The error bars are the standard deviations from the seven measurements at each point. The trace in Figure 26 is linear, and the error bars increase with increasing magnitude of $it_{\mathrm{i}}^{1/2}$, thus reflecting the spread of the data. The trace is well-described by a linear fit. Linear fits were performed using the built-in Levenberg-Marquardt algorithm in Igor Pro, using the error bars as weighting. This provides a meaningful estimate of the error in the fit parameters. The slope value was insensitive to the fit region. A fit to the entire region shown in Figure 26 gave the lowest relative error for the slope, with the resulting value being $m = 15.48 \pm 0.43\ \frac{\mathrm{mV\ cm^2}}{\mathrm{mA\ s^{1/2}}}$. In the absence of any physical understanding for the spread of the data in Figure 25, this approach is the best because it captures contributions from all of the samples.

5.2.2 Concentration-Cell Experiments

In the concentration-cell experiments, electrolyte of a fixed salt concentration, c_{ref}, is placed in diffusional contact with the same electrolyte with a different salt concentration c and the open-circuit potential, $V_{\text{concentration-cell}}(t)$, is measured. $V_{\text{concentration-cell}}(t)$ is monitored until a stable plateau is observed, the value of which is taken as the concentration-cell open-circuit concentration overpotential, $U_{\Delta c}$. This experiment is repeated at various values of c, always holding the composition of c_{ref} constant. The parameter $\frac{d\ln(c)}{dU_{\Delta c}}$ is determined from the slope of $U_{\Delta c}$ vs. $\ln(c)$, taken at $c = c_\infty$, where c_∞ is the bulk concentration. c_{ref} can be fixed at any salt concentration. In theory, the shape of the curve $U_{\Delta c}$ vs. $\ln(c)$ should be the same regardless of the value of c_{ref}. Different values of c_{ref} will contribute different additive offsets to $U_{\Delta c}$ that become irrelevant when the derivative is taken. In practice, c_{ref} should be chosen in a concentration region with well-known phase behavior (i.e.: no coexisting phases or crystallinity).

Freeze-dried electrolyte mixtures of SEO(74-98) and LiTFSI at various compositions were prepared as described in Chapter 1. The samples had salt concentrations, r = [Li]/[EO] ranging from 0.020 to 0.115. The samples were dissolved in NMP at a concentration of 10 mg/mL and were then cast onto clean Kapton sheets using a home-built solvent caster. The cast films were dried for 3 to 12 hours on the solvent caster with a surface temperature of 60 °C. The films were then transferred onto the heating tray inside the glovebox antechamber and were further dried under vacuum at 120 °C for at least 12 hours. The films were then cooled, placed into a plastic zip-top bag, and transferred inside of a desiccator to the concentration-cell setup in a different glovebox. The concentration-cell setup was composed of a copper heating block that was controlled with a feedback temperature controller (Watlow EZ-Zone PM) using an internally potted thermocouple. The surface of the heating block was coated with multiple layers of Kapton tape to provide electrical insulation. A fine-gauge wire thermocouple (Omega) was taped to the top surface of the plate near the sample position to serve as a temperature indicator for the sample. The surface thermocouple was connected to a 16-channel thermocouple reader with built-in cold-junction compensation (National Instruments ENET-9213) via a home-built thermocouple feedthrough, and the temperature reading was recorded to file by a home-built LabVIEW program that interfaced with the thermocouple reader (see Appendix 3). The heating plate was placed on top of a 2.54 cm thick sheet of aluminum-backed fiberglass insulation. When measurements were taken, the fiberglass insulation was folded over the heating plate, and the three open edges were held down by a large C-clamp that encompassed the outline of the enclosed heating plate. The Watlow control setpoint was adjusted so that the surface thermocouple reading was 90 °C.

To perform the concentration-cell measurement, a strip of SEO-coated Kapton at a specified value of r was cut from the Kapton sheet, and the bare Kapton portions were taped onto the stage with Kapton tape. The films are typically transparent, and care was taken to keep the coated side oriented upwards. Next, two rectangular strips of pristine lithium metal foil were cut from the source roll using scissors. One of the lithium strips was pressed onto the outer edge of the coated film using a fresh nitrile glove to obtain good contact with the film. Next, two strips of nickel foil were cut from the source roll, and one of the strips was placed over the lithium foil. The nickel foil was taped securely to the stage with Kapton tape. Care was taken to tape the nickel foil so that some pressure was applied downwards on the nickel | lithium metal | SEO/LiTFSI | lithium metal | nickel assembly. Films of SEO(74-98) at a reference concentration of r = 0.085 were used to complete the concentration cell. The experiment was repeated at various concentrations versus the reference concentration. This experiment was performed with SEO(74-98) at r = 0.085 cast on Kapton, as well as on free-standing SEO(74-98) at r = 0.085.

For the Kapton-backed reference films, the lithium foil was placed on the outer edge of the coated portion, and the nickel foil was taped onto the bare Kapton portion. The entire assembly was inverted and positioned such that the free edge of the coated film overlapped that of the other film. In principle, the experiment is not sensitive to the cell geometry since the experiment is designed to sense the open-circuit potential reading of the concentration gradient. In practice, the electrodes need to be in good contact with the film, and the two films need to overlap and provide good diffusional contact. Once the films are in contact, the concentration profile immediately decays. However, if the diffusion distance between the electrodes and the overlap region is large (on the order of 1 cm), then the initial voltage reading will be relatively constant for a period of hours or days, as predicted by Thompson and Newman.[36] For a free-standing reference film, a rectangular strip was cut out using clean scissors, and the film was pressed onto the other Kapton-backed film. Lithium and nickel were then applied as with the other films. Free-standing films adhered very strongly to the Kapton-backed films, whereas pairings of two Kapton-backed films came apart loosely at the conclusion of the experiment.

Before putting the films in diffusional contact, an electrometer (Keithley 6514) was prepared by turning on zero-check and zero-correct. This step calibrates the instrument and prevents current from passing across the measurement electrodes. If the instrument is used without zero-check, then the measurement electrode potential (and the potential reading) increases steadily over time. The electrode potential difference will gradually grow with time and will eventually pose a safety risk as described in the instrument manual. This would also expose the measurement cell to a large potential, thus ruining the experiment. The electrometer cable is connected into the glovebox using a home-built triaxial cable feedthrough and terminates in two alligator clips (red and black). The electrometer can be operated in either guarded or unguarded mode. In the guarded mode, the middle conductor is regulated at the same potential as the inner-most conductor in order to prevent leakage current through the cable insulation. Guarded mode is designed to measure accurately the potential across extremely resistive samples. The electrometer was used in unguarded mode such that the measurement is taken between the inner-most and middle conductors and the outer conductor is used as shielding. The shielding terminates just before the alligator clips and was insulated with Kapton tape. Once the concentration-cell assembly was complete, the alligator clips were attached to the nickel foil strips at each end of the cell, and the open-circuit potential was monitored. The electrometer was controlled via a home-built LabVIEW program (see Appendix 3).

Figure 27 shows concentration-cell results for a symmetric cell where both films are free-standing SEO(74-98) at r = 0.085. $V_{\text{concentration-cell}}(t)$ undergoes an initial period of instability as the sample temperature equilibrates. Concurrently, the surface temperature rises and equilibrates. The temperature and voltage signals both stabilize and plateau within 7 minutes for this example. The initial open-circuit potential is less than -2 mV, but increases, peaks and then plateaus around -1.3 mV with time. $U_{\Delta c}$ is taken as the plateau value. This experiment is instructive because the potential should be identically zero since both films have the same compositions. However, slight temperature gradients between the electrodes produce a thermocouple contribution to the voltage signal of order 1 mV, as seen here. The cell potential is monitored until a stable, constant reading is achieved. This takes up to 30 minutes – the governing factor being the time required to stabilize the surface temperature. An asymmetric cell is shown in Figure 28 for a Kapton-backed film with r = 0.075 against a free-standing film with r = 0.085. The response is similar to that of Figure 27 except that the stable voltage has a much larger magnitude. The concentration gradient will slowly relax over time (a period of days for the large inter-electrode distance used) and a slight trend towards zero voltage is visible.

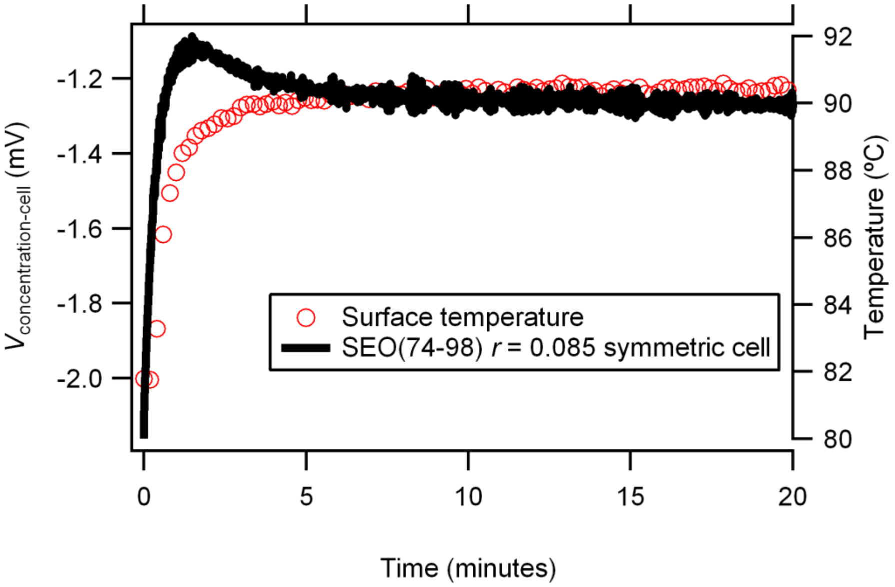


Figure 27 – Symmetric concentration-cell example.

The left axis shows the open-circuit potential of a symmetric concentration cell composed of SEO(74-98) free-standing films, both with r = 0.085 as a function of time. The right axis shows the surface temperature. Both signals plateau within 10 minutes. Lithium metal electrodes were used for both films, and nickel tabs provided electrical connections. $U_{\Delta c}$ is calculated from the average voltage throughout the stable plateau region (about -1.3 mV in this example).

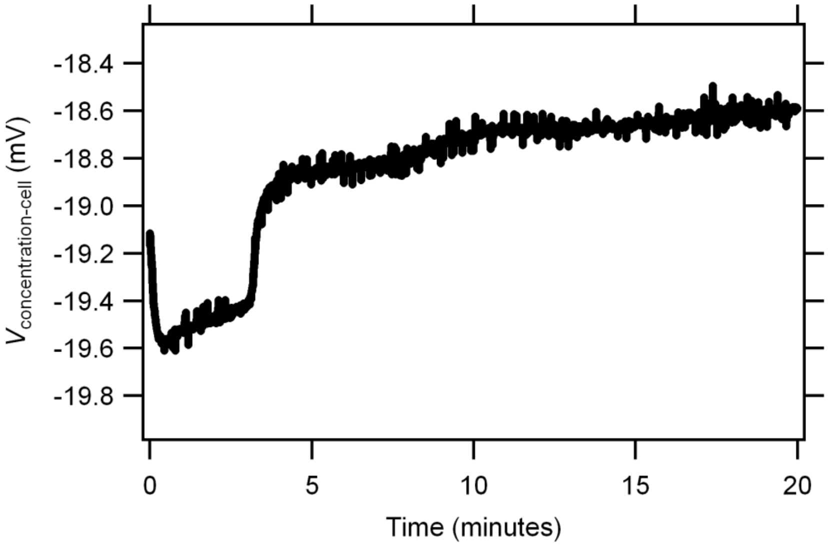


Figure 28 – Asymmetric concentration-cell example.

The left axis shows the open-circuit potential of an asymmetric concentration cell composed of SEO(74-98) Kapton-backed film with r = 0.075 against a free-standing film of SEO(74-98) with r = 0.085 as a function of time. The signal initially has unstable behavior due to the temperature equilibration and cell handling. The signal stabilizes within 5 minutes and slowly decays towards zero. Lithium metal electrodes were used for both films, and nickel tabs provided electrical connections. $U_{\Delta c}$ is calculated from the average voltage throughout the beginning of the stable plateau region (about -18.8 mV in this example).

5.3 Results and Discussion

The concentration-cell experiment was performed with Kapton-backed films of SEO(74-98) with r values ranging from 0.020 to 0.110. Most experiments were near the concentration of interest at r = 0.085. Free-standing and Kapton-backed reference films of SEO(74-98) at r = 0.085 were used to obtain curves of $U_{\Delta c}$ versus salt concentration. From Equation 18, the salt concentration c must have units of [mol/volume]. The salt concentration of polymer electrolytes is typically expressed as a molar ratio (or equivalently a mass ratio) of salt to polymeric repeat units. Estimates of c for a given value of r were obtained by assuming ideal mixing based on the partial molar volumes of the salt and polymer units, as in Equation 22.

$$\rho = \frac{rM_{\mathrm{salt}}+M_{\mathrm{EO}}}{rV_{\mathrm{salt}}^{m}+V_{\mathrm{EO}}^{m}} \qquad [22]$$

where ρ is the solution density [g/cm^3], r is the molar ratio of LiTFSI to ether-oxygen repeat units in PEO, M_{salt} is the molar mass of LiTFSI (287.08 g/mol), M_{EO} is the molar mass of the ether oxygen repeat units, V_{salt}^{m} is the molar volume of LiTFSI (141.9 cm^3/mol) and V_{EO}^{m} is the molar volume of the ether oxygen repeat units (40.87 cm^3/mol). V_{EO}^{m} was obtained from the density of PEO homopolymer at 90 °C (0.928 g/cm^3), as given by Roe.[84] At the limit $r \rightarrow 0$, the density of pure PEO is obtained. At the limit $r \rightarrow \infty$, the density of pure LiTFSI is obtained. This approach was used Georen and Lindbergh[80] for a mixture of poly(ethylene-oxide-*co*-propylene-oxide)/LiTFSI, an analogue to PEO/LiTFSI and was found to agree with density measurements. Values of c can be readily calculated using Equation 23

$$c = \frac{\rho\omega_e}{M_{\mathrm{salt}}} \qquad [23]$$

where ω_{e} is the mass fraction of LiTFSI in the electrolyte, which is calculated as in Equation 24

$$\omega_e = \frac{r_{\mathrm{m}}}{1+r_{\mathrm{m}}} \qquad [24]$$

where r_{m} is the mass ratio of LiTFSI to ether oxygen repeat units ($r_m = r\frac{M_{\mathrm{salt}}}{M_{\mathrm{EO}}}$). Results of these calculations at selected values of r are shown in Table 5, which serves as a convenient reference since concentration is typically expressed in terms of r or its inverse.

Table 5 – Various concentration parameters and densities for PEO/LiTFSI mixtures.
Values of r and ω_e are based on component weights as determined during electrolyte preparation and are accurately known. Values of ρ and c are calculated using the assumption of conserved partial molar volumes as obtained from pure component densities.

		ρ			c	
$1/r$	r	[g/cm^3]	r_m	ω_e	[mol/L]	ln(c [mol/L])
500.0	0.002	1.084	0.013	0.013	0.049	-3.024
50.0	0.020	1.139	0.130	0.115	0.458	-0.782
20.0	0.050	1.218	0.326	0.246	1.042	0.042
18.2	0.055	1.229	0.358	0.264	1.130	0.122
16.7	0.060	1.241	0.391	0.281	1.215	0.195
15.4	0.065	1.252	0.424	0.298	1.298	0.260
14.3	0.070	1.263	0.456	0.313	1.378	0.321
13.3	0.075	1.273	0.489	0.328	1.456	0.376
12.5	0.080	1.283	0.521	0.343	1.532	0.427
11.8	0.085	1.293	0.554	0.356	1.606	0.474
11.1	0.090	1.303	0.587	0.370	1.678	0.517
10.5	0.095	1.312	0.619	0.382	1.748	0.558
10.0	0.100	1.321	0.652	0.395	1.816	0.597
9.5	0.105	1.330	0.684	0.406	1.883	0.633
9.1	0.110	1.339	0.717	0.418	1.948	0.667
8.7	0.115	1.348	0.749	0.428	2.011	0.699

The concentration-cell results are shown in Figure 29. The results from Kapton-backed reference experiments are averages of one to three measurements at each concentration. Error bars are shown for measurements with three measurements. The results from free-standing reference films were obtained from one or two measurements. In general, the two datasets follow the same trend. The free-standing films appear to have a systematically higher voltage on the order of 10 mV. This could possibly be because the free-standing film was prepared separately and thus could have a slightly different salt concentration. The region around r = 0.085 is particularly problematic, however. $U_{\Delta c}$ at r = 0.085 should be zero for both films. The free-standing film has an offset of about 6 mV, which could arise due to a combination of the r value deviating slightly from expected and thermal gradient contributions to the signal. The Kapton-backed film exhibits $U_{\Delta c} < 0$ for $r \geq 0.065$, and the free-standing film exhibits $U_{\Delta c} < 0$ for $r \geq 0.075$ (with the exception of the r = 0.085 value). In theory, $U_{\Delta c}$ should vary monotonically with c and pass through zero at the reference concentration. Both data sets suffer from the fact that $U_{\Delta c}$ does not vary monotonically with c in the region of interest around r = 0.085. Improved experimental data would be helpful, but useful information can be extracted from these data.

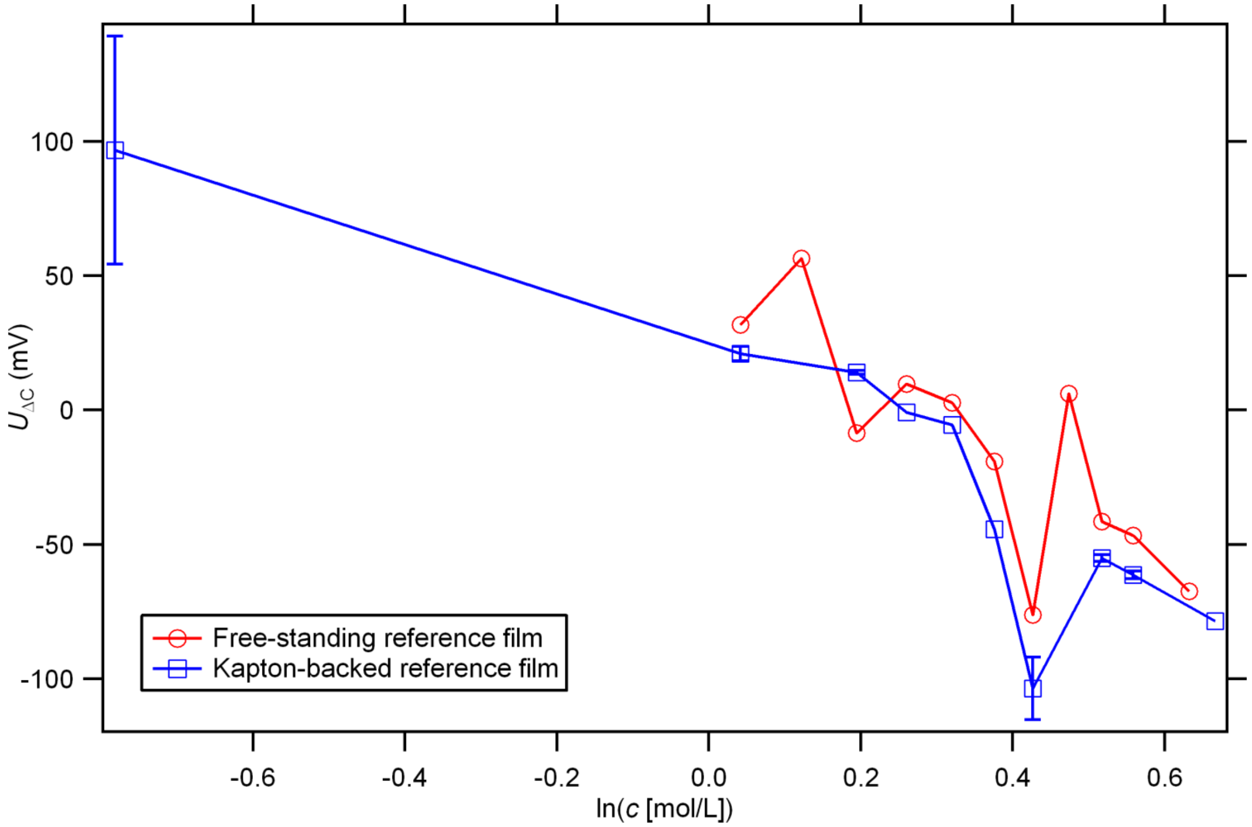


Figure 29 – Concentration-cell results for SEO(74-98) films.
All concentration-cell experiments were performed with reference films with concentrations of $r = 0.085$. Reference films were cast and dried on Kapton (and thus had a Kapton-backing), or were free-standing. The bottom axis is the natural logarithm of the molar salt concentration of the films that were assembled versus the reference. All data are at 90 °C. Error bars represent the standard deviations of cells for which three or more measurements were taken. The lines drawn between data points serve as a visual guide.

It is possible that both of the reference films had the wrong concentration if mistakes were made during electrolyte preparation. Furthermore, some of the films consistently showed problems such as bad contact or physical defects – namely the films at $r = 0.020$ and 0.080. If the measurements on reference films and defective films are excluded, the data become monotonic for the Kapton-backed films (though arguably that is because the data set becomes much smaller). These data are shown in Figure 30. The free-standing films were measured only once for most cases, so that data set is excluded. Due to the noise in the data, meaningful values of $\frac{d\ln(c)}{dU_{\Delta c}}$ can be obtained only by fitting the data and then differentiating the fit equation. Typically, the data are fit with a polynomial equation with several terms.[27, 28, 44, 76, 80] Hafezi and Newman provide guidance for choosing appropriate polynomial fit equations.[76] The data in Figure 30 are best described by a linear fit (shown in Figure 30 as the dashed-green line). Polynomial fits using a basis of ln(c) were not found to describe the data better than the linear fit, and oscillatory fit results are apparent even in a third-order polynomial fit equation. Following work by Hafezi,[85] an equation of the following type was also used:

$$\left(e^{U_{\Delta c}}\right)^n = \left(e^{a_1+a_2\ln(c)}\right)^n + \left(e^{a_3+a_4\ln(c)}\right)^n \qquad [25]$$

where a_1, a_2, a_3, a_4, and n are fit parameters. Using the built-in Levenberg-Marquardt algorithm in Igor Pro, the best fit to the data in Figure 30 gave $a_1 = -0.83132$, $a_2 = -0.23206$, $a_3 = 0.02490$, $a_4 = -0.17425$, and $n = 3.5164$. This fit is shown in Figure 30 as the solid-red line. Remarkably, the fit in Equation 25 overlies the linear fit almost exactly. This result further suggests that the linear fit is the best description of this data set. Furthermore, the linear fit matches the data particularly well in the region around $r = 0.085$. The sign on $\frac{d\ln(c)}{dU_{\Delta c}}$ is negative, as plotted in Figure 29 and Figure 30, but the sign of the measurement signal depends on the way in which the measurement electrodes were connected to the concentration-cell. In the original work by Ma et al., the calculated transference numbers can be reproduced if $\frac{-d\ln(c)}{dU_{\Delta c}}$ is used as the input parameter. Later studies reversed the instrument connections to obtain $\frac{d\ln(c)}{dU_{\Delta c}}$ values that were inherently positive.[28, 80] The instrument configuration is unimportant so long as the connections are consistently made. By definition, $\frac{d\ln(c)}{dU_{\Delta c}}$ is positive. This assertion is apparent from Equation 26. In this work, the instrument connections were backwards. Thus, a value of $\frac{-d\ln(c)}{dU_{\Delta c}}$ was used to calculate transference numbers.

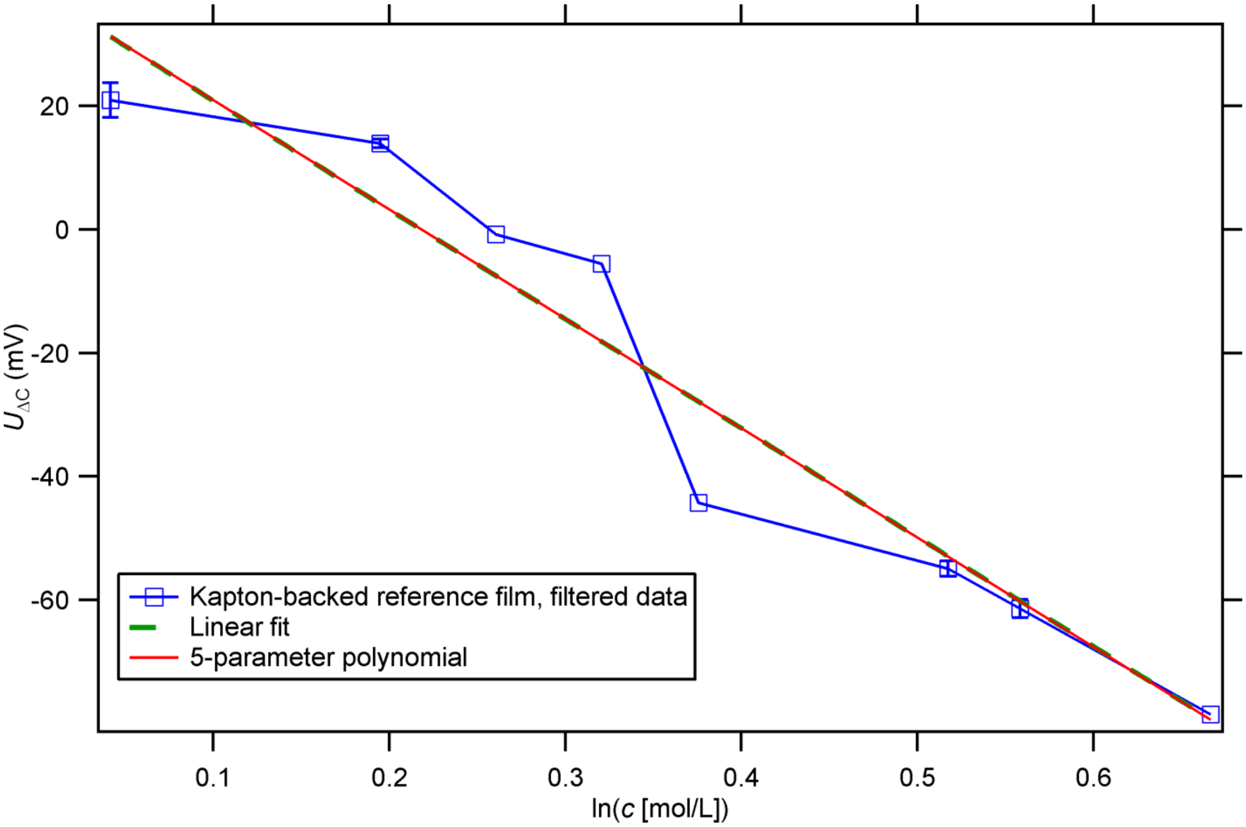


Figure 30 – Filtered concentration-cell results for the Kapton-backed film in Figure 29. The reference film was Kapton-backed and had a concentration of $r = 0.085$. The results at $r = 0.020$ and $r = 0.080$ are excluded for reasons described in the text. All data are at 90 °C. The lines drawn between data points serve as a visual guide. The straight lines represent best fits to the data. The dashed-green line is a linear fit, and the solid-red line is a fit to Equation 25. The two fit results match closely, indicating that a linear fit is the most appropriate.

The lithium ion transference number can now be calculated using Equation 18. A value of 8.45 × 10^{-8} cm^2/s was used for D at 90 °C and r = 0.085, as calculated using the methods to determine D_{avg} in Chapter 2. From the slope of the linear fit, we calculate $\frac{-d\ln(c)}{dU_{\Delta c}}$ = 0.00565 mV^{-1}. The value of c_∞ at r = 0.085 is 1.606 mol/L, and m = 15.48 $\frac{\mathrm{mV\,cm^2}}{\mathrm{mA\,s^{1/2}}}$, as determined above. From these input parameters, t_+^0 = -0.75.

Negative values of t_+^0 are physically possible, and have been reported in several polymer electrolyte systems under various conditions.[27, 78, 79] Aqueous zinc halide solutions are well known systems that unambiguously exhibit negative transference numbers.[86] The physical interpretation of a negative transference number is that Li^+ forms mobile moieties with the anion, such as a triplet of the form $\mathrm{Li(TFSI)_2^-}$. Triplets of this kind can pull some of the Li^+ in the same direction as $TFSI^-$ when the triplet mobility is higher than that of "free" Li^+. According to Ma et al,[27] if the species consist of Li^+ and $\mathrm{Li(TFSI)_2^-}$, $t_+^0 \geq -1$ must be true. This constraint holds for any speciation model which allows only Li^+, $TFSI^-$ and $\mathrm{Li(TFSI)_{k+1}^{k-}}$, where k is an integer and k ≥ 1. However, for a system containing moieties of the type $\mathrm{Li_k(TFSI)_{k+1}^-}$, negative transference numbers of arbitrarily large magnitudes are possible because some moieties can carry multiple cations with a single, negatively-charged species. The experiments in this Chapter are insufficient to validate any speciation model, although the negative transference numbers suggest the presence of mobile triplets or higher-order moieties of some variety. For the purposes of understanding the macroscopic transport properties of any binary electrolyte, t_+^0 as calculated here is sufficient to model cell performance. The negative values of t_+^0 for SEO indicate that the local-scale thermodynamic interactions are different from those in PEO. This suggestion is quite reasonable, considering that previous studies have suggested other differences between PEO/LiTFSI and SEO/LiTFSI systems. For example, in Chapter 2 we reported D_{avg} data for SEO/LiTFSI systems that were significantly different from those of PEO/LiTFSI due to the SEO microstructure. Previous studies have also suggested that SEO/LiTFSI thermodynamic interactions are different from those of PEO/LiTFSI due to local-scale perturbations in the polymer chain configurations induced by the polymer microstructure.[9, 19, 20]

Given the sensitivity of Equation 18 to $\frac{d\ln(c)}{dU_{\Delta c}}$, it is prudent to make further comparisons between the data in Figure 30, theoretical predictions, and literature results for PEO/LiTFSI under similar conditions. These comparisons are shown in Figure 31. The ideal results were calculated using Equation 26, which can be derived from Equation 12.49 in Newman and Thomas-Alyea[34], and is given by Equation 5.20 in Hafezi[85] and similarly in Equation 13 by Georen and Lindbergh,[80] assuming that the activity coefficient is unity. This equation is given as

$$U_{\Delta c} = \frac{2RT}{F}(1 - t_+^0)\ln(\frac{c}{c_{ref}}) \qquad [26]$$

where R is the ideal gas constant and t_+^0 is assumed constant with c. Equation 26 can be determined from Equation 19, assuming the activity coefficient and t_+^0 are constant. Figure 31 shows two ideal curves, one of which was calculated assuming a value of t_+^0 = 0.40, as found in previous work for PEO/LiTFSI under similar conditions.[28] The other ideal curve was calculated assuming a value of t_+^0 = -0.75, as determined from the linear fit in Figure 30, as described above. Equation 26 inherently gives a curve that has $U_{\Delta c} = 0$ when $c = c_{\mathrm{ref}}$. In this analysis we are concerned with the value of $\frac{-d\ln(c)}{dU_{\Delta \mathrm{c}}}$, not $U_{\Delta c}$; hence the concentration-cell data were shifted vertically upwards by 45.2 mV so that the linear fit shown in Figure 30 satisfies the condition that $U_{\Delta c} = 0$ at $c = c_{\mathrm{ref}}$. The line calculated with $t_+^0 = 0.40$ deviates substantially from the concentration-cell data, and $\frac{-d\ln(c)}{dU_{\Delta \mathrm{c}}}$ for that ideal curve is similar to that of the concentration-cell data only towards lower concentrations. The line calculated with $t_+^0 = -0.75$ more accurately captures the magnitude and slope of the concentration-cell data.

Under the assumption of unity activity coefficients, t_+^0 can be calculated directly from $\frac{-d\ln(c)}{dU_{\Delta \mathrm{c}}}$, for which the value 0.00565 mV^{-1} as determined above gives $t_+^0 = -0.91$. The magnitude of this datum agrees with that calculated for $t_+^0 = -0.75$ to better than 8% across the entire range shown in Figure 31, and thus is not shown for clarity. From these idealized predictions, it is clear that the concentration-cell data more closely match the slope and magnitude in the case $t_+^0 = -0.75$. In the ideal case, $f_\pm = 1$ and does not vary with concentration. However, polymer electrolyte systems often contain thermodynamic factors (defined as $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$, and calculated from Equation 19) that vary strongly with concentration and deviate substantially from unity.[27, 28, 79, 80] Using the calculated value $t_+^0 = -0.75$, $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right) = 1.62$ for the SEO(74-98) with $r = 0.085$ at 90 °C. This factor was calculated for other PEO-based systems by Georen and Lindbergh.[80] From their data, at values around $c = 1.6$ M (corresponding to $r = 0.085$ for PEO/LiTFSI), they found PEMO/LiTFSI at 25 °C gave a value of approximately 1.1, PEO/NaTFSI at 85 °C gave a value of approximately 2.7, and PEO/LiTFSI at 85 °C gave a value of approximately 3. In each of these systems, $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$ increased with increasing c. The opposite trend was reported by Ma et al. for the PEO/NaTf system.[27] It is difficult to analyze these data further without having a complete set vs. c, other than to note that the calculated value of $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$ is within a reasonable range, and is closer to the ideal value of 1 than that found for PEO/LiTFSI under similar conditions.

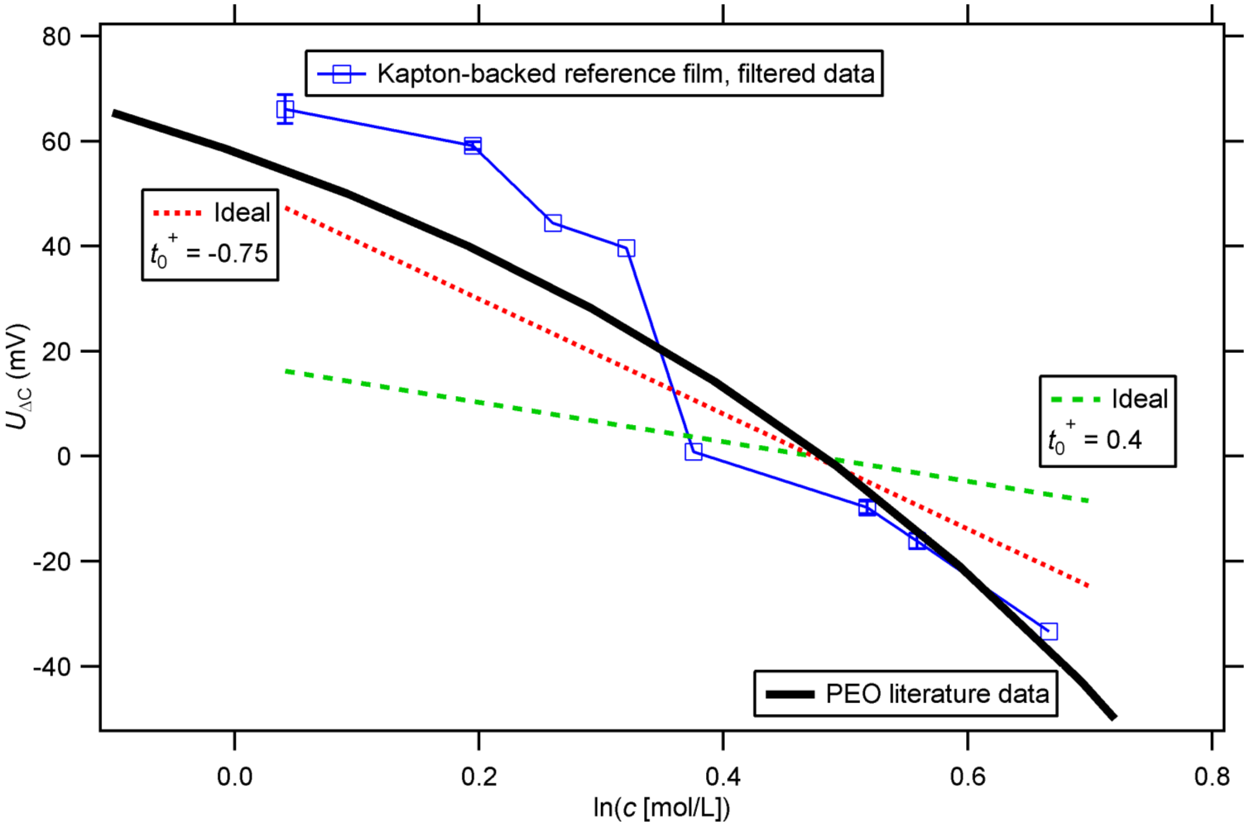


Figure 31 – Concentration-cell data compared to ideal solutions and PEO literature.
The ideal predictions were calculated assuming unity activity coefficients according to Equation 26. The PEO data are for r = 0.083 at 85 °C, as reported by Edman et al.[28] The PEO data were shifted vertically upwards by 8 mV so that $U_{\Delta c}$ = 0 at r = 0.085. The concentration-cell data were shifted vertically upwards by 45.2 mV such that the linear fit to the data meets the same criterion.

Figure 31 also shows the concentration cell curve for PEO/LiTFSI at r = 0.083 and 85 °C.[28] This curve was shifted vertically upwards by 8 mV so that $U_{\Delta c}$ = 0 at r = 0.085. Across the concentration range shown, the PEO data match the trend and magnitude of the SEO/LiTFSI concentration-cell data better than either of the ideal assumptions. These data suggest that, to a first approximation, the SEO/LiTFSI systems are not drastically different from the PEO/LiTFSI systems, and that the presence of nonunity thermodynamic factors help capture the trends and magnitudes of the observed concentration-cell data. It is notable that the value of $U_{\Delta c}$ can be predicted from the integration of Equation 19. Larger values of $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$ and smaller values of t_+^0 give rise to larger values of $U_{\Delta c}$. The PEO/LiTFSI system exhibits a larger thermodynamic factor and a larger transference number than the corresponding SEO/LiTFSI system. One explanation for the similarity of the PEO and SEO concentration-cell data in Figure 31 is that these factors cancel to some extent.

The diffusion coefficient has thus far been overlooked in this discussion. The appropriate value of D to use for SEO systems is unclear, as detailed in Chapter 2. Here, we used the value of D_{avg} that was extracted from the moments of a distribution function for SEO(74-98) with r = 0.085 at 90 °C. In lieu of further analysis along the lines of that in Chapter 2, this appears to be the best choice for the effective diffusion coefficient. However, we observed in Chapter 2 that our measured value of D for PEO/LiTFSI ($1.23 \pm 0.19 \times 10^{-7}$ cm^2/s) was much larger than that reported for similar PEO/LiTFSI systems by Edman et al. (4.65×10^{-8} cm^2/s)[28] and Geiculescu et al. (4.2×10^{-8} cm^2/s).[47] We have no explanation for this discrepancy, other than to note that it cannot be fully explained by temperature and composition differences. This degree of uncertainty in the measurement can have a strong effect on the calculated transference number. In Chapter 2, we found that for high-molecular-weight SEO samples such as the SEO(74-98) used here, $D_{\mathrm{avg}} \approx (2/3)\, D_{\mathrm{PEO}}$, where D_{PEO} is the value of D measured for PEO. Thus, using a value of 2.9×10^{-8} cm^2/s, we recalculate the values of t_+^0 and $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$ as -0.29 and, 2.19, respectively. These values are closer to those reported for PEO, as described above, but even this large uncertainty in D is insufficient to explain the negative transference number for SEO. Within the discernible experimental error, the transference number for SEO is negative.

5.4 Conclusions

The transference number of SEO(74-98) at $r = 0.085$ and 90 °C was calculated using the rigorous method outlined by Ma et al.[27] This method requires 4 experimental quantities: D, m, $\frac{d\ln(c)}{dU_{\Delta c}}$, and c_∞. For block-copolymer electrolytes, D is not well defined, but D_{avg}, as defined in Chapter 2, was used as an approximation. Current-interrupt experiments were used to determine m. Individual samples gave excellent linear responses to applied current, but sample-to-sample variation gave relative error on the order of 20% for m. The reason for this is unclear, and m was determined from the average values of all data points at a given set of experimental conditions, the relative error of which was of order 1%. The concentration-cell experiments used to determine $\frac{d\ln(c)}{dU_{\Delta c}}$ had significant levels of noise, particularly in the region of interest around $r =$ 0.085. The noise persisted even with a free-standing reference film that attained good contact between the various cell components. The best estimate of $\frac{d\ln(c)}{dU_{\Delta c}}$ was obtained from a linear fit to the concentration-cell data. Finally, c_∞ was calculated from r using the assumption of conservation of molar volume during electrolyte mixing. The calculated value of t_+^0 was -0.75, and that of the thermodynamic factor, $\left(1 + \frac{d\ln f_\pm}{d\ln(c)}\right)$, was 1.62.

These results are interpreted as arising from ion-complexation reactions that give rise to mobile species, a hypothesized example of which is $\mathrm{Li(TFSI)}_2^-$. The electrochemical methods used in this Chapter relate only to macroscopic transport observations and have no dependence on the details of ion speciation, although the presence of mobile-ion complexes is necessary to explain the negative transference number. The largest source of error in the transference-number calculation arose from the concentration-cell experiment. The most notable issue was the presence of $U_{\Delta c}$ at regions with $c \ll c_{\mathrm{ref}}$. Despite these problems, the data are still relevant due to the dependence on $\frac{d\ln(c)}{dU_{\Delta c}}$ rather than $U_{\Delta c}$. By scaling the concentration-cell data appropriately, the results were compared to idealized concentration-cell predictions (unity thermodynamic factor), assuming various transference numbers, and to concentration-cell data for an analogous PEO/LiTFSI system. These comparisons suggest that the parameters calculated for the SEO/LiTFSI data cannot be described by errors in the concentration-cell experiment alone, and that the intrinsic nature of PEO/LiTFSI channels confined within SEO-based electrolytes must be different from bulk PEO/LiTFSI.

A complete concentration dependence of t_+^0 depends most crucially on the concentration-cell experiment. If the concentration-cell experiment can be performed with higher levels of precision, the data in Figure 29 must be generated only once for a given polymer. Once those data are secured, the values of D_{avg} and m can be much more readily determined for a given salt concentration. A complete trend of t_+^0 vs. c for a given SEO system would provide significantly more data for analysis, especially relative to analogous PEO systems, and would be useful for modeling full-cell battery behavior.

The calculation to determine t_+^0 is also sensitive to D, m, and c_∞; each of which has various built-in complications, caveats, and relatively large experimental errors. These parameters are easier to define and measure in unstructured electrolytes such as PEO. In particular, the value of D is not well known for SEO, although we showed in this work that the use of different values of D is insufficient to match the SEO and PEO transference numbers and thermodynamic factors.

A broader goal in the study of polymer electrolytes should be to find an alternative approach to determining transference numbers. Perhaps the most fruitful approach would be to find a new method by which to measure the mean molar activity coefficient. The transference number can then be rigorously and accurately determined if the activity coefficient of the electrolyte is well known; thus the concentration-cell experiment would be obviated and the entire process could be executed more readily.

At present, the concentration-cell method is the only rigorous method by which transference numbers can be found for polymer electrolytes. From this study, it is apparent that several improvements to the concentration-cell technique are possible. Suggested improvements to the concentration-cell experiment include: using large, free-standing films to obtain good electrode and inter-film diffusional contact, using fine increments in salt concentration around regions of interest, and using a reference film with a low salt concentration such that complications that occur when using a concentrated reference electrolyte are avoided.

Chapter 6 – Current-Induced Formation of Gradient Crystals

ABSTRACT

Conventional ordered phases such as crystals and liquid crystals have constant domain spacings. In this study we report on the formation of coherently ordered morphologies wherein the domain spacing changes continuously along a specified direction. We have coined the term "gradient crystal" to refer to this structure, a signature of which is a small angle X-ray scattering pattern that resembles a sundial. Gradient crystals composed of a gyroid morphology form spontaneously when ionic current is driven through a block-copolymer electrolyte. We propose that this structure forms because it allows for a continuous change in domain spacing without requiring the introduction of defects. Previous studies have shown that applied electric fields ranging from 1,000 to 40,000 V/mm can induce long-range structural order, alignment, and morphological transitions in block copolymers. Gradient crystals form under applied electric fields as low as 2.5 V/mm due to the presence of direct ionic currents that are absent in the aforementioned studies.

6.1 Introduction

Studies involving the passage of ionic current through nanostructured polymers are motivated by their use as ion conducting membranes for applications such as battery electrolytes and fuel cell membranes.[20, 21, 87, 88] The equilibrium morphologies of these materials have constant domain spacings – a conventional characteristic of crystalline and liquid-crystalline ordered phases.[89, 90] While it may be anticipated that direct ionic current (dc) passage could alter the morphologies of polymer electrolytes, there are no experimental reports that address this issue. Previous work has shown that electric fields can induce long-range structural order [91-94] and morphological transitions [95-97] in ion-free block copolymers. Efforts to understand these effects have focused on the dielectric contrast between the constituent polymer blocks which gives rise to small energetic penalties for certain microstructures and microstructural orientations in an applied electric field. Typical electric field strengths needed to affect morphologies range from 1,000 to 40,000 V/mm.[91-95] Theoretical work by Tsori et al. indicated that the presence of solvated ions in one of the polymer blocks would enhance dielectric contrast and enable morphological-transitions at lower field strengths.[98, 99] Kohn et al. demonstrated this hypothesis by showing that mobile ions enabled alignment at lower field strengths, although ion-blocking electrodes and fields above 4,000 V/mm were used.[100] In this work we demonstrate the formation of large crystals composed of a gyroid morphology within which the domain spacing changes continuously. We refer to these structures as “gradient crystals” and attribute their formation to the concentration gradients that develop during dc ionic current passage and the associated constraints those concentration gradients impose upon block-copolymer self-assembly. These structures form with applied electric fields ranging from 2.5 to 15 V/mm.

6.2 Methods

SEO/LiTFSI electrolytes with selected r_0 values were pressed into insulating spacers made of Garolite G-10 with inner diameters of 3.86 mm to obtain sample thicknesses ranging from 200 to 400 µm. Small-angle X-ray scattering (SAXS) experiments under quiescent conditions were performed on samples contained within sealed cells with Kapton windows as described previously.[24] Symmetric cells for *in situ* experiments were assembled by placing 150 µm-thick lithium foils (FMC Lithium) on both sides of the electrolyte-containing spacers. Nickel tabs were taped to the outside of the foils, and the entire cells were vacuum sealed inside of air-tight pouches (Showa-Denko). All of the steps from mixing the salt and polymer through sealing the pouches were conducted in argon-filled gloveboxes without exposure to ambient air (water and oxygen levels < 1 ppm). *In situ* small-angle X-ray scattering (SAXS) measurements were performed at beamline 7.3.3 at the Advanced Light Source.[101] The sealed cells were loaded into a custom heating stage, and the tabs were connected to a potentiostat (Solartron 1470E or Bio-Logic VMP3). Fixed currents were applied to each cell, and the cell potentials were always $\leq$ 3 V. X-rays passed through the center of the sample in a direction parallel to the direction of ionic current, and 2D scattering patterns were recorded over time. 1D scattering profiles were obtained by azimuthally-averaging 2D scattering patterns[33] and are presented as scattering intensity versus magnitude of the scattering wave vector q, defined as $q = 4\pi\lambda^{-1}\sin(\theta/2)$, where λ is the wavelength of the X-rays (0.124 nm) and θ is the scattering angle. The primary scattering vector, q^*, is the value of q corresponding to the {100} family of reflection planes for a given morphology. The domain spacing, *d*, is calculated from the scattering peak at the lowest observed q value, q', as $d = 2\pi/q'$. The q' peak corresponds to the correlation length in the disordered morphology, the {100} planes for the lamellar morphology, and the {211} planes for the gyroid morphology.

6.3 Results and Discussion

The equilibrium phase behavior of SEO/LiTFSI electrolytes as a function of temperature and salt concentration was obtained in the range of 25 to 180 °C by SAXS, and the results are shown in Figure 1. At room temperature, the electrolytes exhibit disordered morphologies in the range $0 \leq r_0 \leq 0.030$, lamellar morphologies in the range $0.050 \leq r_0 \leq 0.170$, and gyroid morphologies in the range $0.190 \leq r_0 \leq 0.250$. Samples with r_0 ranging from 0.050 to 0.125 exhibit accessible order-to-disorder transitions (ODTs). The ODT temperatures increase with increasing r_0, as shown on the left axis in Figure 32. Samples with $r_0 \geq 0.150$ do not exhibit accessible ODTs within our temperature window. Domain spacings (*d*-spacings) at a fixed temperature of 80 °C are shown on the right axis of Figure 32. Lamellar samples exhibit *d*-spacing values that increase linearly with increasing r_0 in the range $0.050 \leq r_0 \leq 0.170$.

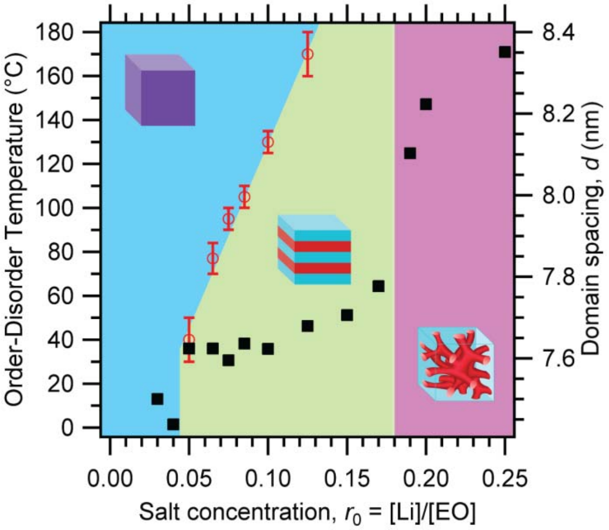


Figure 32 – Phase diagram of SEO(1.7-1.4).
The equilibrium salt concentrations (r_0 = [Li]/[EO]) was varied across a wide range, as shown on the bottom axis. Some samples had accessible temperature-induced order-disorder transitions, as shown on the left axis (○). Domain spacings, which were based on primary scattering peaks at 80 °C (■), increase monotonically with increasing salt concentration (right-axis). Error bars in the measured order -disorder transition temperatures reflect the size of the temperature step that resulted in the transition.

The morphology of the electrolytes in the presence of direct ionic current (dc) was monitored by *in situ* SAXS as shown in Figure 33a. The electrochemical processes that occur during the experiment are shown schematically in Figure 33b. Current passage causes the development of salt concentration gradients by well-known processes.[34] This leads to a salt concentration, $r(t,z)$, that depends on both time, t, and position across the cell, z as shown in Figure 33b. The origin of the Cartesian coordinate axis (z = 0) is defined as the location of the cathode/electrolyte interface. For our cell geometry we expect variations in r only along z. The measured SAXS profiles contain scattering contributions from the electrolyte at all r values that existed in the cell during image acquisition. If the salt diffusion coefficient and Li^+ transference numbers are independent of salt concentration, the steady-state salt concentration profile across the cell is linear,[34] and the maximum concentration gradient is obtained at limiting current, under which salt concentrations at the boundaries $z = 0$ and $z = L$ are 0 and $2r_0$, respectively (L is the thickness of the electrolyte). These concentration limits can then be used in combination with the data in Figure 32 to predict the evolution of structures in a cell. Concentration gradients have associated concentration overpotentials that are sensed as a contribution to the cell voltage during a charging experiment.[34] For a fixed current density, the cell voltage increases continuously over time as the concentration profile develops. For a fixed cell potential, the current density decreases slowly over time after a fast initial rise. In all cells the concentration gradients develop slowly and steady-state operation is never reached, which indicates that the concentration gradients are never fully developed. We thus expect our cells to contain concentration profiles that are qualitatively similar to those shown at intermediate times in Figure 33b.

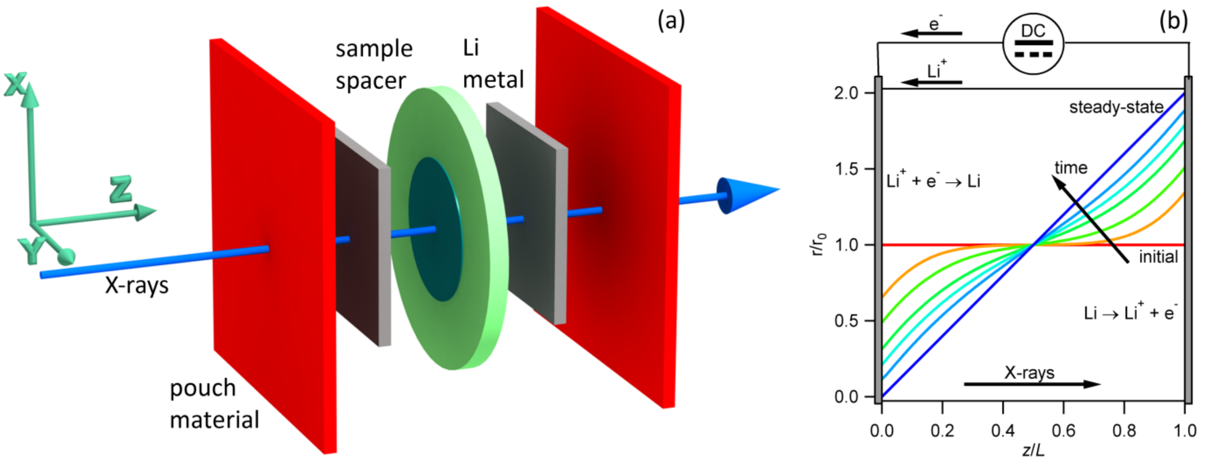


Figure 33 – *In situ* SAXS experimental schematics.
(a) Schematic of *in situ* SAXS experiment. Symmetric Li | block-copolymer electrolyte | Li cells are sealed inside a pouch and connected to an external potentiostat for electrochemical control. Incident X-rays pass through the entire cell assembly. (b) Symmetric cell schematic showing the development of salt concentration profiles when dc current is passed through the cell. $z = 0$ is defined to be at the cathode/electrolyte interface. Salt concentration, r, is normalized by the average (equilibrium) concentration, r_0, and distance, z, is normalized by the distance between electrodes, L.

Figure 34a shows the time dependence of the measured one-dimensional (1D) SAXS profiles for an electrolyte with $r_0 = 0.085$ at 135 °C under a fixed current density of 1.06 mA/cm^2. The inset graphics show the expected morphologies corresponding to the scattering profiles at each time point. At equilibrium, the sample exhibits a lamellar morphology at room temperature and an ODT at 110 °C (see Figure 32). The sample was initially equilibrated at 135 °C and exhibited the disordered morphology. Following the arguments above, we expect the salt concentrations to be within the range $0 \leq r \leq 0.170$, corresponding to coexistence between lamellar and disordered morphologies (Figure 32). The passage of current causes the rapid formation of a sharp peak that grows with time and dominates the SAXS profile within 12 min. Higher-order peaks are not apparent, indicating the lack of large coherent grains. The reason for this is clarified below. The ordered-phase morphology is presumed to be lamellar because the expected salt concentration profile at small times will produce lamellar regions and the peak position and width are similar to those associated with the equilibrium lamellar phase. The ordered peak broadens with time, as seen in the low-q shoulder of the 32 min SAXS profile. This is expected due to the establishment of a salt concentration gradient across the cell which leads to a gradient in d-spacing in accordance with the data in Figure 32 and Figure 33b. The 32 min data are consistent with the presence of small lamellar grains with d-spacings in the range 7.6 to 8.1 nm. The local d-spacing is expected to increase with increasing z. At 48 min the SAXS profile is abruptly punctuated by Bragg spots, and the low-q shoulder decreases in intensity relative to the profile at 32 min. The Bragg spots index to the gyroid morphology,[102, 103] for which the reflections $q/q^* = \sqrt{6}$, $\sqrt{8}$, and $\sqrt{16}$ are indicated by arrows in Figure 34a. Since few spots are observed, the scattering must originate from large, spatially-coherent grains with the gyroid morphology. Current-induced formation of Bragg spots indexing to the gyroid morphology was observed under a wide variety of experimental conditions, including applied current densities ranging from 0.74 to 1.28 mA/cm^2, applied electric fields ranging from 2.5 to 15 V/mm, and r_0 values of 0.065, 0.075, and 0.085. Samples charged at slower rates tended to form more intense Bragg spots. Figure 34b shows the 1D profile of a sample after 65 min under 0.74 mA/cm^2 applied current density. The reflections at q/q^* = √6, √8, √14, √16, √20, and √22 are in excellent agreement with those for the r_0 = 0.200 sample that exhibits the gyroid morphology at equilibrium.

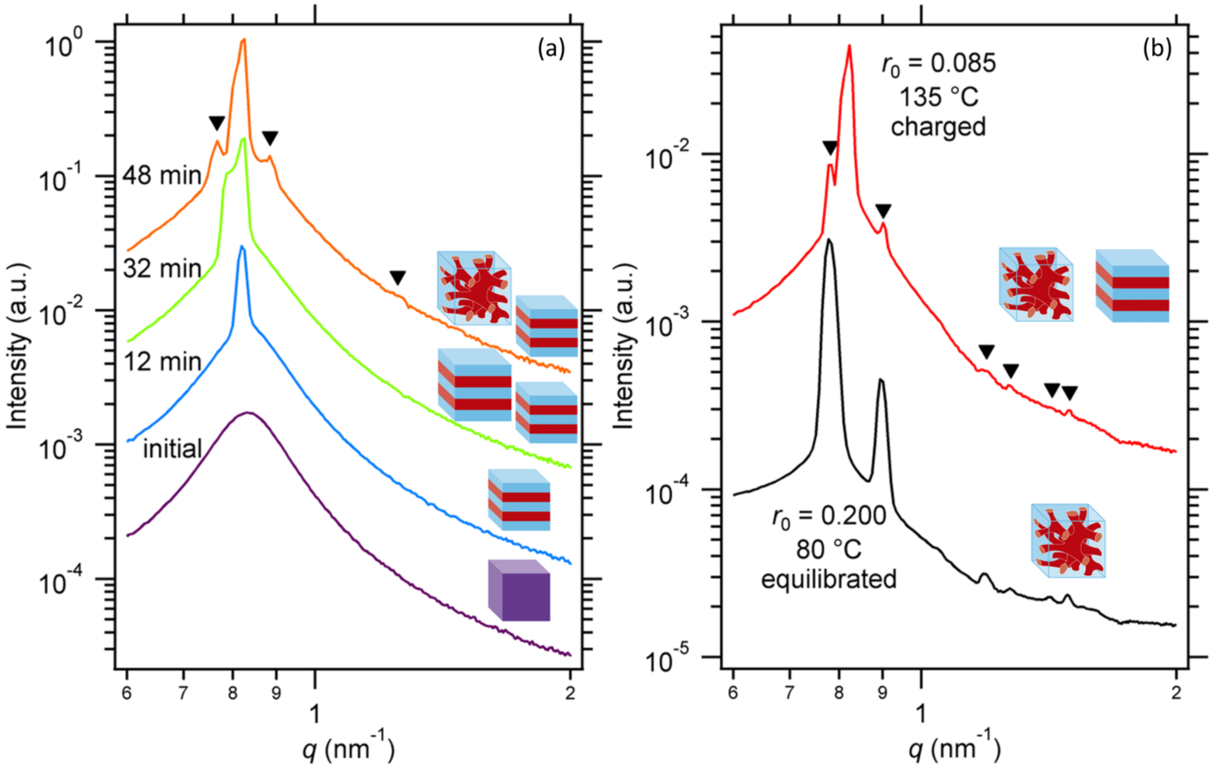


Figure 34 – Formation of Bragg spots during *in situ* charging.
(a) 1D SAXS profiles of electrolyte with r_0 = 0.085 at 135 °C under 1.06 mA/cm^2 applied current at various times (offset vertically for clarity). The initial profile is shown before the current was applied, and subsequent profiles were taken after application of current with elapsed times as indicated. Inset schematics illustrate the sample morphology as determined by the corresponding SAXS profiles. The arrows indicate Bragg reflections at q/q^* = √6, √8, and √16. (b) 1D SAXS profile of electrolyte with r_0 = 0.085 at 135 °C at 65 min under 0.74 mA/cm^2 applied current compared with that of an equilibrated electrolyte with r_0 = 0.200 at 80 °C. The arrows indicate Bragg reflections at q/q^* = √6, √8, √14, √16, √20, and √22.

Further morphological changes are observed with larger applied electric fields (or currents) and longer charging times. The application of a 15 V/mm electric field on an electrolyte with r_0 = 0.085 at 135 °C resulted in the formation of Bragg spots indexing to gyroid reflections at q/q^* = √6 and √8 within 13 min (not shown for brevity). With time, the spots stretched radially towards lower q-values in the form of streaks with constant azimuthal angles to give a scattering pattern that resembles a sundial. An example of such a SAXS pattern, obtained at 46 min, is shown in Figure 35a. It indicates the coexistence of isotropically-distributed lamellar grains (the ring) and a few gyroid "crystals" with a systematic increase (gradient) in d-spacing. We have coined the term "gradient crystal" to define such an ordered phase within which the d-spacing is position-dependent but the orientation is not. A schematic of a gradient crystal is shown in Figure 35b. Each gradient crystal will give sets of streaks depending on the orientation of the crystal in the beam. The radial streaks have a sharply defined q-range in the scattering plane. For the sample in Figure 35a, the lowest and highest *d*-spacings in the gradient crystals are 8.1 and 8.8 nm. The largest *d*-spacings of the streaks are well outside the range observed under equilibrium conditions (Figure 32).

The development of large gyroid gradient crystals from a lamellar phase with little long range order is surprising. For a bulk concentration of $r_0 = 0.085$ the maximum expected salt concentration under applied current is $r = 0.170$, at which concentration the equilibrium morphology is unambiguously lamellar (Figure 32 and Figure 34b). We propose that gyroid gradient crystals form due to constraints on self-assembly imposed by the passage of dc ionic current. The electrochemical reactions at the electrodes create concentration gradients which, at steady state, would lead to a continuous change in *d*-spacing along *z* in accordance with the data in Figure 32. In other words, dc current is expected to result in the formation of wedge-shaped grains, oriented with the bigger end nearer the anode (larger *r* and *d*) and the smaller end nearer the cathode (smaller *r* and *d*). Wedges of this kind composed of a one-dimensional periodic structure cannot fill three-dimensional space effectively and thus a large number of defects must be generated in order to accommodate them, i.e., there is a mismatch between the dimensionality of the lamellar phase and that of the space that it must occupy. The absence of higher-order peaks in the SAXS profiles of lamellae formed under the influence of ionic current (Figure 34a) is attributed to this effect. In contrast, a cubic structure like the gyroid phase can simultaneously accommodate an increase in *d*, required by the salt concentration gradient, and fill space without the introduction of defects. While the SAXS data provide information about morphological distortions in the *xy*-plane only, it is clear that current in the *z*-direction is responsible for this. We thus expect distortions in all three directions as shown schematically in Figure 35b.

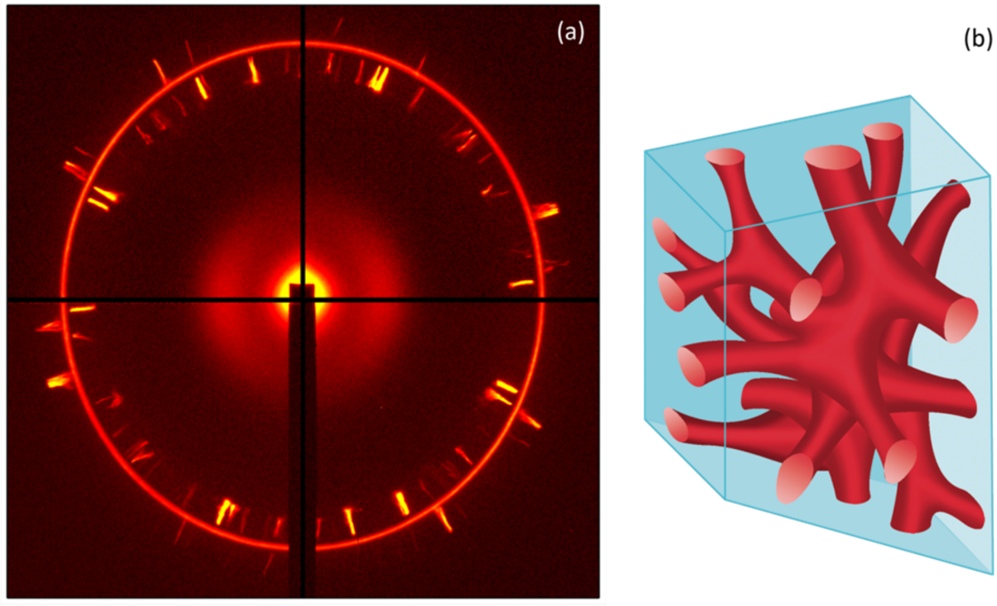


Figure 35 – Sundial scattering pattern and gyroid gradient crystal.
(a) Two-dimensional SAXS pattern of electrolyte with $r_0 = 0.085$ at 135 °C after 46 minutes under an applied electric field of 15 V/mm. The beam center is located at the center of the pattern. The bright continuous ring is due to isotropically-distributed lamellar grains. Bright streaks emanating from the ring are due to gradient crystals composed of the gyroid morphology. The black bars are due to the beamstop and detector gaps. The low-*q*

scattering lobes close to the beam center are background contributions from the pouch. (b) Schematic of a gyroid gradient crystal.

6.4 Conclusions

In summary, we show that the application of electric fields on self-assembled electrolytes can result in the formation of spatially coherent ordered phases with a continuous change in domain spacing. The lack of translational order, a common characteristic of traditional materials, may lead to interesting physical properties.

References

1. Monroe, C.; Newman, J. *Journal of the Electrochemical Society* **2005,** 152, (2), A396-A404.

2. Shi, J.; Vincent, C. A. *Solid State Ionics* **1993,** 60, (1-3), 11-17.

3. Kang, Y. K.; Cheong, K.; Noh, K. A.; Lee, C.; Seung, D. Y. *Journal of Power Sources* **2003,** 119, 432-437.

4. Tatsuma, T.; Taguchi, M.; Oyama, N. *Electrochimica Acta* **2001,** 46, (8), 1201-1205.

5. Zhang, X. W.; Li, Y. X.; Khan, S. A.; Fedkiw, P. S. *Journal of the Electrochemical Society* **2004,** 151, (8), A1257-A1263.

6. Arbizzani, C.; Mastragostino, M.; Hamaide, T.; Guyot, A. *Electrochimica Acta* **1990,** 35, (11-12), 1781-1785.

7. Cho, B. K.; Jain, A.; Gruner, S. M.; Wiesner, U. *Science* **2004,** 305, (5690), 1598-1601.

8. Giles, J. R. M.; Gray, F. M.; Maccallum, J. R.; Vincent, C. A. *Polymer* **1987,** 28, (11), 1977-1981.

9. Gomez, E. D.; Panday, A.; Feng, E. H.; Chen, V.; Stone, G. M.; Minor, A. M.; Kisielowski, C.; Downing, K. H.; Borodin, O.; Smith, G. D.; Balsara, N. P. *Nano Letters* **2009,** 9, (3), 1212-1216.

10. Gray, F. M.; Maccallum, J. R.; Vincent, C. A.; Giles, J. R. M. *Macromolecules* **1988,** 21, (2), 392-397.

11. Hirahara, K.; Takano, A.; Yamamoto, M.; Kazama, T.; Isono, Y.; Fujimoto, T.; Watanabe, O. *React. Funct. Polym.* **1998,** 37, (1-3), 169-182.

12. Jannasch, P. *Chem. Mat.* **2002,** 14, (6), 2718-2724.

13. Khan, I. M.; Fish, D.; Delaviz, Y.; Smid, J. *Makromolekulare Chemie-Macromolecular Chemistry and Physics* **1989,** 190, (5), 1069-1078.

14. Kishimoto, K.; Yoshio, M.; Mukai, T.; Yoshizawa, M.; Ohno, H.; Kato, T. *J. Am. Chem. Soc.* **2003,** 125, (11), 3196-3197.

15. Kosonen, H.; Valkama, S.; Hartikainen, J.; Eerikainen, H.; Torkkeli, M.; Jokela, K.; Serimaa, R.; Sundholm, F.; ten Brinke, G.; Ikkala, O. *Macromolecules* **2002,** 35, (27), 10149-10154.

16. Niitani, T.; Shimada, M.; Kawamura, K.; Dokko, K.; Rho, Y. H.; Kanamura, K. *Electrochemical and Solid State Letters* **2005,** 8, (8), A385-A388.

17. Niitani, T.; Shimada, M.; Kawamura, K.; Kanamura, K. *Journal of Power Sources* **2005,** 146, (1-2), 386-390.

18. Ohtake, T.; Ogasawara, M.; Ito-Akita, K.; Nishina, N.; Ujiie, S.; Ohno, H.; Kato, T. *Chem. Mat.* **2000,** 12, (3), 782-789.

19. Panday, A.; Mullin, S.; Gomez, E. D.; Wanakule, N.; Chen, V. L.; Hexemer, A.; Pople, J.; Balsara, N. P. *Macromolecules* **2009,** 42, (13), 4632-4637.

20. Singh, M.; Odusanya, O.; Wilmes, G. M.; Eitouni, H. B.; Gomez, E. D.; Patel, A. J.; Chen, V. L.; Park, M. J.; Fragouli, P.; Iatrou, H.; Hadjichristidis, N.; Cookson, D.; Balsara, N. P. *Macromolecules* **2007,** 40, (13), 4578-4585.

21. Soo, P. P.; Huang, B. Y.; Jang, Y. I.; Chiang, Y. M.; Sadoway, D. R.; Mayes, A. M. *Journal of the Electrochemical Society* **1999,** 146, (1), 32-37.

22. Trapa, P. E.; Huang, B. Y.; Won, Y. Y.; Sadoway, D. R.; Mayes, A. M. *Electrochemical and Solid State Letters* **2002,** 5, (5), A85-A88.

23. Trapa, P. E.; Won, Y. Y.; Mui, S. C.; Olivetti, E. A.; Huang, B. Y.; Sadoway, D. R.; Mayes, A. M.; Dallek, S. *Journal of the Electrochemical Society* **2005,** 152, (1), A1-A5.

24. Wanakule, N. S.; Panday, A.; Mullin, S. A.; Gann, E.; Hexemer, A.; Balsara, N. P. *Macromolecules* **2009,** 42, (15), 5642-5651.

25. Wang, C. X.; Sakai, T.; Watanabe, O.; Hirahara, K.; Nakanishi, T. *Journal of the Electrochemical Society* **2003,** 150, (9), A1166-A1170.

26. Young, W. S.; Epps, T. H. *Macromolecules* **2009,** 42, (7), 2672-2678.

27. Ma, Y.; Doyle, M.; Fuller, T. F.; Doeff, M. M.; De Jonghe, L. C.; Newman, J. *Journal of the Electrochemical Society|Journal of the Electrochemical Society* **1995,** 142, (6), 1859-68.

28. Edman, L.; Doeff, M. M.; Ferry, A.; Kerr, J.; De Jonghe, L. C. *Journal of Physical Chemistry B* **2000,** 104, (15), 3476-3480.

29. Sax, J.; Ottino, J. M. *Polymer Engineering and Science* **1983,** 23, (3), 165-176.

30. Mullin, S. A.; Stone, G. M.; Panday, A.; Balsara, N. P. *Journal of the Electrochemical Society* **2011,** 158, (6), A619-A627.

31. Hamley, I. W.; Castelletto, V. *Prog. Polym. Sci.* **2004,** 29, (9), 909-948.

32. Roe, R. J., *Methods of X-ray and neutron scattering in polymer science*. Oxford University Press: New York, 2000; p xiv, 331 p.

33. Ilavsky, J. *Nika 2D Small Angle Scattering macro for Igor Pro*, 1.50; Argonne, 2011.

34. Newman, J. S.; Thomas-Alyea, K. E., *Electrochemical systems*. 3rd ed.; J. Wiley: Hoboken, N.J., 2004.

35. Newman, J.; Chapman, T. W. *Aiche Journal* **1973,** 19, (2), 343-348.

36. Thompson, S. D.; Newman, J. *Journal of the Electrochemical Society* **1989,** 136, (11), 3362-3369.

37. Pollard, R.; Comte, T. *Journal of the Electrochemical Society* **1989,** 136, (12), 3734-3748.

38. Edman, L.; Ferry, A.; Oradd, G. *Physical Review E* **2002,** 65, (4).

39. Oradd, G.; Edman, L.; Ferry, A. *Solid State Ionics* **2002,** 152, 131-136.

40. Nzikou, J. M.; Baklouti, M.; Vincent, L. M.; Lapicque, F. *Chemical Engineering and Processing* **1997,** 36, (2), 161-165.

41. Stewart, S. G.; Newman, J. *Journal of the Electrochemical Society* **2008,** 155, (1), F13-F16.

42. Umino, S.; Newman, J. *Journal of the Electrochemical Society* **1993,** 140, (8), 2217-2221.

43. Doeff, M. M.; Edman, L.; Sloop, S. E.; Kerr, J.; De Jonghe, L. C. *Journal of Power Sources* **2000,** 89, (2), 227-231.

44. Doeff, M. M.; Georen, P.; Qiao, J.; Kerr, J.; De Jonghe, L. C. *Journal of the Electrochemical Society* **1999,** 146, (6), 2024-2028.

45. Ferry, A. *Journal of Chemical Physics* **1997,** 107, (21), 9168-9175.

46. Ferry, A. *Journal of Physical Chemistry B* **1997,** 101, (2), 150-157.

47. Geiculescu, O. E.; Rajagopal, R.; Creager, S. E.; DesMarteau, D. D.; Zhang, X. W.; Fedkiw, P. *Journal of Physical Chemistry B* **2006,** 110, (46), 23130-23135.

48. Wang, Q.; Li, H.; Huang, X. J.; Chen, L. Q. *Journal of the Electrochemical Society* **2001,** 148, (7), A737-A741.

49. Thorat, I. V.; Stephenson, D. E.; Zacharias, N. A.; Zaghib, K.; Harb, J. N.; Wheeler, D. R. *Journal of Power Sources* **2009,** 188, (2), 592-600.

50. Onishi, L. M.; Prausnitz, J. M.; Newman, J. *The Journal of Physical Chemistry B* **2007,** 111, (34), 10166-10173.

51. Ilavsky, J. *Nika 2D SAS macro for Igor Pro*, 1.48; Argonne, 2010.

52. Bates, F. S.; Fredrickson, G. H. *Physics Today* **1999,** 52, (2), 32-38.

53. Helfand, E. *Macromolecules* **1975,** 8, (4), 552-556.

54. Helfand, E.; Wasserman, Z. R. *Macromolecules* **1976,** 9, (6), 879-888.

55. Provencher, S. W. Inverse Problems. http://www.s-provencher.com/ (Accessed November 2010),

56. Provencher, S. W. *Computer Physics Communications* **1982,** 27, (3), 213-227.

57. Provencher, S. W. *Computer Physics Communications* **1982,** 27, (3), 229-242.

58. Giurleo, J. T.; Talaga, D. S. *Journal of Chemical Physics* **2008,** 128, (11).

59. Lascaud, S.; Perrier, M.; Vallee, A.; Besner, S.; Prudhomme, J.; Armand, M. *Macromolecules* **1994,** 27, (25), 7469-7477.

60. Garetz, B. A.; Newstein, M. C.; Dai, H. J.; Jonnalagadda, S. V.; Balsara, N. P. *Macromolecules* **1993,** 26, (12), 3151-3155.

61. Chang, M. Y.; Abuzaina, F. M.; Kim, W. G.; Gupton, J. P.; Garetz, B. A.; Newstein, M. C.; Balsara, N. P.; Yang, L.; Gido, S. P.; Cohen, R. E.; Boontongkong, Y.; Bellare, A. *Macromolecules* **2002,** 35, (11), 4437-4447.

62. Sablani, S. S.; Goosen, M. F. A.; Al-Belushi, R.; Wilf, M. *Desalination* **2001,** 141, (3), 269-289.

63. Pinnau, I.; Koros, W. J. *Journal of Applied Polymer Science* **1991,** 43, (8), 1491-1502.

64. Gopinadhan, M.; Majewski, P. W.; Osuji, C. O. *Macromolecules* **2010,** 43, (7), 3286-3293.

65. Majewski, P. W.; Gopinadhan, M.; Jang, W. S.; Lutkenhaus, J. L.; Osuji, C. O. *J. Am. Chem. Soc.* **2010,** 132, (49), 17516-17522.

66. Naidu, S.; Ahn, H.; Gong, J.; Kim, B.; Ryu, D. Y. *Macromolecules* **2011,** 44, (15), 6085-6093.

67. Weber, R. L.; Ye, Y. S.; Schmitt, A. L.; Banik, S. M.; Elabd, Y. A.; Mahanthappa, M. K. *Macromolecules* **2011,** 44, (14), 5727-5735.

68. Wanakule, N. Thermodynamics and Ionic Conductivity of Block Copolymer Electrolytes. University of California, Berkeley, 2010.

69. Teran, A. A.; Tang, M. H.; Mullin, S. A.; Balsara, N. P. *Solid State Ionics* **2011,** 203, (1), 18-21.

70. Zugmann, S.; Fleischmann, M.; Amereller, M.; Gschwind, R. M.; Wiemhofer, H. D.; Gores, H. J. *Electrochimica Acta* **2011,** 56, (11), 3926-3933.

71. Wang, C.; Lee, D. H.; Hexemer, A.; Kim, M. I.; Zhao, W.; Hasegawa, H.; Ade, H.; Russell, T. P. *Nano Letters* **2011,** 11, (9), 3906-3911.

72. Virgili, J. M.; Tao, Y. F.; Kortright, J. B.; Balsara, N. P.; Segalman, R. A. *Macromolecules* **2007,** 40, (6), 2092-2099.

73. Handlin, D. L.; Thomas, E. L. *Macromolecules* **1983,** 16, (9), 1514-1525.

74. Sawyer, L. C.; Grubb, D. T.; Meyers, G. F., Polymer microscopy. In 3rd ed.; Springer,: New York, 2008; pp xiv, 540 p. ill. (some col.) 27 cm.

75. Thomas, K. E.; Sloop, S. E.; Kerr, J. B.; Newman, J. *Journal of Power Sources* **2000,** 89, (2), 132-138.

76. Hafezi, H.; Newman, J. *Journal of the Electrochemical Society* **2000,** 147, (8), 3036-3042.

77. Newman, J.; Thomas, K. E.; Hafezi, H.; Wheeler, D. R. *Journal of Power Sources* **2003,** 119, 838-843.

78. Ferry, A.; Doeff, M. M.; DeJonghe, L. C. *Electrochimica Acta* **1998,** 43, (10-11), 1387-1393.

79. Ferry, A.; Doeff, M. M.; De Jonghe, L. C. *Journal of the Electrochemical Society* **1998,** 145, (5), 1586-1592.

80. Georen, P.; Lindbergh, G. *Electrochimica Acta* **2001,** 47, (4), 577-587.

81. Ciosek, M.; Marcinek, M.; Zukowska, G.; Wieczorek, W. *Electrochimica Acta* **2009,** 54, (19), 4487-4493.

82. Evans, J.; Vincent, C. A.; Bruce, P. G. *Polymer* **1987,** 28, (13), 2324-2328.

83. Doyle, M.; Newman, J. *Journal of the Electrochemical Society* **1995,** 142, (10), 3465-3468.

84. Roe, R. J. *Journal of Physical Chemistry* **1968,** 72, (6), 2013-&.

85. Hafezi, H. Characterization of Transport Phenomena in Polymer Electrolyte Systems. University of California, Berkeley, Berkeley, 2002.

86. Robinson, R. A.; Stokes, R. H., *Electrolyte Solutions*. Academic Press: New York, 1960.

87. Park, M. J.; Downing, K. H.; Jackson, A.; Gomez, E. D.; Minor, A. M.; Cookson, D.; Weber, A. Z.; Balsara, N. P. *Nano Letters* **2007,** 7, (11), 3547-3552.

88. Elabd, Y. A.; Hickner, M. A. *Macromolecules* **2011,** 44, (1), 1-11.

89. Chaikin, P. M.; Lubensky, T. C., *Principles of condensed matter physics*. Cambridge University Press: Cambridge ; New York, NY, USA, 1995; p 699 p.

90. Gennes, P. G. d.; Prost, J., *The physics of liquid crystals*. 2nd ed.; Clarendon Press Oxford, 1993; p xvi, 597 p.

91. Amundson, K.; Helfand, E.; Davis, D. D.; Quan, X.; Patel, S. S.; Smith, S. D. *Macromolecules* **1991,** 24, (24), 6546-6548.

92. Boker, A.; Elbs, H.; Hansel, H.; Knoll, A.; Ludwigs, S.; Zettl, H.; Urban, V.; Abetz, V.; Muller, A. H. E.; Krausch, G. *Physical Review Letters* **2002,** 89, (13).

93. Schmidt, K.; Schoberth, H. G.; Ruppel, M.; Zettl, H.; Hansel, H.; Weiss, T. M.; Urban, V.; Krausch, G.; Boker, A. *Nature Materials* **2008,** 7, (2), 142-145.

94. Xu, T.; Zvelindovsky, A. V.; Sevink, G. J. A.; Lyakhova, K. S.; Jinnai, H.; Russell, T. P. *Macromolecules* **2005,** 38, (26), 10788-10798.

95. Schmidt, K.; Pester, C. W.; Schoberth, H. G.; Zettl, H.; Schindler, K. A.; Boker, A. *Macromolecules* **2010,** 43, (9), 4268-4274.

96. Wang, J. Y.; Xu, T.; Leiston-Belanger, J. M.; Gupta, S.; Russell, T. P. *Physical Review Letters* **2006,** 96, (12).

97. Xu, T.; Zvelindovsky, A. V.; Sevink, G. J. A.; Gang, O.; Ocko, B.; Zhu, Y. Q.; Gido, S. P.; Russell, T. P. *Macromolecules* **2004,** 37, (18), 6980-6984.

98. Tsori, Y.; Tournilhac, F.; Andelman, D.; Leibler, L. *Physical Review Letters* **2003,** 90, (14).

99. Tsori, Y.; Tournilhac, F.; Leibler, L. *Macromolecules* **2003,** 36, (15), 5873-5877.

100. Kohn, P.; Schroter, K.; Thurn-Albrecht, T. *Physical Review Letters* **2009,** 102, (21).

101. Hexemer, A.; Bras, W.; Glossinger, J.; Schaible, E.; Gann, E.; Kirian, R.; MacDowell, A.; Church, M.; Rude, B.; Padmore, H. *Journal of Physics: Conference Series* **2010**, 012007 (11 pp.).

102. Hajduk, D. A.; Harper, P. E.; Gruner, S. M.; Honeker, C. C.; Kim, G.; Thomas, E. L.; Fetters, L. J. *Macromolecules* **1994,** 27, (15), 4063-4075.

103. Vigild, M. E.; Almdal, K.; Mortensen, K.; Hamley, I. W.; Fairclough, J. P. A.; Ryan, A. J. *Macromolecules* **1998,** 31, (17), 5702-5716.

Appendix A – WAXS and XRD Analysis

Wide-angle X-ray scattering (WAXS) and X-ray diffraction (XRD) are similar techniques that were used to determine the presence or absence of crystallinity. Both techniques were performed in transmission mode, similar to the SAXS setup illustrated in Figure 1. The distinction between the techniques, as applied in this work, is that WAXS experiments were performed at the Advanced Light Source beamline 7.3.3, often simultaneously with SAXS experiments, whereas XRD experiments were performed at the Molecular Foundry Inorganic Nanostructures Facility using an X-ray diffractometer with a copper K-α X-ray source and a 2D detector (Bruker AXS D8). WAXS and XRD experiments probe large scattering angles (θ = 2 to 85°), thus providing information about structures on atomic length-scales. WAXS experiments are typically much higher quality due to the high intensity, coherency, and monochromatic X-rays from the synchrotron source. WAXS experiments also exhibit a larger q-range due to the large size of the 2D detector. XRD offers the advantage of easy access relative to synchrotron facilities.

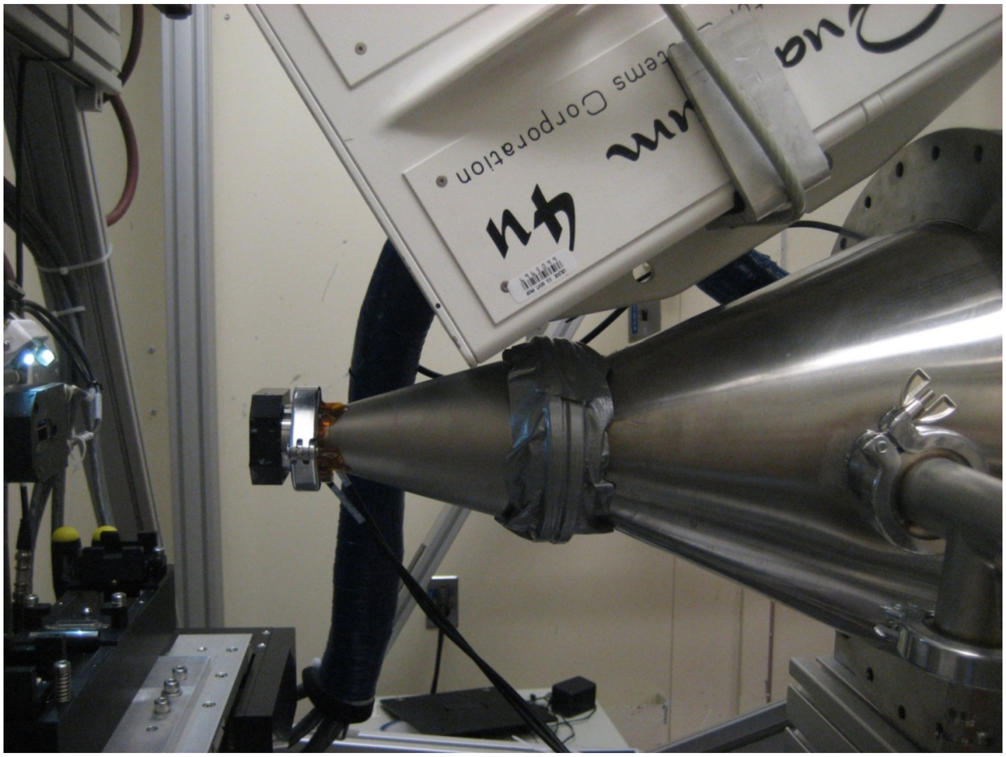


Figure 36 – WAXS detector setup at ALS beamline 7.3.3.
The X-rays originate from the left and strike a sample (not shown). X-rays scattered at wide angles strike the tilted detector for WAXS measurements. X-rays scattered at small angles travel through the flight tube and strike another detector for SAXS measurements. Tilting the WAXS detector allows simultaneous SAXS/WAXS acquisition, but complicates WAXS data analysis.

The WAXS detector in Figure 36 is tilted towards the sample. At the time these experiments were performed, the Nika X-ray scattering data analysis package was not programmed to handle tilted detectors. In order to convert the 2D detector images to 1D scattering profiles, custom Igor Pro code was written to calculate the q-value corresponding to each pixel and then integrate the pixel intensities along a constant-*q* contour. This code is described in Appendix B. The newer versions of Nika perform this algorithm, and the program now handles tilts correctly.

Figure 37 compares WAXS and XRD profiles for a sample of SEO(54-23) $r = 0.085$ at 90 °C. This sample contained contaminated LiTFSI (from Sigma Aldrich) that exhibited salt precipitation upon heating. This behavior is not observed for highly purified LiTFSI (acquired from Novolyte Technologies, formerly known as Ferro Corp.).

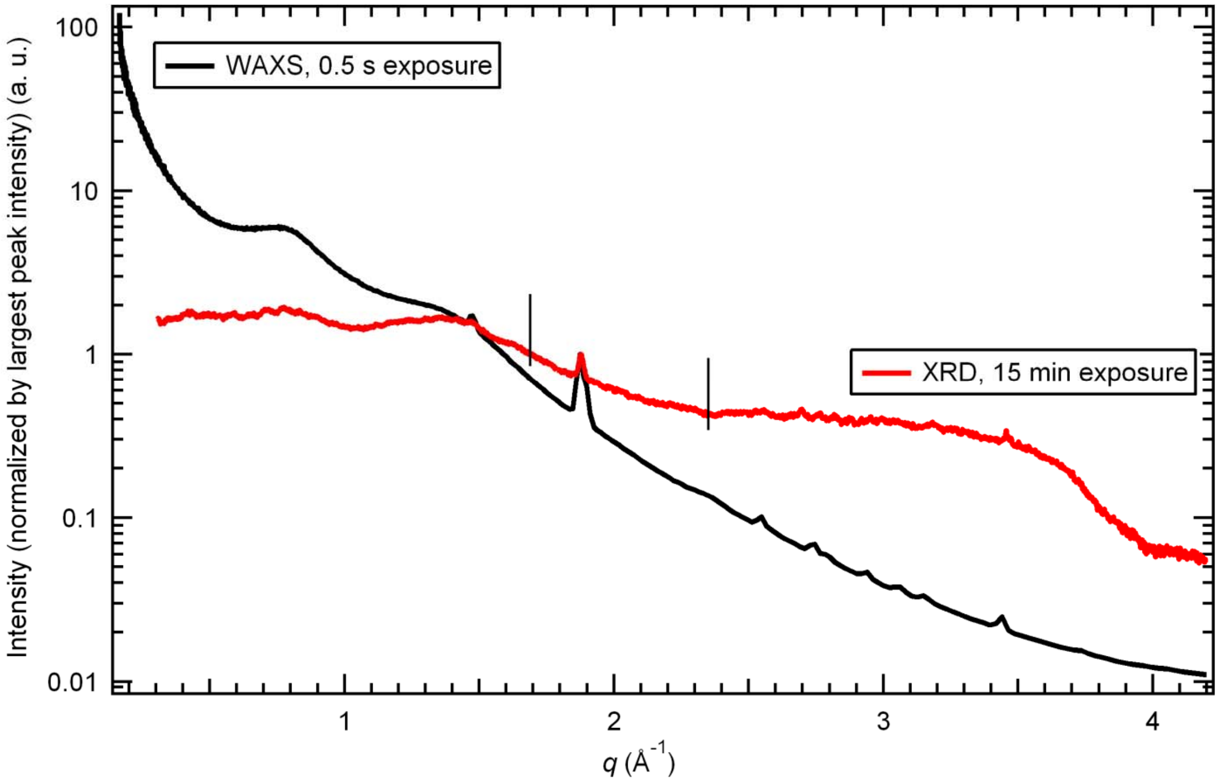


Figure 37 – WAXS and XRD comparison.
Both traces are for the same sample of SEO(54-23) $r = 0.085$ at 90 °C, showing precipitation of contaminants. The XRD data is taken from three different detector angles, each of which accesses a different *q*-range. The regions above 3.6 Å^{-1} in the XRD trace show reduced intensity due to interference with the sample holder. The three regions are stitched together on the XRD trace and the middle region is demarcated with vertical lines. All of the traces are normalized by the peak intensity at 1.88 Å^{-1} for comparison purposes. The WAXS pattern has a significantly higher signal/noise ratio and a wider *q*-range in one image than can be accessed via the XRD. Both scattering profiles have a clear peak at 1.88 Å^{-1} that indicates the presence of crystalline regions. The WAXS pattern has large humps in the range of 0.5 to 1.5 Å^{-1} that arise from the amorphous polymer regions. The XRD data do not clearly show the amorphous peaks.

Figure 38 shows the WAXS profiles for a sample of SEO(54-23) with $r = 0.085$ at various temperatures. Highly purified LiTFSI was used for this sample. The room-temperature WAXS profiles show extensive crystallinity, as indicated by the sharp peaks. This is typical behavior for PEO/LiTFSI systems at room temperature. Upon heating to 60 °C, the crystals melt, and the WAXS profiles are characterized by broad humps and no sharp peaks are apparent.

The sample remains amorphous at 120 °C, indicating that no salt precipitation occurred. This was a key discovery for the conductivity study in Chapter 3, since it indicated that the conductivity drops were not caused by salt precipitation.

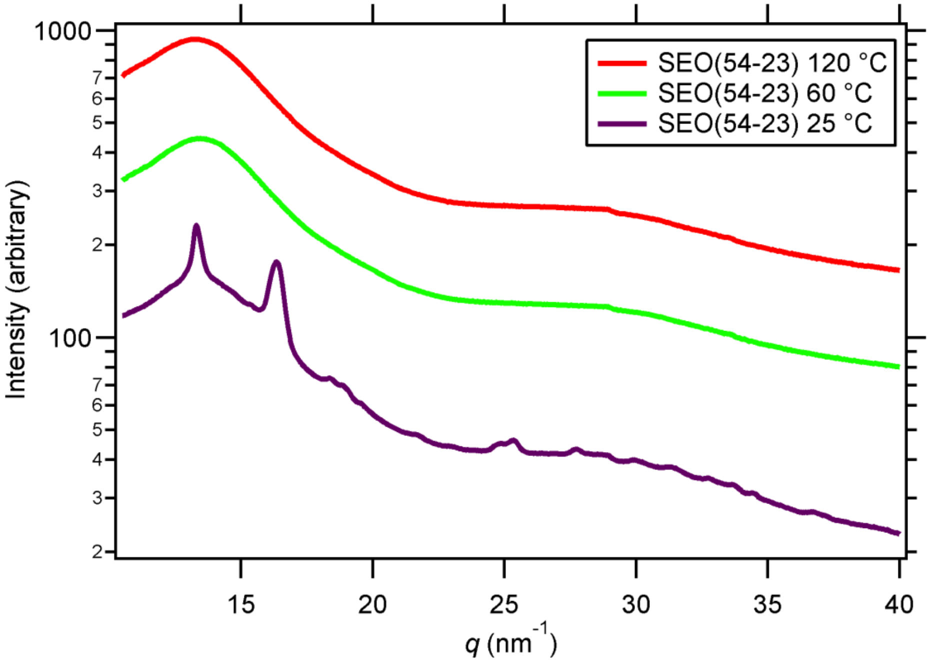


Figure 38 – WAXS profiles of SEO(54-23) *r* = 0.085 at various temperatures.
The 25 °C WAXS profile shows crystalline peaks that are consistently seen in room temperature PEO/LiTFSI mixtures. Above the PEO/LiTFSI melting point, the sample shows no signs of crystallinity.

Appendix B – Selected Igor Pro Functions

Igor Pro is a data analysis and graphing program created by Wavemetrics Inc. Custom Igor Pro functions were used extensively in this work for data analysis and visualization. All functions were tested on Igor Pro version 6.22A. An exhaustive list of all custom functions would require hundreds of pages, and most functions perform variants of similar tasks. Instead, functions that perform unique tasks are presented in this Appendix. Many functions are inter-dependent on each other, on functions included in the Igor Pro "Nika" macro (available from Jan Ilavsky at the Advanced Photon Source), or other custom functions that I have written but not included here.

This Appendix is not instructive with regards to Igor Pro programming. For that purpose, please consult the Igor Pro manual, which is of very high quality, or the Wavemetrics website, which includes links to a discussion forum and email list.

Instead, this Appendix provides short explanations of key functions. A motivated user should find these examples quite instructive. Many of the functions that were not included build upon these core functions; for example, a function was written that executes the Contin algorithm with various fitting conditions. Given the necessary functions described here ("ExecuteContin", "makeContin", and "ReadContin") that function is simple to implement using basic Igor programming techniques. A list of the Igor Pro functions chosen for this Appendix follows on the next page, along with a brief description of what the function does. Further details regarding the functions' implementation can be found within the comments of the functions themselves.

The complete programming package is called "SuperPanel". The name originated from the graphical design of the program, which was developed around the Igor Pro object known as a "panel". The interface ties together most of the functions used in this work. Many, but not all, of the functions can be executed directly from the panel using buttons. Functions that take a text input "ctrlname" are typically designed to be executed by a button ─ the button name being the input parameter for "ctrlname" when the button is pressed.

List of Selected Igor Pro Functions

Description of Functions

Auto_Base_N_Peak(ctrlname)

This function operates on all selected 1-dimensional small-angle X-ray scattering profiles. The profiles must all have the same number of points. It is also necessary to specify two masks, as evident in the code. One mask is used to mask off everything that is not part of the "background". The residual portions are then fitted to an exponential decay, which is then subtracted from the data. The other mask isolates regions around the peak that should not be fit. For instance, if there is a broad disordered peak with a sharp ordered peak overlapping, the sharp peak should be masked off. The function then fits the region around the peak, as selected by cursors placed by the user, to a Gaussian. The Gaussian fit is then subtracted (to remove the disordered portion of the peak) and the residual ordered portion is displayed. The function then calculates various parameters, including the scattering invariant of the residual ordered peak and its relative intensity to the total scatter in the selected region. This function implements advanced fitting methods, including automation, fitting constraints and masks.

cleanstringfortran(stringtoclean)

Converts certain strings to a Fortran-readable format.

ContinComplete(ywave, twave, basis, header)

This function creates a Contin input file (using "makeContin"), executes Contin to create an output file (using "ExecuteContin") and then reads the output file back into Igor (using "ReadContin").

ConvertWAXSImages(ctrlname)

Executes WAXS2Dto1D to reduce all of the selected 2-dimensional (2D) detector images to 1D profiles. This function relies on the Nika package to select and load 2D images.

ExecuteContin(inputName, outputName, pathStr)

This function is called after a Contin input file has been written (using "makeContin"). The input file must be in the same folder as the Contin executable. The executable must be named "contin.exe" and the directory must be chosen beforehand using SuperPanel. Contin is executed via the Windows command line. This function creates a batch file with the appropriate commands and then executes the batch file. This has been tested extensively without problems on both Windows XP and Windows 7. Later versions of Windows may deprecate this functionality. This will have to be re-written for use on other operating systems.

FastWAXSQMap(ctrlname)

This function takes the same inputs and performs the same task as the SlowWAXSQMap function. Instead of using a for-loop, however, this function makes use of Igor Pro's wave arithmetic. Wave arithmetic is essentially just optimized code for matrix operations. This speeds up the program execution by a factor of 2 over the use of a loop. Additionally, the MultiThread keyword is used so that the code executes on all available processor cores. This parallelization gives an additional speed boost. For example, for a dual-core processor, using this function is approximately 4 times faster than using SlowWAXSQMap. For a six-core processor, the speedup is approximately 12 times faster. This functionality is now included in the Nika package, but was not available at the time this algorithm was written.

lnspacer(timewavetoscale, lowest, highest)

Used to create natural-logarithmically-spaced, truncated data from that which is input. This is necessary due to limitations in Contin that restrict how many data points can be input.

makeContin(ywave, twave, basis,header)

Creates a Fortran-readable text file that is designed to be input to the Contin executable program. The Contin executable is available freely on the internet from Steven Provencher - the creator of the Contin algorithm. This input file contains all of the data to be fit by Contin. The data can be manually input to this function, or it can be selected from a graph in SuperPanel and automatically extracted. The input file also contains all of the parameters to be used by Contin, which are taken from the Contin manual. See the references by Steven Provencher for details.[55-57] The input file uses a very archaic Fortran input style. All of the data must have positive values. There is a maximum of around 8000 data points. All data must have a specific number of spaces. If more than 8000 data points are used, then the input data is filtered so that natural-logarithmically-spaced points are obtained (using the lnspacer function), as recommended by Provencher. Many other Contin controls are available. The controls used in this function were designed for specific use for the analysis in Chapter 2. The input data should be either evenly spaced with time (constant sampling rate), or should be logarithmically-spaced with time.

MakeSAXSMovie([StartNewMovie, CloseMovieWhenFinished])

Creates a movie from the selected X-ray scattering data sets. The movie controls are hard-coded (for instance, the number of frames per second). The display format of the scattering image is also hard-coded and should be adjusted manually for each data set to be reduced. The function can be told whether or not to start a new movie and to close the movie when it is finished. This provides flexibility for adding frames in a certain order. This function is dependent on the Nika package for loading and displaying X-ray detector images.

MakeSupPanel()

Builds the graphical user interface (GUI) around which SuperPanel is based. This show the origin of all of the panel objects (buttons, lists, checkboxes, etc.). It also declares most of the global variables and waves used to manage the program. Many of the global variables and objects on the panel are referenced within other functions.

MakeTimeStampWaves()

This function creates a set of timestamps based on the selected X-ray scattering files. The first file is assigned time zero, and subsequent files are given times relative to the first. The times are based on the "last modified" data on the file. This function is dependent upon the Nika package.

ReadContin(basis, setNewPath, autoFileSelect)

Reads the selected Contin file(s). This function will read in the "chosen" distribution function, as well as the specified moments (0th, 1st and 2nd). The files are chosen from the list in SuperPanel. The ".txt" file extension should be used, and the directory must be chosen as well.

ReadOrdErrAbs(refnum, basis)

Reads the "chosen" distribution function from a Contin output file. This function returns the line number of the last line of the distribution function. That number is later used by ReadContin as a starting point for reading the solution moments.

SaveSAXSImagesToFile()

This function will load and display X-ray detector images. The images are then saved to file. The display format of the scattering image is also hard-coded and should be adjusted manually for each data set to be reduced. This function is dependent on the Nika package for loading and displaying X-ray detector images.

SlowWAXSQMap(ctrlname)

Using hard-coded input, this function creates a 2-dimensional map. Each point on the map corresponds to a pixel on a 2-dimensional wide-angle X-ray scattering (WAXS) detector. A *q*-value is calculated for each point based on the inpur parameters. The input parameters include corrections for the vertical and horizontal tilts. This map is later used to find the integrated intensity for a given *q*-value.

Solartron_load_generic(ctrlname)

File loader used to load .csv-format files exported by the Solartron potentiostat software. The .csv file contains data from 1 to 8 channels, depending on how many were grouped when the experiment was run, that is interleaved randomly in the order from it was collected. This function loads that file, sorts the data by channel, step and repeat number, and then graphs the results. It was extended to include capacity calculations which are useful for batteries. If the capacities are not important - in diffusion/relaxation studies, for example - the input "diffusion" can be used to prevent the capacities from being output into tables.

Spaces(numspaces)

Returns a string consisting of spaces, the number of which is determined by the input parameter. This function is also useful for readability of the MakeContin function. It simply adds a string with the specified number of spaces. For example, Spaces(10) is equivalent to the string "   ".

WAXS2Dto1D(ctrlname)

This function uses the *q*-map created by SlowWAXSQMap or FastWAXSQMap to create 1-dimensional WAXS profiles. First, a detector image is loaded. Next, each point on the detector is referenced to the *q*-map. The *q*-values are binned, and the intensity of each pixel is added to the total for the corresponding *q*-value. Finally, the data is normalized to correct for the number of pixels that are masked and the 1D is displayed to the user. This relies on the Nika package to select and load 2D images.

```
Function Auto_Base_N_Peak(ctrlname)
	//1) Duplicates data
	//2) Subtracts baseline from duplicates (all operations on duplicated data) (Uses MaskForFitting)
	//3) Deletes points after cursor B and before cursor A
	//4) Fits Gaussian to data between cursors C and D
	//5) Prints "invariant" for the Gaussian fit and for the remaining portion
	string ctrlname

	if(strlen(ctrlname) == 0)
		Abort "ctrlname was blank.  Input to Auto_Base_N_Peak must include a string (eg Auto_Base_N_Peak(\"InputName\")"
	endif

	string OldDF = GetDataFolder(1)

	if( !DataFolderExists("root:Packages:SuperPanel:AutoBase"))
		NewDataFolder root:Packages:SuperPanel:AutoBase
	endif
	if(!DataFolderExists("root:Packages:SuperPanel:KeyResults"))
		NewDataFolder root:Packages:SuperPanel:KeyResults
	endif

	SetDataFolder root:Packages:SuperPanel:AutoBase

	variable i, ptsinlist
	wave CurWindowTraceListIndices = root:Packages:SuperPanel:CurWindowTraceListIndices
	wave/T CurWindowTraceList = root:Packages:SuperPanel:CurWindowTraceList
	wave WaveToIntegrate, WaveToIntegrateX
	ptsinlist = dimsize(CurWindowTraceListIndices,0)
	SVAR SelectedGraphName = root:Packages:SuperPanel:SelectedGraphName
	wave MaskForFitting = root:Packages:SuperPanel:AutoBase:MaskForFitting
	wave MaskForFitting2 = root:Packages:SuperPanel:AutoBase:MaskForFitting2

	if(!waveexists(MaskForFitting))
		Abort "MaskForFitting (for baseline subtraction fit) was not found - aborting"
	endif
	if(!waveexists(MaskForFitting2))
		Abort "MaskForFitting2 (for peak fitting fit) was not found - aborting"
	endif

	if(sum(CurWindowTraceListIndices) < 1)
		Abort "No data selected - aborting"
	endif

	variable CursorA, CursorB
	try
		CursorA = pcsr(A, SelectedGraphName); AbortOnRTE
		CursorB = pcsr(B, SelectedGraphName); AbortOnRTE
	catch
	//	Abort "Cursors A and B must both be on the following graph: " + SelectedGraphName + " - aborting"
	endtry

	variable alphaPt =0 //alphaPt and betaPt are hard-coded points that define where integration regions between peaks should be
	variable betaPt = 0

	variable temp
	if(CursorB < CursorA) //if cursors are backwards
		temp = CursorA
		CursorA = CursorB
		CursorB = temp //reverse them
	endif

	if(alphaPt == 0 && betaPt != 0)
		Abort "If alphaPt is 0, betaPt must also be 0.  These values are hard-coded, change before next execution -- aborting"
	endif
	if(alphaPt != 0)
		if(cursorA > alphaPt || cursorB < alphaPt)
			abort "alphaPt must be between cursors A and B -- aborting"
		endif
		if(betaPt !=0)
			if(alphaPt > betaPt || cursorB < betaPt)
```

```
                abort " betaPt must be between alphaPt and cursor B - aborting"
            endif
        endif
    endif

    //co-opt the "ctrlname" input to stand in for a basis name.  Will change later if this is made into a button
    string KeyResidualName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_KRS" //stands for "key residual"
    string KeyRatioName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_KRA" //stands for "key ratio"
    string KeyWaveName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_KWN" //stands for "key wave name"
    string KeyInvariantName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_INV"//stands for "invariant"
    string KeyInvariantRatioName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_INR"//stands for "invariant ratio"
    string Peak1WaveName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_PK1"
    string Peak2WaveName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_PK2"
    string Peak3WaveName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_PK3"
    string Peak1RatioName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_P1R"
    string Peak2RatioName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_P2R"
    string Peak3RatioName = "root:Packages:SuperPanel:KeyResults:" + ctrlname[0,26] + "_P3R"
    make/D/O/N=0 $KeyResidualName, $KeyRatioName, $Peak1WaveName, $Peak2WaveName, $Peak3WaveName,
    $Peak1RatioName, $Peak2RatioName, $Peak3RatioName, $KeyInvariantName, $KeyInvariantRatioName
    make/D/O/T/N=0 $KeyWaveName
    wave KeyResidual = $KeyResidualName
    wave KeyRatio = $KeyRatioName
    wave/T KeyWave = $KeyWaveName
    wave KeyInvariantWave = $KeyInvariantName
    wave KeyInvariantRatioWave = $KeyInvariantRatioName
    wave Peak1Wave = $Peak1WaveName
    wave Peak2Wave = $Peak2WaveName
    wave Peak3Wave = $Peak3WaveName
    wave Peak1Ratio = $Peak1RatioName
    wave Peak2Ratio = $Peak2RatioName
    wave Peak3Ratio = $Peak3RatioName

    string ScrapText
    if(alphaPt == 0)
        ScrapText = "Name of Wave\tResidual integral\tResidual ratio"
    elseif(betaPt == 0)
        ScrapText = "Name of Wave\tResidual integral\tResidual ratio\tPeak1 integral\tPeak1 ratio\tPeak2 integral\tPeak2 ratio"
    else
        ScrapText = "Name of Wave\tResidual integral\tResidual ratio\tPeak1 integral\tPeak1 ratio\tPeak2 integral\tPeak2
    ratio\tPeak3 integral\tPeak3 ratio"
    endif
    ScrapText += "\r"
    print ScrapText //this initial value is the header that is printed to the command line and is printed to the clipboard for later
    pasting
    PutScrapText ScrapText

    variable k = 0
    for(i=0;i<ptsinlist;i+=1)
        if(CurWindowTraceListIndices[i]==1)
            wave WaveToFit = tracenametowaveref(SelectedGraphName,  curwindowtracelist[i])
            wave WaveToFitX = XWaveRefFromTrace(SelectedGraphName, CurWindowTraceList[i])
            string Basis = cleanupname(NameOfWave(WaveToFit),0)

            string ManipulatedWaveNameY = Basis[0,28] + "_Y"
            string ManipulatedWaveNameX = Basis[0,28] + "_X"
            Duplicate/O WaveToFit, $ManipulatedWaveNameY
            Duplicate/O WaveToFitX, $ManipulatedWaveNameX
            wave ManipulatedWaveY = $ManipulatedWaveNameY
            wave ManipulatedWaveX = $ManipulatedWaveNameX

            if(numpnts(MaskForFitting) != numpnts(ManipulatedWaveY))
                Abort "MaskForFitting was a different size than " + nameofwave(ManipulatedWaveY) + "aborting"
            else
                Make/D/O/T/N=3 T_Constraints
                T_Constraints[0] = {"K0 > 2e-5","K1 > .005","K2 > .001"}
                CurveFit/X/N/Q/NTHR=0 exp_XOffset ManipulatedWaveY /X=ManipulatedWaveX
    /M=MaskForFitting/D/C=T_Constraints
//              wave w_coef
//              print num2str(w_coef[0]) + "\t" + num2str(w_coef[1]) + "\t" + num2str(w_coef[2]) + "\t" + num2str(w_coef[3])
            endif
            string S_info
            string BaselineWaveName = stringbykey("AUTODESTWAVE", S_info, "=") // S_info is generated by curveFit above - includes name of destination wave
            wave BaselineWave = $BaselineWaveName
            string PreferredBaselineName = basis[0,26] + "_BAS"
            Duplicate/O BaselineWave, $PreferredBaselineName
            killwaves/Z BaselineWave
            wave BaselineWave = $PreferredBaselineName
//Subtracts baseline from "manipulated" data
            ManipulatedWaveY = WaveToFit - BaselineWave
            if(numpnts(MaskForFitting2) != numpnts(ManipulatedWaveY))
                Abort "MaskForFitting2 was a different size than " + nameofwave(ManipulatedWaveY) + " aborting"
            else
                Make/D/O/T/N=6 T_Constraints
                T_Constraints[0] = {"K0>-1E-4","K1 >4E-8","K1 < 1E-6","K2 > .075","K2 < .085","K3 > 2E-4"}
                CurveFit/X/N/Q/NTHR=0 lor ManipulatedWaveY[CursorA,CursorB] /X=ManipulatedWaveX /M=MaskForFitting2 /D//C=T_Constraints//Fits lorentzian btw cursors A and B
//          wave w_coef
//          print num2str(w_coef[0]) + "\t" + num2str(w_coef[1]) + "\t" + num2str(w_coef[2]) + "\t" + num2str(w_coef[3])
            endif

            string PeakFitName = stringbykey("AUTODESTWAVE",S_info,"=")//gets name of fitted wave
            wave PeakFitWave = $PeakFitName
            string PreferredPeakName = Basis[0,26] + "_LOR"
            duplicate/O PeakFitWave, $PreferredPeakName
            killwaves/Z PeakFitWave
            wave PeakFitWave = $PreferredPeakName

            variable j, buffer, PeakMax
            buffer = ManipulatedWaveY[CursorA]
            PeakMax = CursorA //this is the starting position -- this function will find the x-value corresponding to the max peak intensity in the y-value
            for(j=CursorA+1; j< CursorB; j+=1)
                buffer = ManipulatedWaveY[j]
                if(buffer > ManipulatedWaveY[PeakMax])
                    PeakMax = j
                endif
            endfor

            string ResidualName = basis[0,26] + "_RES"
            duplicate/O ManipulatedWaveY, $ResidualName
            wave Residual = $ResidualName
            Residual = ManipulatedWaveY - PeakFitWave
            if(k == 0)
                print "Preparing to subtract basis from all residuals: " + nameofwave(Residual)
                Residual = Residual[x]*(Residual[x] > 0) //any negative value on the wave "Residual" is converted to zero
                duplicate/O Residual, ResidualPrime
                wave ResidualPrime
            else
                Residual = Residual - ResidualPrime
            endif
            Residual = Residual[x]*(Residual[x] > 0) //any negative value on the wave "Residual" is converted to zero
            variable left, right
//      for(j=PeakMax;j>0;j-=1)
//          buffer = Residual[j]
//          if(buffer > 0) //looking for the left edge where Residual dips negative
//              left = j
//          else
//              break
//          endif
//      endfor

//      for(j=PeakMax;j<numpnts(Residual);j+=1)
//          buffer = Residual[j]
//          if(buffer > 0) //looking for the right edge where Residual dips negative
//              right = j
//          else
//              break
//          endif
//                      endfor


//Since Residual is now >= 0 at all points, there is no need to find the edges that dip negative.
            left = cursorA
            right = cursorB
            string ResidualIntegralName = basis[0,24] + "_RESIN"
            string FullIntegralName = basis[0,24]+"_FULIN"

    //regular integrals
            Integrate/Meth=1 Residual /D=$ResidualIntegralName /X=ManipulatedWaveX
            Integrate/Meth=1 ManipulatedWaveY /X=ManipulatedWaveX /D=$FullIntegralName
            wave ResidualIntegral = $ResidualIntegralName
            wave FullIntegral = $FullIntegralName
            variable ResidualIntegralResult = ResidualIntegral[right] -ResidualIntegral[left]
            variable FullIntegralResult = FullIntegral[right] -FullIntegral[left]
            variable Pk1 = ResidualIntegral[alphaPt] - ResidualIntegral[left]
            variable Pk1Ratio = Pk1/(FullIntegral[alphaPt] - FullIntegral[left]
            variable Pk2, Pk2Ratio, Pk3, Pk3Ratio
            if(betaPt == 0)
                Pk2 = ResidualIntegral[right] - ResidualIntegral[alphaPt]
                Pk2Ratio = Pk2/(FullIntegral[right] - FullIntegral[alphaPt])
            else
                Pk2 = ResidualIntegral[betaPt] - ResidualIntegral[alphaPt]
                Pk2Ratio = Pk2/(FullIntegral[betaPt] - FullIntegral[alphaPt])
                Pk3 = ResidualIntegral[right] - ResidualIntegral[betaPt]
                Pk3Ratio = Pk3/(FullIntegral[right] - FullIntegral[betaPt])
            endif

            string output = nameofwave(WaveToFit) + "\t" + num2str(ResidualIntegralResult) + "\t" +
    num2str(ResidualIntegralResult/FullIntegralResult)
            insertpoints k, 1, KeyWave, KeyResidual,KeyRatio
            KeyWave[k] = nameofwave(WaveToFit)
            KeyResidual[k] = ResidualIntegralResult
            KeyRatio[k] = ResidualIntegralResult/FullIntegralResult

            if(alphaPt > 0)
                insertpoints k, 1, Peak1Wave, Peak2Wave, Peak1Ratio, Peak2Ratio
                Peak1Wave[k] = Pk1
                Peak1Ratio[k] = Pk1Ratio
                Peak2Wave[k] = Pk2
                Peak2Ratio[k] = Pk2Ratio

                if(betaPt == 0)//case where only alphaPt is specified
                    output += "\t" + num2str(Pk1) + "\t" + num2str(Pk1Ratio) + "\t" + num2str(Pk2) + "\t" + num2str(Pk2Ratio)
                else //case where alphaPt and betaPt are both specified
                    output += "\t" + num2str(Pk1) + "\t" + num2str(Pk1Ratio) + "\t" + num2str(Pk2) + "\t" + num2str(Pk2Ratio) +
    "\t" + num2str(Pk3) + "\t" + num2str(Peak3Ratio)
                    insertpoints k, 1, Peak3Wave, Peak3Ratio
                    Peak3Wave[k] = Pk3
                    Peak3Ratio[k] = Pk3Ratio
                endif
            endif

//invariant integrals
            Duplicate/O ManipulatedWaveY, ManipulatedWaveY2
            Duplicate/O Residual, Residual2
            ManipulatedWaveY2 = ManipulatedWaveY2 * ManipulatedWaveX^2
            Residual2 = Residual2 * ManipulatedWaveX^2 //Converts to Iq^2 (scattering invariant format)
            string ResidualInvariantName = basis[0,24] + "_RSINV"
            string FullInvariantName = basis[0,24] + "_FLINV"
            Integrate/Meth= 1 Residual2 /D=$ResidualInvariantName /X=ManipulatedWaveX
            Integrate/Meth=1 ManipulatedWaveY2 /X=ManipulatedWaveX /D=$FullInvariantName
            wave ResidualInvariant = $ResidualInvariantName
            wave FullInvariant = $FullInvariantName
            variable ResidualInvariantResult = ResidualInvariant[right] - ResidualInvariant[left]
            variable FullInvariantResult = FullInvariant[right] - FullInvariant[left]
            insertpoints k, 1, KeyInvariantWave, KeyInvariantRatioWave
            KeyInvariantWave[k] = ResidualInvariantResult
            KeyInvariantRatioWave[k] = ResidualInvariantResult/FullInvariantResult
            if(k == 0)
                KeyResidual[0] = 0
                KeyRatio[0] = 0
                Peak1Wave[0] = 0
                Peak1Ratio[0] = 0
                Peak2Wave[0] = 0
                Peak2Ratio[0] = 0
                Peak3Wave[0] = 0
                Peak3Ratio[0] = 0
                KeyInvariantWave[0] = 0
                KeyInvariantRatioWave[0]=0
            endif

            if(k==0)
                string keyInvName = basis[0,24] + "_"
            endif

            print output
//          print "(L,R) = " + num2str(left) +"\t" + num2str(right)
            ScrapText += output + "\r"
            PutScrapText ScrapText

            doWindow BaselineSubtracted
            if(V_Flag ==1 || V_Flag ==2)   //this means window exists
                doWindow/F BaselineSubtracted
                appendtograph/W=BaselineSubtracted  ManipulatedWaveY vs ManipulatedWaveX
                appendtograph/W=BaselineSubtracted  PeakFitWave, BaselineWave
//              appendtograph/W=BaselineSubtracted WaveToFit vs WaveToFitX
            else
                Display /N=BaselineSubtracted ManipulatedWaveY vs ManipulatedWaveX
                ModifyGraph log(left)=1
                appendtograph/W=BaselineSubtracted PeakFitWave, BaselineWave
//              appendtograph/W=BaselineSubtracted WaveToFit vs WaveToFitX
                Label bottom "\\Z16 Q (1/Angstrom)"
                Label left "\\Z16I"
            endif
            doWindow Residual
            if(V_Flag ==1 || V_Flag ==2)   //this means window exists
                doWindow/F Residual
                appendtograph/W=Residual Residual vs ManipulatedWaveX

            else
                Display /N=Residual Residual vs ManipulatedWaveX
                Label bottom "\\Z16 Q (1/Angstrom)"
                Label left "\\Z16I (a.u.)"
            endif

            k += 1
        endif
    endfor

    killwaves /Z ResidualIntegral, FullIntegral, ResidualInvariant, FullInvariantResult

    if(alphaPt == 0)
        killwaves/z Peak1Wave, Peak2Wave, Peak3Wave, Peak1Ratio, Peak2Ratio, Peak3Ratio
    endif
    if(betaPt == 0) //program aborts early if alphaPt ==0 and betaPt does not
        killwaves/z Peak3Wave, Peak3Ratio //alphaPt could be > 0, in which case we only want to kill Peak3Wave.  It may
    already be dead from the previous block of code but that's ok
    endif

    SetDataFolder OldDF
end
```

```
Function/s cleanstringfortran(stringtoclean)
	string stringtoclean
	string finaloutput =stringtoclean
	finaloutput = replacestring("+",finaloutput,"")

	finaloutput = replacestring("0.",finaloutput,".0")
	finaloutput = replacestring("1.",finaloutput,".1")
	finaloutput = replacestring("2.",finaloutput,".2")
	finaloutput = replacestring("3.",finaloutput,".3")
	finaloutput = replacestring("4.",finaloutput,".4")
	finaloutput = replacestring("5.",finaloutput,".5")
	finaloutput = replacestring("6.",finaloutput,".6")
	finaloutput = replacestring("7.",finaloutput,".7")
	finaloutput = replacestring("8.",finaloutput,".8")
	finaloutput = replacestring("9.",finaloutput,".9")

	finaloutput = replacestring("E06",finaloutput,"E07")
	finaloutput = replacestring("E05",finaloutput,"E06")
	finaloutput = replacestring("E04",finaloutput,"E05")
	finaloutput = replacestring("E03",finaloutput,"E04")
	finaloutput = replacestring("E02",finaloutput,"E03")
	finaloutput = replacestring("E01",finaloutput,"E02")
	finaloutput = replacestring("E00",finaloutput,"E01")
	finaloutput = replacestring("E-01",finaloutput,"E00")
	finaloutput = replacestring("E-02",finaloutput,"E-1")
	finaloutput = replacestring("E-03",finaloutput,"E-2")
	finaloutput = replacestring("E-04",finaloutput,"E-3")
	finaloutput = replacestring("E-05",finaloutput,"E-4")
	finaloutput = replacestring("E-06",finaloutput,"E-5")
	finaloutput = replacestring("E-07",finaloutput,"E-6")
	finaloutput = replacestring("E-08",finaloutput,"E-7")
	finaloutput = replacestring("E-09",finaloutput,"E-8")

	return finaloutput
end
```

```
Function ContinComplete(ywave, twave, basis, header)
	wave ywave, twave
	string basis, header

//	setContinPath() //brings up a dialog for user to select the file path

	PathInfo ContinPath

	makeContin(ywave, twave, basis, header)

	ExecuteContin(basis + "_IN.txt", basis + "_OUT.txt", s_path)

	ReadContin(basis, 0, 1)
End
```

```
Function ConvertWAXSImages(ctrlname) // adapted from Nika's "NI1A_LoadManyDataSetsForConv()" function
    string ctrlname
    string OldDF=GetDataFolder(1)
    setDataFolder root:Packages:Convert2Dto1D
    Wave ListOf2DSampleDataNumbers=root:Packages:Convert2Dto1D:ListOf2DSampleDataNumbers
    Wave/T ListOf2DSampleData=root:Packages:Convert2Dto1D:ListOf2DSampleData
    string SelectedFileToLoad
    variable i, imax = numpnts(ListOf2DSampleDataNumbers)
    string DataWaveName="CCDImageToConvert"

    For(i=0;i<imax;i+=1)
        if (ListOf2DSampleDataNumbers[i]) //Files selected in the "select input box" are given a 1, else 0
            SelectedFileToLoad=ListOf2DSampleData[i]        //this is the file selected to be processed
            NI1A_ImportThisOneFile(SelectedFileToLoad)
            NI1A_LoadParamsUsingFncts(SelectedFileToLoad)

            DoUpdate
            WAXS2DTo1D("ConvertWAXSImages()")
            string loadedName=RemoveEnding(SelectedFileToLoad,".img")
            loadedName = CleanupName(loadedName,1)
            string newname = loadedName[0,29] + "_W"

            Duplicate/O  root:Packages:Convert2Dto1D:WAXSIntensity, $newname

            wave NewWaveName = $newname

            string newnote=note(CCDImageToConvert)
            newnote+=NI1A_CalibrationNote()
            newnote += "OriginalFileName=" + ListOf2DSampleData[i] + ";"
            note/K NewWaveName
            note  NewWaveName, newnote
        endif
    endfor
    setDataFolder OldDF
end
```

```
Function ExecuteContin(inputName, outputName, pathStr)
	string inputName, outputName, pathStr

//	pathStr = replaceString(":", pathStr, "\\")
//	pathStr = replaceString(pathStr[0] , pathStr, pathStr[0]+":")

	pathStr = parseFilePath(5, pathStr, "\\",0,0)
//	print "pathStr now equals " + pathStr

	string IPUFpath = SpecialDirPath("Igor Pro User Files",0,1,0)// guaranteed writeable path in IP6
	string batchFileName = "ExecuteContin.bat"
	variable refNum

	NewPath/O/Q IgorProUserFiles, IPUFpath
	Open/P=IgorProUserFiles refNum as batchFileName     // overwrites previous batchfile

	string exeStr, inStr, outStr
	exeStr = "\"" + pathStr + "contin.exe" + "\""
	inStr = "\"" + pathStr + inputName + "\""
	outStr = "\"" + pathStr + outputName + "\""
	string cmdStr = exeStr + "<" + inStr + ">" + outStr
	fprintf refnum, "cmd/c \"%s\"", cmdstr
	Close refnum

	ExecuteScriptText/B "\"" + IPUFpath + "\\" + batchFileName + "\""
End
```

```
Function FastWAXSQMap(ctrlname) //Generates QMap, which is used to reduce 2D to 1D WAXS profiles
    //Scott Mullin, December 2009, samullin@gmail.com
    //Uses WaveArithmetic (p, q and r functions) and the MultiThread tag to speed up operation
    //Execution speed scales well with number of CPU cores/threads available

    string ctrlname
//  variable timerRefNum = StartMSTimer
//  variable timerRefNum2 = StartMSTimer
//  variable t1,t2

    string OldDF=GetDataFolder(1)
    if( !DataFolderExists("root:Packages:"))
        NewDataFolder root:Packages

    endif

    if( !DataFolderExists("root:Packages:Convert2Dto1D"))
        NewDataFolder root:Packages:Convert2Dto1D
    endif

    setDataFolder root:Packages:Convert2Dto1D

    Variable Htilt, Rot,i, j, pixelSize, lambda,xcenter,zcenter,Vmag,Vtilt,SDD

    // Calculate waves for the angle and q vectors; add them to the plots
    Vtilt=(-35)*pi/180//rotation along x axis
    Htilt=(0)*pi/180//rotation along z axis
    xcenter = 1993//beamCenterX in pixels
    zcenter = -1015//beamCenterY in pixels
    pixelsize = 816000//in Angstrom//PixelSizeX
    lambda = 1.239842//Wavelength in Angstrom
    SDD=2219000000//sample detector distance in Angstrom

    Make/O/N=(2304,2304) QMap,MagMap//,PicZ,PicPal, PicInplane
    Make/O/N=(2304,2304,3) PreMap,QcomponentMap
    Wave QMap,MagMap//,PicPal,PicInplane,PicZ
    Wave PreMap

    Make/O/N=3 Kin,Kout
    Wave Kin,Kout

    Kin[0]=0
    Kin[1]=-2*pi/lambda
    Kin[2]=0

    Make/O/N=3 V,V1, V2,V3
    Wave V,V1,V2,V3//vector define by the origin and points on the detector

    variable sinHtilt, cosHtilt, sinVtilt, cosVtilt
    sinHtilt = sin(Htilt)
    cosHtilt = cos(Htilt)
    sinVtilt = sin(Vtilt)
    cosVtilt = cos(Vtilt)

//  t2 = stopMsTimer(timerRefNum2)
//  print "initialization time = " + num2str(t2/1000000) + " seconds"

    //***  This section executes the tilt corrections.  It is the same as the SlowWAXSQMap, but is implemented using the p, q and r
    functions for speed
    //Also, this provides some flexibility.  For instance, you could add or subtract values to PreMap[][][1] to change the SDD, then
    only do the remaining operations to speed up analysis
    MultiThread PreMap[][][0]=cosHtilt*(p-xcenter)*pixelsize +sinHtilt*sinVtilt*(q-zcenter)*pixelsize
    MultiThread PreMap[][][1]=sinHtilt*(p-xcenter)*pixelsize+cosHtilt*(-sinVtilt)*(q-zcenter)*pixelsize -SDD
    MultiThread PreMap[][][2]=cosVtilt*(q-zcenter)*pixelsize
    MultiThread MagMap[][]=sqrt(PreMap[p][q][0]^2+PreMap[p][q][1]^2+PreMap[p][q][2]^2)
    MultiThread QcomponentMap[][][]=2*pi/Lambda*PreMap[p][q][r]/MagMap[p][q]-Kin[r]
    MultiThread QMap[][]=sqrt(QcomponentMap[p][q][0]^2+QcomponentMap[p][q][1]^2+QcomponentMap[p][q][2]^2)

    setDataFolder OldDf
//  t1 = stopmstimer(timerRefNum)
```

```
//	print "execution time = " + num2str(t1/1000000) + " seconds"
End
```

```
Function lnspacer(timewavetoscale, lowest, highest)
	wave timewavetoscale
	variable lowest, highest

	if(timewavetoscale[0] != 0)
		print "Time wave should start with 0, aborting operation!"
	endif

	variable lnhighest, lnlowest
	lnhighest = ln(highest)
	lnlowest = ln(lowest)

	variable lndifference = lnhighest -lnlowest
	variable spaces = numpnts(wavetoscale)
	variable i

	for(i=1;i<numpnts(wavetoscale);i+=1)
		timewavetoscale[i]= exp((lndifference/(numpnts(timewavetoscale)-2)*i + lnlowest))

	endfor
end
```

```
Function makeContin(ywave, twave, basis,header)  //formats data in the style necessary to make an input file for DISCRETE by Stephen Provencher.  The input format is esoteric, so this will help make the files properly.
    wave ywave, twave
    string basis, header
    variable numpts
    variable textfileref

    variable lnspaced = 1  //if this !=0, the data will be natural logarithmically-spaced

    PathInfo/S ContinPath
//  print S_path + " is the current data path"
    if(!V_Flag)
        print "activated"
        setContinPath()
    endif
    open/P=ContinPath textfileref as basis + "_IN.txt"

    variable nmax = 8000 //maximum number of points.  not sure what it is, exactly, but his code indicates 333 as a possibility, but that was in the 1980's...
    variable segmentlength, i

    if(numpnts(ywave) != numpnts(twave))
        print "Aborting - y-wave and t-wave are different sizes"
        return 0
    endif

    if(numpnts(ywave) > nmax)
        print "y-wave too large, filtering into equal-length segments, for a total size of " + num2str(nmax)
        print "Filtered data placed in two new waves - ywavecopy and twavecopy"
        numpts = nmax //this is to count the number of points used in the final output for the DISCRETE input file
        segmentlength = numpnts(ywave)/nmax  //determine spacing for newly filtered data to give even spacings up to nmax # of points
        make/O /D/N=(nmax) ywavecopy,twavecopy
        if(!lnspaced) //check to see if data should be log or linear spaced.  The first option is linear spacing
            for(i=0; i<nmax; i+=1)          //filter the data and place into the two new waves
                ywavecopy[i] = ywave[i * segmentlength]
                twavecopy[i] = twave[i*segmentlength]
            endfor
        else
            print "ln-spacing the data w.r.t. time with min value " + num2str(wavemin(twave)) + " and max value " + num2str(wavemax(twave))
            print "NOTE: it is assumed in this implementation that the data is more or less evenly spaced in time STARTING AT ZERO"
//              lnspacer(twavecopy, twave[1], wavemax(twave)) //re-scale twavecopy to be ln-spaced
            //          timewavetoscale[i]= exp((lndifference/(numpnts(timewavetoscale)-2)*i + lnlowest))
//          variable lnspacer = ln(wavemax(twave)-wavemin(twave))
            variable lnspacer = ln(numpnts(twave))
            twavecopy = 0
            ywavecopy[0]=ywave[0]
            for(i=1;i<nmax; i+=1) //the starting time element should be wavemin(twave), the only way to do it without getting -inf is to start counting at 1
                twavecopy[i]=twave[exp((lnspacer/(nmax-1))*i)]
                ywavecopy[i]=ywave[exp((lnspacer/(nmax-1))*i)]
            endfor
        endif
    else
//      print "The original input data is less than " + num2str(nmax) + " points, so the data is used as-input"
        duplicate/O ywave, ywavecopy //use the same wave names as prev. case, but in this case they are simply duplicates, since they were smaller than nmax
        duplicate/O twave, twavecopy
        numpts = numpnts(ywave)
    endif

    string commands, ydatastring, tdatastring //this is the order the info will be compiled, and ultimately printed to the Igor command line as "finaloutput"

    fprintf textfileref, header + "\r\n" //In Windows, this has to be added or it puts a little square instead of a carriage return

    fprintf textfileref, " LAST                1." + "\r\n"          //=1 if this is last data set
    fprintf textfileref, " GMNMX     1         1.E-5 " + "\r\n"      //lower limit for time constants
    fprintf textfileref, " GMNMX     2          1.E-1 " + "\r\n"      //upper limit for time constants
    fprintf textfileref, " IWT                      1." + "\r\n"       //error weighting to use - usually method 1
    fprintf textfileref, " NERFIT                  10." + "\r\n"          //number of points to use for error fitting
    fprintf textfileref, " IFORMY                     " + "\r\n"    //Format of the y-input (see next line)
    fprintf textfileref, " (5E15.6)                   " + "\r\n"           //Format 5E15.6 was also used in DISCRETE, so i'm using it here as the format is familiar to me now
    fprintf textfileref, " IFORMT                     " + "\r\n"    //Format of the t-input (see next line) -- after this section IFORMW can also be added if a weighting profile is to be specified
    fprintf textfileref, " (5E15.6)                   " + "\r\n"           //Format 5E15.6 was also used in DISCRETE, so i'm using it here as the format is familiar to me now
    fprintf textfileref, " IQUAD                   1.    " + "\r\n"  //Quadrature method.  Use 1 for exponential decays.  THIS IS NOT what his paper says, but this gives the correct results.  This will allow data to be fitted from the output directly.
    fprintf textfileref, " NONNEG                  1.    " + "\r\n"      //Specifies the time constants aren't negative.  Always true for exponential decay problems
    fprintf textfileref, " NINTT                   0. " + "\r\n"           //
    fprintf textfileref, " NLINF                   1." + "\r\n"            //
    fprintf textfileref, " NG                    500. " + "\r\n"           //Number of time constants to use
    fprintf textfileref, " END                        " + "\r\n"       //End statement for input
    fprintf textfileref, " NY    " +num2str(numpnts(ywavecopy)) + "\r\n"         //Number of points in the y-data set.  Number of points in y, t and w inputs must all be the same else it won't run

    variable j

    string tempstring = ""

    //This section adds the raw data to the output file.  It is identical to the DISCRETE file maker
        for(i=0;i<numpnts(twavecopy)/5; i+=1)    //this is to add a carriage return after every 5 entries
            tempstring = ""
            for(j=0;j<5;j+=1)
                sprintf tempstring, "%6.5E" twavecopy[5*i+j]
                fprintf textfileref, cleanstringfortran(spaces(5) + tempstring)
            endfor
                fprintf textfileref, "\r\n"
        endfor

        for(i=0;i<numpnts(ywavecopy)/5; i+=1)
            ydatastring = ""
            for(j=0; j<5; j+=1)
                tempstring = ""
                sprintf tempstring, "%6.5E" ywavecopy[5*i+j]
                ydatastring += spaces(5) + tempstring
                fprintf textfileref, cleanstringfortran(spaces(5) + tempstring)
            endfor
                fprintf textfileref, "\r\n"
        endfor

    Close textfileref
End
```

```
Function MakeSAXSMovie([StartNewMovie, CloseMovieWhenFinished])
    variable StartNewMovie, CloseMovieWhenFinished

    string oldDf=GetDataFOlder(1)
    setDataFolder root:Packages:Convert2Dto1D

    Wave ListOf2DSampleDataNumbers=root:Packages:Convert2Dto1D:ListOf2DSampleDataNumbers
    Wave/T ListOf2DSampleData=root:Packages:Convert2Dto1D:ListOf2DSampleData
    NVAR ImageRangeMin= root:Packages:Convert2Dto1D:ImageRangeMin     //images are manually scaled to have the same
    contrast, which is hard coded below
    NVAR ImageRangeMax = root:Packages:Convert2Dto1D:ImageRangeMax
    NVAR InvertImages = root:Packages:Convert2Dto1D:InvertImages
    SVAR ColorTableName=root:Packages:Convert2Dto1D:ColorTableName
    Wave CCDImageToConvert_dis=root:Packages:Convert2Dto1D:CCDImageToConvert_dis

    string SelectedFileToLoad
    variable basistimestamp, i, imax = numpnts(ListOf2DSampleDataNumbers)
    variable tempInvert = InvertImages
    InvertImages = 0 //forces the orientation to be "inverted" - which for ALS data usually has the beamstop facing downwards

//  Pathinfo/S PicturePath //initialization
//  NewPath/C/O/M="Select path to save pictures" PicturePath //creates the path
//  Pathinfo PicturePath //generates info from the path - not sure if this is redundant but it's how Nika does it
//  string PicturePathStr = S_path //this is generated by the Pathinfo operation

    SVAR MainPathInfoStr=root:Packages:Convert2Dto1D:MainPathInfoStr
    //print "path is " + MainPathInfoStr
    string FullPathName, timestampstr

    if(sum(ListOf2DSampleDataNumbers) < 1)
        print "No data selected.  Aborting"
        return -1
    endif

    variable j = 0
    For(i=0;i<imax;i+=1)//Just makes the movie and adds the first frame.
        if (ListOf2DSampleDataNumbers[i]) //Files selected in the "select input box" are given a 1, else 0
            SelectedFileToLoad=ListOf2DSampleData[i]          //this is the file selected to be processed
            GetFileFolderInfo /Q/P=Convert2Dto1DDataPath SelectedFileToLoad
//          NI1A_ButtonProc("DisplaySelectedFile")
            NI1A_ImportThisOneFile(SelectedFileToLoad)
            NI1A_DisplayLoadedFile()
//          NI1A_DoDrawingsInto2DGraph()
            TextBox/K/N=text0
            ImageRangeMin = 1.065//October data //1.21 Oct pub sundial
            ImageRangeMax = 2.566//October data  //2.44 Oct pub sundial      2.697//January data
            NI1A_TopCCDImageUpdateColors(0)
            ModifyGraph noLabel=2, tick=3, margin=5

            SetAxis/R left 1636,672 //October 2010 dataset
            SetAxis top 644,1660        //October 2010 dataset
            //SetAxis left 735,430
            if(InvertImages == 1)
            //  SetAxis bottom 150,450
            else
            //  SetAxis top 150,450
            endif

            if(j==0)
                BasisTimeStamp =  V_modificationDate//gets the absolute date, in seconds
                TimeStampStr = Secs2Time(0, 5) //sets  relative date to zero for 1st point
            else
                TimeStampStr =  Secs2Time((V_modificationDate-BasisTimeStamp),5) //gets the relative date, in seconds
            endif


            TimeStampStr += "\r" //+ SelectedFileToLoad //converts elapsed time to a date.  In this case, we are starting at 0
    because the first frame is at zero elapsed time
            TextBox /W=CCDImageToConvertFig/K/N=TimeStamp
            TextBox /W=CCDImageToConvertFig/N=TimeStamp/A=RB/F=0/B=0/G=(0,0,0) TimeStampStr
            DoUpdate

//          FullPathName =PicturePathStr + cleanupname(SelectedFileToLoad,0) + ".png"

//          SavePICT/C=1/O/E=-5 as FullPathName
            if(j==0)
                if(ParamIsDefault(StartNewMovie) || (StartNewMovie == 1)) //if StartNewMovie wasn't specified, or it was
    specified to = 1, then start a new movie file
                    NewMovie/A/F=1 //Must have a picture displayed before it will work!
                    addmovieframe
                else
                    addmovieframe  //assume a movie already exists, then add a frame.
                endif
            else
                addmovieframe
            endif

            j += 1
        endif
    endFor

    setDataFolder OldDf

    if(ParamIsDefault(CloseMovieWhenFinished) || (CloseMovieWhenFinished == 1)) //if CloseMovieWhenFinished wasn't
    specified, or it was specified to = 1, then close movie file
        CloseMovie
    endIf

    //MakeTimeStampWaves()
end
```

```
Function MakeSupPanel()
    string OldDf = GetDataFolder(1)

    if( !DataFolderExists("root:Packages"))
        NewDataFolder root:Packages
    endif

    if( !DataFolderExists("root:Packages:SuperPanel"))
        NewDataFolder root:Packages:SuperPanel
    endif

    setDataFolder root:Packages:SuperPanel

    //these are used by SuperPanel to track lists of files or traces that the user selects from
    make/O/T EChemDataList, CurNoteList, ContinDataList
    make/O EChemDataListIndices, ContinDataListIndices

    //Declare global variables
    variable /G ymultoffset, ylinoffset, xmultoffset,xlinoffset,colorspacing,AzimLeft,AzimRight,VertShift, BringToFront,
    ShowHiddenTraces
    variable /GshowVisibleTraces, AutoLoadCor, AlphaContin, Davg, GammaContin, PrefactorContin, OffsetContin, StartPtContin
    string /G EChemPathStr, ContinPathStr, DataFileExtension, ContinDataFileExtension,EChemWaveBasis, ContinWaveBasis
    string/G FittingResults,SelectedGraphName, MatchStr, MatchStr2

    //initialize some global variables
    DataFileExtension = ".csv"
    ContinDataFileExtension = ".txt"
    EChemWaveBasis = "test"
    showHiddenTraces = 0
    showVisibleTraces = 1
    ymultoffset = 1
    ylinoffset = 0
    xmultoffset=1
    xlinoffset=0
    colorspacing=1
    VertShift = 0.000

    //Makes a list of all open graphs and lists the traces if any graph is selected (ie: SuperPanel already existed and it's being re-
    opened)
    UpdateGraphList("makeSupPanelFunction")
    updatetracelist("makeSupPanelFunction")

    DoWindow/K SuperPanel//kills any existing panel
    string/g SelectedGraphName =""
    string listofcolortables = ctablist()
    variable numCTabs = itemsinlist(listofcolortables)
    make/o/T/N=(numctabs) ColorTableList = stringfromlist(p,listofcolortables)

//The following blocks lay out the panel and all of the objects on it
    //Outside of the tab region
    NewPanel /N=SuperPanel/W=(150,50,1000,500)
    TabControl SupPanelTabs,pos={330,10},size={500,360},proc=SupPanTabProc,tabLabel(0)="Editing",value=0
    TabControl SupPanelTabs, tabLabel(1)="Azimuthal",tabLabel(2)="E-Chem",tabLabel(3)="Analysis"
    TabControl SupPanelTabs,tabLabel(4)="Notes",tabLabel(5)="WAXS",tabLabel(6)="Contin",tabLabel(7)="Wave Ops"
    Button UpdateTracesButton title="Update trace list", proc=UpdateTraceList, pos = {10, 7}, size={125, 20}, help={"Refreshes list
    of traces for the selected graph."}
    Button SelectAllTracesButton title="Select/deselect all traces", proc=SelDeselAllTraces, pos = {145,7}, size={150,20}, help={"If
    all traces are selected, then all are deselected.  Else, all are selected"}
    ListBox TraceSelector pos = {10,30}, row=0, size={300,175}, mode=3,
    listwave=CurWindowTraceList,proc=SupPanelUpdateNote
    ListBox TraceSelector editStyle=1, selWave=CurWindowTraceListIndices, help={"Select waves to operate on. Select multiple
    waves with shift key"}
    CheckBox ShowHiddenTracesCheck title="Show hidden traces?", variable = ShowHiddenTraces,
    proc=SupPanelShowHiddenTracesCheck, pos={10, 210}, help={"Displays hidden traces if checked."}
    CheckBox ShowVisibleTracesCheck title="Show visible traces?", variable = showVisibleTraces,
    proc=SupPanelShowVisibleTracesCheck, pos={10, 225}, help={"Displays visible (un-hidden) traces if checked."}
    Button HideUnhideButton title="Hide/un-hide selected traces", proc=HideUnhideTraces, pos = {145,214}, size={150,20},
    help={"Switches hidden/visible property of each selected trace."}
```

```
Button UpdateGraphsButton title="Update graph list",proc=UpdateGraphList, pos = {20,260}, size={125,20}, help={"Refreshes list of graphs.  Does not happen automatically."}
ListBox GraphSelector listwave=CurGraphsList, mode=5,selWave=CurGraphsListIndices,pos={20,280},size={230,60}, proc=SupPanelListBoxProc, help={"Select a graph on which to operate"}
CheckBox BringToFrontCheck title="Bring to front?", variable = BringToFront, pos = {160, 260}, help={"If checked, graphs come into focus when selected from list."}
SetVariable SortTraceListKey title="Match string",pos ={145,240},size={130,20}, limits={-inf,inf,0},proc=SelectTracesByMatch, value=MatchStr,help={"Traces that match this string are selected."}

//tab0, "Editing"
Button ApplyScalingToSelection, title="Scale selection", proc=ScaleSelectionButton, pos={345, 50}, size={125,20}, help={"Applies linear and multiplicative offsets to selected traces."}
Button ApplyScalingToEverything, title = "Scale everything", proc=ScaleEverythingButton, pos={480,50}, size={125,20}, help={"Applies linear and multiplicative offsets to all traces on all graphs."}
Button ResetScalingParams, title = "Reset scaling params", proc=ResetScalingParamsButton, pos={675,87}, size={125,20}, help={"Resets linear and multiplicative offsets."}
SetVariable ymultoffset title="Y mult offset factor",size={150,20},value=ymultoffset,pos={345,80},proc=SupPanelSetScalingControls, help={"Y multiplier offset default 1.  Applies to selected waves in descending order, relative to first wave selected.  No autoreset."}
SetVariable ylinoffset title="Y lin offset factor",size={150,20},value=ylinoffset,pos={520,80},proc=SupPanelSetScalingControls, help={"Y linear offset default 0. Applies to selected waves in descending order, relative to first wave selected.  No autoreset."}
SetVariable xmultoffset title="X mult offset factor",size={150,20},value=xmultoffset,pos={345,100},proc=SupPanelSetScalingControls, help={"X multiplier offset default 0.  Applies to selected waves in descending order, relative to first wave selected.  No autoreset."}
SetVariable xlinoffset title="X lin offset factor",size={150,20},value=xlinoffset,pos={520,100},proc=SupPanelSetScalingControls, help={"X linear offset default 0. Applies to selected waves in descending order, relative to first wave selected.  No autoreset."}
Button ApplyColorsToSelection, title="Color selection", proc=ColorSelectionButton, pos={345, 160}, size={125,20}, help={"Applies color table to selected traces."}
Button ApplyColorsToEverything, title="Color everything", proc=ColorEverythingButton, pos={480,160}, size={125,20}, help={"Applies color table to all traces on all graphs."}
PopupMenu TraceColorPopup title = "Choose Trace Colors",  proc=SupPanelApplyTraceColorsPopup, value ="*COLORTABLEPOP*", pos={345,190}, help={"Applies colors from table equally spaced across color range.  Uncertain: problem when more traces size of table?"}


//tab1 "Azimuthal"
Button MakeAzimuthals title = "Make Azimuthal Waves", proc = SupPanelButton, pos ={345,40}, size={150,20},help={"Creates Azimuthal intensity plots.  First select 2D data in NIka main panel and enter left/right integration limits"}
SetVariable AzimLeftSetVar title="Left pixel limit",size={150,20},value=AzimLeft,pos={345,70},help={"Left integration limit.  Detemine by first making Sector Plot in Nika and choosing values around the region of interest."}
SetVariable AzimRightSetVar title="Right pixel limit",size={150,20},value=AzimRight,pos={520,70},help={"Right integration limit.  Detemine by first making Sector Plot in Nika and choosing values around the region of interest."}

//tab2 "E-Chem"
SetVariable EChemWaveBasis title="Wave Names Basis",live=1, size={200,20}, value=EChemWaveBasis,format="",limits={-inf,inf,0},pos={345,40},help={"Waves are generated with this as the prefix to their name."}
Button VMP3_load_CP title="Load VMP3 CP Data",size={130,20}, proc=VMP3_load_cp_button, pos={345,60},help={"Loads ChronoPotentiometry .mpt files from VMP3 EC-Lab.  WARNING header length changes in each vs of EC lab, check the header length against code in function!"}
Button ReMakeLnWaves title="Re-make -ln(abs()) copy", size={130,20},proc=ReMakeLnWaves,pos={680,60},help={"Creates -ln(abs(Voltage)) wave, overwrites existing if any"}
Button VMP3_load_CA title="Load VMP3 CA Data",size={130,20}, proc=VMP3_load_ca_button, pos={490,60},help={"Loads ChronoAmperometry .mpt files from VMP3 EC-Lab.  WARNING header length changes in each vs of EC lab, check the header length against code in function!!"}
Button Solartron_load_DC title="Load Solartron DC Data",size={150,20}, proc=Solartron_load_generic, pos={345,90},help={"Loads data taken from the DC Solartron.  Parses data into each channel (up to 8) which originally overlaps randomly."}
Button Solartron_load_COR    title="Load Solartron COR",size={150,20}, proc=load_solartron_multistat, pos={500,90},help={"Loads Solartron data from MultiStat software .cor extension"}
CheckBox AutoLoadCorBox title="Autoload COR?",value=1, variable=AutoLoadCor, pos={660,93}, help={"Auto name and load selected COR files?"}
Button GetEChemPathButton title="Select path for E-chem data",size={160,20}, proc=SupPanelGetEChemPath, pos={345,170},help={"Sets file-path for electrochemical data."}
SetVariable EChemSetFileExtension title="File Extension",pos ={510,170},size={130,20}, limits={-inf,inf,0}, value=DataFileExtension,help={"Data in the chosen directory is filtered by this file extension."}
TitleBox EChemPathTitleBox pos={345,200},variable=EChemPathStr
ListBox EChemFileList listwave=EChemDataList,mode=3,pos={345,220},size={450,125},selWave=EChemDataListIndices, help={"WARNING! Select only one file to load."}

//tab3 "Analysis"
```

```
    Button FitLinesButton title = "Fit lines", proc = MultiLineFit, pos ={345,40}, size={60,20},help={"Fits lines to the chosen waves
    between the cursors A and B.  Output is printed and added to the string named FittingResults"}
    Button ResetFitsButton title="Reset fits", proc = ResetFits, pos ={345,70}, size={60,20},help={"Removes ALL traces with the
    prefix fit_ and resets FittingResults to empty string."}
    Button FitExpsButton title = "Fit expontentials", proc = MultiExpFit, pos={415, 40}, size={120,20},help={"Fits exponentials to the
    chosen waves between the cursors A and B.  Output is printed and added to the string named FittingResults"}
    //tab4 "Notes"
    ListBox CurNoteListBox listwave=CurNoteList,mode=0,
    userColumnResize=1,selWave=$"",pos={345,40},size={450,310},selWave=EChemDataListIndices, help={"Displays any notes
    attached to the FIRST selected wave."}

    //tab5 "WAXS"
    Button MakeQMap title = "Make Q Map", proc = FastWAXSQMap, pos ={345,40}, size={120,20},help={"Creates a Q-Map using
    the input geometric parameters"}
    Button OverlayQMap title = "Overlay Q Map (Al std)", proc = OverlayQMap, pos ={345,70}, size={120,20},help={"Overlays the
    top image with the QMap showing Aluminum peaks according to the given calibration."}
    Button CopyWAXSMask title="Copy WAXS Mask", proc = CopyWAXSMask, pos ={345,100}, size={120,20},help={"First make
    a mask in Nika and apply it to an image. This copies that mask for WAXS analysis."}
    Button ConvertWAXSImages title="Convert WAXS Images", proc=ConvertWAXSImages, pos={345,130}, size={120,20},
    help={"Select 1 or more files in the Nika main window. QMap and Mask must be made first."}

    //tab6 "Contin"
    SetVariable ContinWaveBasis title="Wave Names Basis",size={200,20}, value=ContinWaveBasis,format="",limits={-
    inf,inf,0},pos={345,40},help={"Waves are generated with this as the prefix to their name."}
    Button Run_Contin title="Run Contin",size={130,20}, proc=RunContinButton, pos={345,60},help={""}
    SupPanTabProc("MakeSupPanel",0)      //Updates so that all controls are hidden except tab 0
    Button MinInRange title="Min in range",size={130,20}, proc=MinInRangeButton, pos={490,60},help={"Prints the minimum value
    from the selected range in the selected wave only.  This program can only accept positive input values."}
    Button ReverseSignButton title="Reverse sign",size={130,20}, proc=ReverseSignCtrl, pos={635,60},help={"Reverses the sign
    of the first selected wave.  Contin can only accept positive input values."}
    Button ReadContinButton title="Read Contin files", size={130,20}, proc=ReadContinButtonProc, pos={345, 145}, help={"Reads
    Contin output files"}
    TitleBox ContinPathTitleBox pos={345,200},variable=ContinPathStr
    Button GetContinPathButton title="Get contin path", size={130,20}, proc=SupPanelGetContinPath, pos={345,170}
    ListBox ContinFileList listwave = ContinDataList, mode=4,pos={345,220},size={450,125},selWave=ContinDataListIndices,
    help={"Select multiple Contin output files to load"}
    SetVariable ContinSetFileExtension title="File Extension",pos ={510,170},size={130,20}, limits={-inf,inf,0},
    value=ContinDataFileExtension,help={"Data in the chosen directory is filtered by this file extension."}
    Button TripExpFitButton title="Fit Triple Exponential", size={130, 20}, proc=TripExpFitButton, pos={345, 90},help={"Fits a triple
    exponential to the first selected trace."}
    Button ShiftVertical title="Apply Vertical
    Shift",size={150,20},proc=ManipulateTracesVertical,pos={360,120},help={"Permanently shifts the wave by the specified
    amount"}
    SetVariable VertShift title=" ",size={60,20}, value=VertShift, pos={520,120},help={"Sets value of shift to apply and (re)makes -
    ln(abs()) waves"}

    //tab7 "Wave Ops"
    ListBox TraceSelector2 pos = {340,30}, row=0, size={300,175}, mode=3, listwave=CurWindowTraceList, editStyle=1,
    selWave=CurWindowTraceListIndices2, help={"Select one wave as basis."}
    Button DivideByBasis pos={650, 30}, size={130,20}, proc=DivideByBasisButton, help={"Divide selected traces by selected
    basis."}, title="Divide by basis"
    Button MultiplyByBasis pos={650, 60}, size={130,20}, proc=MultiplyByBasisButton, help={"Multiply selected traces by selected
    basis."}, title="Multiply by basis"
    Button AddBasis pos={650, 90}, size={130, 20}, proc=AddBasisButton, help={"Add basis to selected traces."}, title="Add basis"
    Button SubtractBasis pos={650, 120}, size={130,20}, proc=SubtractBasisButton, help={"Subtract basis from selected traces."},
    title="Subtract basis"

    SetVariable SortTraceListKey2 title="Match string",pos ={345,215},size={130,20}, limits={-
    inf,inf,0},proc=SelectTracesByMatch2, value=MatchStr2,help={"Traces in above window that match this string are selected."}
    Button IntegrateSelection pos={345, 245}, size={130,20}, proc=Auto_Integrate_capacity, help={"Integrate selected waves"},
    title="Integrate"
    Button ConcatenateButton, title="Concatenate", pos={650,180},size={130,20},proc=AutoConcatenateWavesFromPanel,
    help={"Concatenate waves from two listbox selections.  Interleaves selection - be careful"}

    setDataFolder OldDf
End
```

```
Function MakeTimeStampWaves()
    string oldDf=GetDataFOlder(1)
    setDataFolder root:Packages:Convert2Dto1D

    Wave ListOf2DSampleDataNumbers=root:Packages:Convert2Dto1D:ListOf2DSampleDataNumbers
    Wave/T ListOf2DSampleData=root:Packages:Convert2Dto1D:ListOf2DSampleData

    string SelectedFileToLoad, TimeStampStr
    variable i, j, imax = numpnts(ListOf2DSampleDataNumbers)

    variable BasisTimeStamp, CurrentTimeStamp

    SVAR MainPathInfoStr=root:Packages:Convert2Dto1D:MainPathInfoStr
    //print "path is " + MainPathInfoStr
    string FullPathName

    if(sum(ListOf2DSampleDataNumbers) < 1)
        print "No data selected.  Aborting"
        return -1
    endif

    if( !DataFolderExists("root:Packages:SuperPanel:RelativeTimeWaves"))
        NewDataFolder root:Packages:SuperPanel:RelativeTimeWaves
    endif

    j=0
    string ScrapText = ""
    PutScrapText ScrapText

    For(i=0;i<imax;i+=1)//Just makes the movie and adds the first frame.
        if (ListOf2DSampleDataNumbers[i]) //Files selected in the "select input box" are given a 1, else 0
            SelectedFileToLoad=ListOf2DSampleData[i]        //this is the file selected to be processed

            string basis =  cleanupname(RemoveEnding(SelectedFileToLoad,".edf"),0)
            string TimeStampWaveName, TimeStampCompanionName
            TimeStampWaveName = "root:packages:SuperPanel:RelativeTimeWaves:TIME_"+basis[0,24]
            TimeStampCompanionName = "root:packages:SuperPanel:RelativeTimeWaves:Name_"+basis[0,24]

            GetFileFolderInfo /Q/P=Convert2Dto1DDataPath SelectedFileToLoad

            if(j==0)
                make/O/N=1 $TimeStampWaveName
                make/O/T/N=1 $TimeStampCompanionName
                wave TimeStampWave = $TimeStampWaveName
                wave/T TimeStampCompanionWave = $TimeStampCompanionName
                BasisTimeStamp =  V_modificationDate//gets the absolute time, in seconds
                TimeStampWave[0] = 0
                TimeStampCompanionWave[0] =  SelectedFileToLoad
                TimeStampStr = Secs2Time(0, 5)
            else
                CurrentTimeStamp = V_modificationDate
                CurrentTimeStamp = CurrentTimeStamp - BasisTimeStamp //converts to relative time, in seconds from the
    image note provided in Nika
                InsertPoints j, 1, TimeStampWave
                InsertPoints j, 1, TimeStampCompanionWave
                TimeStampWave[j] = CurrentTimeStamp
                TimeStampCompanionWave[j] =  SelectedFileToLoad
                TimeStampStr = Secs2Time(CurrentTimeStamp, 5)
            endif

            ScrapText += SelectedFileToLoad + "\t" + TimeStampStr + "\r"
            PutScrapText ScrapText
//          print SelectedFileToLoad + "\t" + TimeStampStr
            TimeStampStr += "\r" + SelectedFileToLoad //converts elapsed time to a date.  In this case, we are starting at 0
    because the first frame is at zero elapsed time
            j+=1
        endif
    endFor
    setDataFolder OldDf
end
```

```
Function ReadContin(basis, setNewPath, autoFileSelect)
    string basis
    variable setNewPath, autoFileSelect
    Variable refnum

    Print "Reading Contin output file " + basis + "_OUT.txt"

    PathInfo ContinPath

    if(autoFileSelect)
        Open/R/P=ContinPath refnum as basis + "_OUT.txt"
    else
        Open/R refnum as basis
    endif

    if (refnum == 0)
        return -1
    endif

    variable lineNumber, i
    string buffer
    variable foundChosen = 0

    lineNumber = 0
    variable chosenSolutionLine, dataReadFirstLine
    do
        FReadLine refnum, buffer
        if(stringMatch(buffer, "*CHOSEN SOLUTION*"))
            if(foundChosen == 0)
                foundChosen +=1
            else
                chosenSolutionLine = lineNumber
                //print "Chosen solution begins on line " + num2str(lineNumber)
                break
            endif
        endif

        lineNumber += 1
    while(1)

    do
        FReadLine refnum, buffer
        if(stringMatch(buffer, "*ORDINATE*"))
            lineNumber += 1
            dataReadFirstLine = lineNumber
            if(dataReadFirstLine - lineNumber != 0)
                print "Data read error: Chosen solution statement was on line " + num2str(chosenSolutionLine) + " and data read line (supposed to be line " + num2str(chosenSolutionLine + 7) + ") was on line " + num2str(dataReadFirstLine) + " instead\R ABORTING"
                return -1
            endif
            //print "Data begins on line " + num2str(lineNumber) + " which should be 7 lines beneath the chosen solution statement"
            break
        endif

        lineNumber += 1
    while(1)

    lineNumber += ReadOrdErrAbs(refnum,basis) //The distribution function is read here!!!

    string prefactorsStr, exponentsStr, errPrefactorsStr, errExponentsStr, EntSolExp, EntSolPre
    prefactorsStr = "Prefac_" + basis
    exponentsStr = "Exponents_" + basis
    errPrefactorsStr  = "ErrPref_" + basis
    errExponentsStr = "ErrExp_" + basis
//  EntSolExp = "FullSolnExp_" + basis
//  EntSolPre = "FullSolnPre_" + basis

    make/D/O/N=15 $prefactorsStr, $exponentsStr, $errPrefactorsStr, $errExponentsStr//, $EntSolExp, $EntSolPre
    wave Exponents = $exponentsStr
    wave Prefactors = $prefactorsStr
    wave ErrPrefactors = $errPrefactorsStr
    wave ErrExponents = $errExponentsStr
//  wave EntireSolExponent = $EntSolExp
//  wave EntireSolPrefactor = $EntSolPre
    Exponents = 0
    Prefactors = 0
    ErrPrefactors = 0
    ErrExponents = 0
    variable dat1, dat2, dat3, dat4, dat5
    variable peakNum = 0
    for(i=0; i<98; i+=1) //this searches for a max number of peaks equal to 15 (6 lines/peak output) + 8 lines for "moments of entire
    solution" output
        FReadLine refnum, buffer
        lineNumber += 1

        if(stringMatch(buffer, "*MOMENTS OF ENTIRE SOLUTION*")) //this adds a blank line of zeroes to delineate the entire
    solution moments
            Prefactors[peakNum] = 0
            ErrPrefactors[peakNum] = 0
            Exponents[peakNum] = 0
            ErrExponents[peakNum] = 0
            if(peakNum <=6)
                peakNum = 7
            else
                peakNum += 1
            endif
        endif

        if(stringMatch(buffer, "* 0 *"))   //these lines in the text file, j = 0, hold the prefactor moment
            sscanf buffer, "                              0    %e X (10**  %e)       %e", dat1, dat2, dat3 //dat1 * 10^dat2 is the
    pre-exponent, and dat3 is its error
            Prefactors[peakNum] = dat1 * 10 ^ dat2
            ErrPrefactors[peakNum] = dat3
//          print "prefactor = " + num2str(dat1*10^dat2)
        endif
        if(stringMatch(buffer, "* 1 *"))   //these lines in the text file, j = 1, hold the exponential moment
            sscanf buffer, "                              1    %e X (10**  %e)       %e         %e        %e", dat1, dat2, dat3, dat4,
    dat5 //only interest is dat4 and dat5, which contain the exponent and its error
            Exponents[peakNum] = dat4
            ErrExponents[peakNum] = dat5
            peakNum += 1

        endif

    endFor
    Close refnum
    NVAR GammaContin
    NVAR PrefactorContin
    GammaContin = Exponents[7]
    PrefactorContin = Prefactors[7]

//  print "Apparently successful data read: finished on line  " + num2str(lineNumber)
//  print "Total solution moments for " + basis + ": Exponent \t" + num2str(Exponents[7]) + "\t Prefactors \t" +
    num2str(Prefactors[7])
End
```

```
Function/D ReadOrdErrAbs(refnum, basis) //returns number of lines read from the file indicated by refnum
    variable refnum
    string basis
    variable lineNumber
    string buffer
    variable index
    variable dat1, dat2, dat3

    string OffsetCoefficientStr = "Offset_" + basis
    string OrdWaveStr = "Ordinates_" + basis
    string ErrWaveStr = "Error_" + basis
    string AbsWaveStr = "Abscissa_" + basis

    for(lineNumber=0; 1; lineNumber+=1)
        index = lineNumber +1  //points are inserted before "index" for OrdWave, ErrWave and AbsWave

        FReadLine refnum, buffer


        if(stringMatch(buffer, "*LINEAR COEFFICIENTS*"))
            variable linOffset
            sscanf buffer, "0LINEAR COEFFICIENTS =   %e", linOffset
            make/D/O/N=1 $OffsetCoefficientStr
            wave OffsetCoefficient = $OffsetCoefficientStr
            OffsetCoefficient[0] = linOffset
            NVAR OffsetContin
            OffsetContin = linOffset
//          print "linear offset coefficient = " + num2str(linOffset)
            return lineNumber
        endif

        if(lineNumber == 0) //this block either makes waves, or adds points to them
            make/D/O/N=1 $OrdWaveStr, $ErrWaveStr, $AbsWaveStr
            wave OrdWave = $OrdWaveStr
            wave ErrWave = $ErrWaveStr
            wave AbsWave = $AbsWaveStr
        else
            insertPoints index, 1, OrdWave
            insertPoints index, 1, ErrWave
            insertPoints index, 1, AbsWave
        endif

        buffer = ReplaceString("D", buffer, "E")
//      print "buffer = " + buffer
        sscanf buffer, "   %e  %e  %e", dat1, dat2, dat3
//      print "line number = " + num2str(lineNumber)
//      print "dat1 = " + num2str(dat1) + " dat2 = " + num2str(dat2) + " dat3 = " +  num2str(dat3)

        OrdWave[lineNumber] = dat1
        ErrWave[lineNumber] = dat2
        AbsWave[lineNumber] = dat3
    endFor
End
```

```
Function SaveSAXSImagesToFile()

    string oldDf=GetDataFOlder(1)
    setDataFolder root:Packages:Convert2Dto1D
    Wave ListOf2DSampleDataNumbers=root:Packages:Convert2Dto1D:ListOf2DSampleDataNumbers
    Wave/T ListOf2DSampleData=root:Packages:Convert2Dto1D:ListOf2DSampleData
    string SelectedFileToLoad
    variable i, imax = numpnts(ListOf2DSampleDataNumbers)

    NVAR ImageRangeMin= root:Packages:Convert2Dto1D:ImageRangeMin
    NVAR ImageRangeMax = root:Packages:Convert2Dto1D:ImageRangeMax
    NVAR InvertImages = root:Packages:Convert2Dto1D:InvertImages
    SVAR ColorTableName=root:Packages:Convert2Dto1D:ColorTableName
    Wave CCDImageToConvert_dis=root:Packages:Convert2Dto1D:CCDImageToConvert_dis
    string/G PicturePathStr
    Pathinfo/S PicturePath //initialization
    NewPath/C/O/M="Select path to save pictures" PicturePath //creates the path
    Pathinfo PicturePath //generates info from the path - not sure if this is redundant but it's how Nika does it
    PicturePathStr = S_path //this is generated by the Pathinfo operation

    variable tempInvert = InvertImages
    InvertImages = 0
    string fullPathName

    For(i=0;i<imax;i+=1)
        if (ListOf2DSampleDataNumbers[i]) //Files selected in the "select input box" are given a 1, else 0
            SelectedFileToLoad=ListOf2DSampleData[i]          //this is the file selected to be processed
            NI1A_ImportThisOneFile(SelectedFileToLoad)
            NI1A_DezingerDataSetIfAskedFor("CCDImageToConvert")
            NI1A_DisplayLoadedFile()
            //NI1A_DisplayStatsLoadedFile("CCDImageToConvert")
            NI1A_DoDrawingsInto2DGraph()
            TextBox/K/N=text0
            ImageRangeMin = 0.3
            ImageRangeMax = 3.25
            NI1A_TopCCDImageUpdateColors(0)
            SetAxis left 830,340
            SetAxis top 55, 545
//          NI1M_DisplayMaskOnImage()
//          NI1M_ChangeMaskColor("black")
            ModifyGraph tick=3,mirror=0,noLabel=2, margin = 1, width=288, height=288
            DoUpdate
            FullPathName =PicturePathStr + removeending(SelectedFileToLoad, ".edf") + ".tif"
            SavePICT/O/E=-7/B=600 as FullPathName
        endif
    endfor
    setDataFolder OldDf

    InvertImages = tempInvert

end
```

```
Function SlowWAXSQMap(ctrlname)
    //Scott Mullin, December 2009, samullin@gmail.com
    //Same result as FastWAXSQMap() - generates QMap for later 2D-1D data reduction
    //Implementation uses for-loops rather than "Wave Arithmetic", so it runs a lot slower

    string ctrlname
    string OldDF=GetDataFolder(1)
    setDataFolder root:Packages:Convert2Dto1D

    variable timerRefNum = StartMSTimer
    variable t1

    Variable i, j, pixelSize, lambda,xcenter,zcenter,Vmag,Vtilt,SDD, Htilt

    // Calculate waves for the angle and q vectors; add them to the plots
    Vtilt=(-35)*pi/180//rotation along x axis
    Htilt=(0)*pi/180//rotation along z axis
    xcenter = 1993//beamCenterX in pixels
    zcenter = -1015//beamCenterY in pixels
    pixelsize = 816000//in Angstrom//PixelSizeX
    lambda = 1.239842//Wavelength in Angstrom
    SDD=2219000000//sample detector distance in Angstrom

    Make/O/N=(2304,2304) QMap
    Wave QMap

    Make/O/N=3 Kin,Kout
    Wave Kin,Kout

    Kin[0]=0
    Kin[1]=-2*pi/lambda
    Kin[2]=0

    Make/O/N=3 V,V1, V2,V3
    Wave V,V1,V2,V3//vector define by the origin and points on the detector

    Make/O/N=3 Qa //scattering vector Ko-Ki
    Wave Qa

    variable sinHtilt, cosHtilt, sinVtilt, cosVtilt
    sinHtilt = sin(Htilt)
    cosHtilt = cos(Htilt)
    sinVtilt = sin(Vtilt)
    cosVtilt = cos(Vtilt)

    QMap = 0

    For (i=0;i<2304;i+=1)
        For (j=0;j<2304;j+=1)
            //rotatation of Axis using Euler Angle according to ZXZ' convention
            V[0]=(i-xcenter)*pixelsize //Vectors defined using arbitrary detector axis
            V[1]=0
            V[2]=(j-zcenter)*pixelsize

            V1[0]=V[0] //V1: rotate V along X axis with angle Vtilt
            V1[1]=0-sinVtilt*V[2]
            V1[2]=0+cosVtilt*V[2]

            V3[0]=cosHtilt*V1[0]-sinHtilt*V1[1] //V3: rotate  V2 along Z axis with angle Htilt
            V3[1]=sinHtilt*V1[0]+cosHtilt*V1[1]-SDD//throw in the SDD correction here.  Ideally this could be done on the whole
                2-D map at the very end
            V3[2]=V1[2]

            Vmag=sqrt(V3[0]^2+V3[1]^2+V3[2]^2)

            Kout=2*pi/Lambda*V3/Vmag

            Qa = Kout-Kin

            QMap[i][j]=sqrt(Qa[0]^2+Qa[1]^2+Qa[2]^2)
```

```
		endfor
	endfor
	t1 = stopmstimer(timerRefNum)
	print "execution time = " + num2str(t1/1000000) + " seconds"

	setDataFolder OldDf
End
```

```
Function Solartron_load_generic(ctrlname)
    string ctrlname

    string OldDF = GetDataFolder(1)
    SetDataFolder root:Packages:SuperPanel

    string columnInfo
    variable headerline, startline

    headerline = 4
    startline = 5
    wave/T EChemDataList
    wave EChemDataListIndices
    SVAR pathname = EChemPathStr
    SVAR basis = EChemWaveBasis

    string FileToLoad

    variable i
    for(i=0; i<dimsize(EChemDataListIndices,0); i+=1)
        if(EChemDataListIndices[i]==1)
            FileToLoad = EChemDataList[i]
        endif
    endfor

    columnInfo =
    "N='_skip_';N="+basis+"_h;N='_skip_';N='_skip_';N="+basis+"_Ch;N="+basis+"_step;N="+basis+"_rep;N='_skip_';N=" + basis
    + "_V;N=" + basis + "_mA;N='_skip_';N='_skip_';N='_skip_';N='_skip_';N='_skip_';N="+basis+"_mAh;"
    string fullpathname = pathname + filetoload
    print "loading " + fullpathname
    LoadWave/A/J/D/B=columnInfo/D/O/K=1/V={","," $",0,0}/L={headerline,startline,0,0,0} fullpathname

    //these are the master waves that are loaded from the file, and include data from all channels run dumped into a single list in
    the order they were generated
    wave TimeWave = $(basis+"_h")
    wave ChannelWave = $(basis + "_Ch")
    wave StepWave = $(basis + "_step")
    wave RepeatWave = $(basis + "_rep")
    wave VoltageWave = $(basis + "_V")
    wave CurrentWave = $(basis + "_mA")
    wave CapacityWave = $( basis + "_mAh")

//  edit TimeWave, ChannelWave, StepWave, RepeatWave, VoltageWave, CurrentWave, CapacityWave //diagnostic to see
    loaded data

    CurrentWave = CurrentWave * 1000 //converts from A to mA
    CapacityWave = CapacityWave * 1000 //converts from Ah to mAh
    TimeWave = TimeWave/3600 //converts from seconds to hours
    variable TotalPoints = numpnts(ChannelWave)      //the program steps through the list of data, using this wave to figure out
    which channel the data is from


//  make/T/O/N=0 MasterList
    string MasterList = ";" //initialize it so it isn't null during the first WhichListItem test below
    variable j
    for(i=0; i< TotalPoints; i+=1)
        string OperatingWaveName
        OperatingWaveName = basis + "_Ch" + num2str(ChannelWave[i]) + "_St" + num2str(StepWave[i]) + "_Rp" +
    num2str(RepeatWave[i])
        if(WhichListItem(OperatingWaveName, MasterList) == -1) //name is checked to see if it's new (ie: it is listed on the
    MasterList)
            Make/D/O/N=0 $(OperatingWaveName + "_h"), $(OperatingWaveName + "_V"), $(OperatingWaveName + "_mA"),
    $(OperatingWaveName + "_mAh")
            MasterList = MasterList + OperatingWaveName + ";" //add the new name to the list -- if the item was already listed
    the parent IF statement doesn't execute this code
        endif

        wave OperatingTimeWave = $(OperatingWaveName + "_h")
        wave OperatingVoltageWave = $(OperatingWaveName + "_V")
        wave OperatingCurrentWave = $(OperatingWaveName + "_mA")
        wave OperatingCapacityWave = $(OperatingWaveName + "_mAh")

        variable insertionIndex = numpnts(OperatingTimeWave) //the insertion index is coming from OperatingTimeWave but should be the same for all waves in this "set"
        insertpoints insertionIndex, 1, OperatingTimeWave
        insertpoints insertionIndex, 1, OperatingVoltageWave
        insertpoints insertionIndex, 1, OperatingCurrentWave
        insertpoints insertionIndex, 1, OperatingCapacityWave

        OperatingTimeWave[insertionIndex] = TimeWave[i]
        OperatingVoltageWave[insertionIndex] = VoltageWave[i]
        OperatingCurrentWave[insertionIndex] = CurrentWave[i]
        OperatingCapacityWave[insertionIndex] = CapacityWave[i]
    endfor

    variable k
    string MasterList2 = ";"
    for(k=1; k<(itemsinlist(MasterList)); k+=1) //MasterList has a leading ";", so that is ignored by starting at k=1 and subtracting it from itemsinlist
        wave OperatingTimeWave = $(stringfromlist(k, MasterList) + "_h") //reference the MasterList to determine which waves exist -- each should have a _h, _V, _mA and _mAh variant unless there was an error above
        wave OperatingVoltageWave = $(stringfromlist(k, MasterList) + "_V")
        wave OperatingCurrentWave = $(stringfromlist(k, MasterList) + "_mA")
        wave OperatingCapacityWave = $(stringfromlist(k, MasterList) + "_mAh")

        variable StartingCapacity = OperatingCapacityWave[0]
        OperatingCapacityWave -= StartingCapacity //this zeroes the capacity waves for the entire segments

        string CurrentListItem = stringfromlist(k, MasterList)
        variable index1 = strsearch(CurrentListItem, "_St",0)+3
        variable index2 = strsearch(CurrentListItem, "_Rp",0)-1
        variable CurrentStepNumber = str2num(CurrentListItem[index1, index2])
        variable index3 = strsearch(CurrentListItem, "_Ch",0)+3
        variable index4 = strsearch(CurrentListItem, "_St",0)-1
        variable CurrentChNumber = str2num(CurrentListItem[index3, index4])

        string OperatingWaveName2 = "Cap_" + basis + "_Ch" + num2str(CurrentChNumber) + "_St" + num2str(CurrentStepNumber) + "_mAh"
        if(WhichListItem(OperatingWaveName2, MasterList2) == -1) //name is checked to see if it's new (ie: it is listed on the MasterList)
            Make/D/O/N=0 $(OperatingWaveName2)
            MasterList2 = MasterList2 + OperatingWaveName2 + ";" //add the new name to the list -- if the item was already listed the parent IF statement doesn't execute this code
        endif

        wave OperatingWave = $(OperatingWaveName2)
        variable insertionIndex2 = numpnts(OperatingWave) //the insertion index is coming from OperatingTimeWave but should be the same for all waves in this "set"
        insertpoints insertionIndex2, 1, OperatingWave
        OperatingWave[insertionIndex2] = abs(OperatingCapacityWave[numpnts(OperatingCapacityWave)-1])


//      print "Capacity (mAh) \t" + nameofwave(OperatingCapacityWave) +"\t"+ num2str(OperatingCapacityWave[numpnts(OperatingCapacityWave)-1]) + "\r"

        doWindow VoltagePlots
        if(V_Flag ==1 || V_Flag ==2)   //this means window exists
            doWindow VoltagePlots
            appendtograph/W=VoltagePlots OperatingVoltageWave vs OperatingTimeWave //voltage curves go on the L axis
            appendtograph/W=VoltagePlots/R OperatingCurrentWave vs OperatingTimeWave
        else
            Display /N=VoltagePlots OperatingVoltageWave vs OperatingTimeWave
            appendtograph/W=VoltagePlots/R OperatingCurrentWave vs OperatingTimeWave
            Label bottom "Time (hours)"
            Label left "Voltage (V)"
            Label right "Current (mA)"
        endif

        doWindow CapacityOverTime
        if(V_Flag ==1 || V_Flag ==2)   //this means window exists
```

```
            doWindow CapacityOverTime
            appendtograph/W=CapacityOverTime OperatingCapacityWave vs OperatingTimeWave //capacity curves go on the L axis
            appendtograph/W=CapacityOverTime/R OperatingCurrentWave vs OperatingTimeWave
        else
            Display /N=CapacityOverTime OperatingCapacityWave vs OperatingTimeWave
            appendtograph/W=CapacityOverTime/R OperatingCurrentWave vs OperatingTimeWave
            Label bottom "Time (hours)"
            Label left "Capacity (mAh)"
            Label right "Current (mA)"
        endif
    endfor

    if(cmpstr(ctrlname, "diffusion"))
        variable h
        for(h=1; h<(itemsinlist(MasterList2)); h+=1) //MasterList has a leading ";", so that is ignored by starting at k=1 and subtracting it from itemsinlist
            string CurrentListItem2 = stringfromlist(h, MasterList2)
            variable index5 = strsearch(CurrentListItem2, "_Ch",0)+3
            variable index6 = strsearch(CurrentListItem2, "_St",0)-1
            string CurrentChNumber2 = CurrentListItem2[index5, index6]
            wave OperatingWave = $(stringfromlist(h, MasterList2))
            string CapacityTableName = "CapacityTableCh" + CurrentChNumber2 // Makes separate table for each channel
            doWindow $CapacityTableName
            if(V_Flag ==1 || V_Flag ==2)   //this means window exists
                doWindow $CapacityTableName
                appendToTable/W=$CapacityTableName OperatingWave //capacity curves go on the L axis
            else
                Edit /N=$CapacityTableName OperatingWave
            endif

            doWindow CapacityVersusRepeatNumber
            if(V_Flag ==1 || V_Flag ==2)   //this means window exists
                doWindow CapacityVersusRepeatNumber
                appendtograph/W=CapacityVersusRepeatNumber OperatingWave //capacity curves go on the L axis
            else
                Display /N=CapacityVersusRepeatNumber OperatingWave
                Label bottom "Cycle number"
                Label left "Capacity (mAh)"
            endif
        endfor
    endif

    SetDataFolder OldDF
end
```

```
Function/s Spaces(numspaces)
        variable numspaces
        variable i
        string spacesoutput = ""

        for(i=0; i<numspaces; i+=1)
                spacesoutput += " "
        endfor

        return spacesoutput
end
```

```
Function WAXS2Dto1D(ctrlname) //Compares 2D detector image to Qmap to generate a 1D wave
    //Scott Mullin, December 2009, samullin@gmail.com
    //Does not count areas masked by MaskImage
    //Normalizes by the number of points at each q-value to account for masked areas, detector tilt, detector cutoff, etc.
    string ctrlname

    string OldDF=GetDataFolder(1)
    setDataFolder root:Packages:Convert2Dto1D

    Variable i,j, a, x,Qmin, Qmax,maskCounter
    Wave QMap
    Qmin=waveMin(QMap)
    Qmax=waveMax(QMap)
    Make/O/N=1000 WAXSIntensity, xwave, normalizer
    normalizer = 1

    Setscale/I x,Qmin,Qmax,""WAXSIntensity
    Setscale/I x,Qmin,Qmax,""xwave
    Setscale/I x,Qmin,Qmax,""normalizer

    Wave WAXSIntensity, xwave
    WAXSIntensity=1
    wave ccdimagetoconvert //this is generated by Nika when you "display" an image .. defaults to last image made
    wave WAXSMaskImage


    For (i=0;i<2304;i+=1)
        For (j=0;j<2304;j+=1)
            if (WAXSMaskImage[i][j] != 0)
                a=QMap[i][j]     //check map to find corresponding Q value for the given pixel
                x=x2pnt(xwave, a) //find the closest Q-value on the above-defined scale
                normalizer[x] += 1 //at end, normalize by this to adjust for masked points, etc
                WAXSIntensity[x]+=ccdimagetoconvert[i][j] //final output, plot by itself, scaling is self-contained in the wave
                    description
            else
                maskCounter+=1
            endif
        endfor
    endfor

    WAXSIntensity=WAXSIntensity/normalizer
    printf "fraction area masked is: %g\r" maskCounter/(2304*2304)
    setDataFolder OldDf
End
```

Appendix C – Selected LabVIEW Examples

In this Thesis, custom LabVIEW programs were used extensively for data logging and instrument control. Not all of the LabVIEW vi's (virtual instruments - the name of LabVIEW programs) or sub-vi's are detailed here. Instead, this Appendix is designed to illustrate the key functionalities for a few of the vi's. This Appendix is written with the assumption that the reader is familiar with LabVIEW. In order to learn LabVIEW, several excellent resources exist, including the National Instruments web forums, the LabVIEW manual, and built-in LabVIEW examples. These vi's were all written and tested in LabVIEW 2010. All of them follow similar protocols for graphing and data logging, the features of which are detailed in the next section, "16-Channel Thermocouple Reader Program". These are the most modern variants of the programs I wrote during my PhD, however they are by no means "complete" in the sense of professionally-written programs. Each was written "organically" as the important concepts were learned, implemented and improved upon.

The following sections describe the chosen LabVIEW programs, including their purpose, simple instructions for using them, select details about their implementation, and caveats related to their use, including known bugs.

16-Channel Thermocouple Reader Program

The thermocouple reader program interfaces with a National Instruments thermocouple reader, model number ENET-14A384D. This device accommodates up to 16 separate thermocouple inputs. Physically, each of the 16 input terminals is wired with K-type thermocouple wire to K-type thermocouple ports, all of which are mounted side-by-side on a rail. This enables a user to readily plug K-type thermocouples into any or all of the ports. The device is Ethernet-equipped. The device is found and configured in LabVIEW using the Measurement and Automation Explorer (MAX). If connection problems arise, the device should be restarted. If Ethernet connectivity issues arise, use MAX and attempt to communicate with and reset the device. For persistent connectivity issues, try directly connecting the device to the LabVIEW computer, or connect both the device and the computer to a shared, managed router. National Instruments technical support is also valuable with this type of issue. Once the device connection is established through MAX, the program can be run. As long as no connectivity issues arise, it is unnecessary to use MAX – even through reboots of the computer.

The front panel of this vi is shown in Figure 39. When the program first runs, it automatically starts reading temperatures from all 16 channels and asks the user for a directory in which to store data files. However, the temperature data points are not logged to file or graphed until the "Acquire" button is enabled for the chosen channel. Once the "Acquire" button is enabled for a given channel, a new file is automatically generated and given a name with the format, "M_D_YYYY_Reader_ChX_000.txt", where the initial portion is the date, X is the channel number, and the suffix indicates the file number. If the "Acquire" button is disabled, and then later enabled, a new file is generated with the same format but the suffix is incremented. For instance, if the date is August 2, 2011 and data is acquired on channel 16, the first file will be named "8_2_2011_Reader_Ch16_000.txt", the second will be named "8_2_2011_Reader_Ch16_001.txt". The program automatically increments the suffix until an available number is found. This could lead to weird behavior if earlier files are deleted. It could also cause a crash if more than 1000 data files are generated on a given day for a given channel – a highly unlikely scenario. If the program is restarted and a different directory is chosen, the suffix starts over at 000. Similarly, the suffix starts over at 000 if the date changes. This approach is used for all of the LabVIEW programs outlined in this Appendix since it simplifies the collection and categorization of numerous, disparate data-log-files.

This program contains a few controls. The "Acquire" buttons are unique to each channel, although in the block diagram they are grouped into a 16-element array, and their behavior is described above. The "Comment" input boxes are implemented similarly. The text input to each channel's "Comment" box goes into the first line of the header when the file is written. The "Minimum time between file writes (seconds)" is used to regulate how often data is logged. The program executes continuously and the only timing constraints are the speed at which the LabVIEW program executes and the speed at which the thermocouple inputs can be read. The "Minimum time" sets a lower limit on how often the program can write to file. This helps keep data files to a manageable size. The default is 10 s. Finally, the "Clear graph?" control will clear any data from the graph for an actively graphing channel. There is a bug in the implementation such that any channels that are not actively graphing are not cleared when this button is pressed.

This bug can prevent the graph from ever being totally cleared, but the functionality is still useful.

The program also includes several displays. The "T (Celsius)" box displays the thermocouple readings. Every time the program loops, data from all 16 channels is read and output to this indicator. If the thermocouple is disconnected, a reading is still taken, but the value is very high (on the order of 2400 °C). If the reading is above 300 °C, the program assumes the reading is in error and the "Connected?" indicator light is turned off. Otherwise the "Connected?" indicator light is turned on. This indicator helps trouble-shoot thermocouple connectivity issues. Furthermore, if the reading is detected as disconnected, the program will not graph the data point or log it to file. This helps keep the graph from auto-scaling to an overly-high value and keeps the data files clean. If a channel is detected as having a good connection and the corresponding "Acquire?" button is enabled, the reading will be logged to file at the rate governed by "Minimum time between file writes" and will be graphed at the same rate at which the data is acquired.

The inner portion of the program is "open-loop". See the portion inside the loop in Figure 40 for details. Every time the outer loop executes, the thermocouple readings are taken from all 16 channels and the results are displayed to the indicators, written to file, and/or graphed according to the control settings as described above. The "Clear graph?" button latches when pressed, such that it is read during the subsequent loop execution. The time between file writes is governed by a counter that tracks the time since the last data point was written to file. Due to the "open-loop" configuration, the time between file writes responds in real-time to the control setting. For instance, if the control is changed from 0 seconds (writing as fast as data is collected) to 10 seconds, the file will have at least a 10 s gap between data points. Correspondingly, if the control is changed from 10 s to 0 s, the data will suddenly write to file much faster. All of the key LabVIEW vi's in this Appendix were built as open-loop to improve responsiveness and modularity.

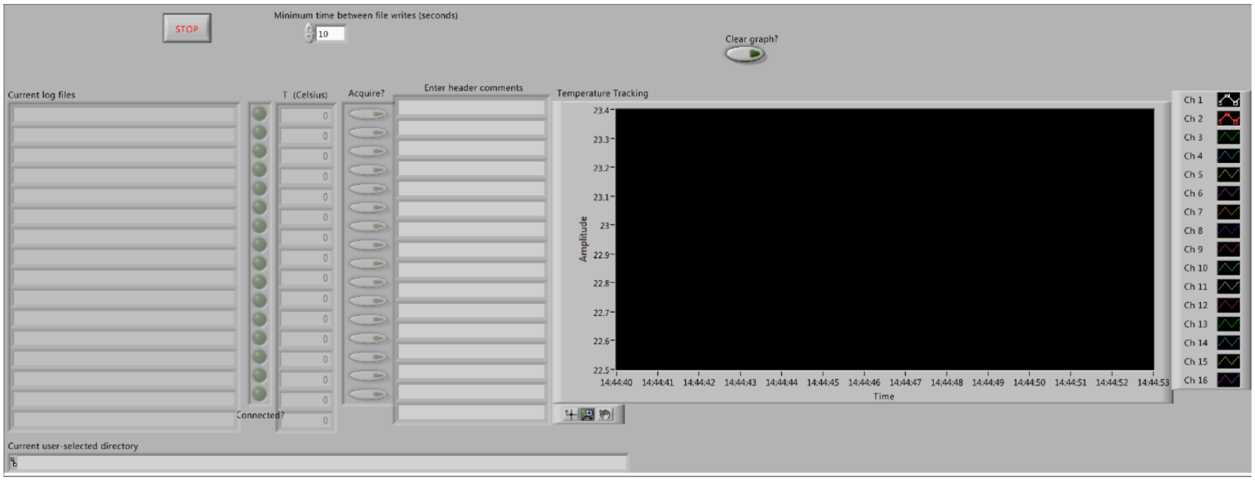

Minimum time between file writes (seconds): 10

Figure 39 – Thermocouple reader front panel.

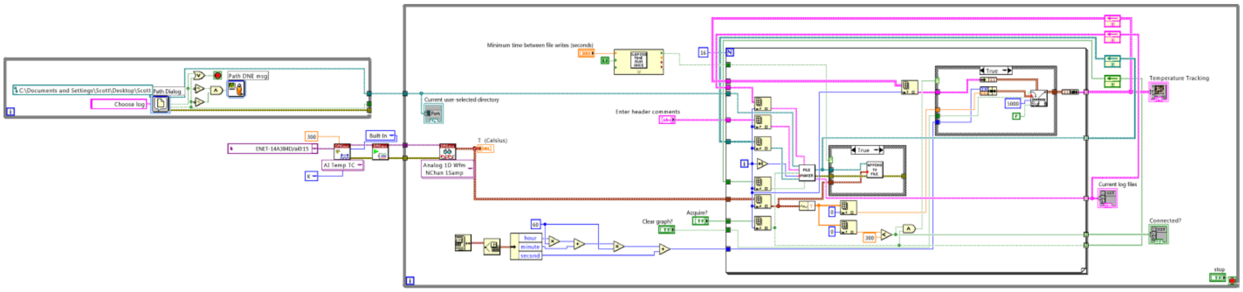


Figure 40 – Thermocouple reader block diagram.

The block diagram in Figure 40 contains several custom sub-vi's. The vi labeled "File Maker" is shown in Figure 41. This sub-vi detects whether the "Record?" input is true (ie, "Acquire?" is enabled). It checks if the previous state was "disabled", which indicates that a new file should be written. If a new file is to be written, the sub-vi generates a new file with a name as described in the paragraphs above and passes along the file path. If "Record?" is true, but no new file is needed, then the previous file path is simply passed along. Variants of this design are used in all LabVIEW programs in this Appendix.

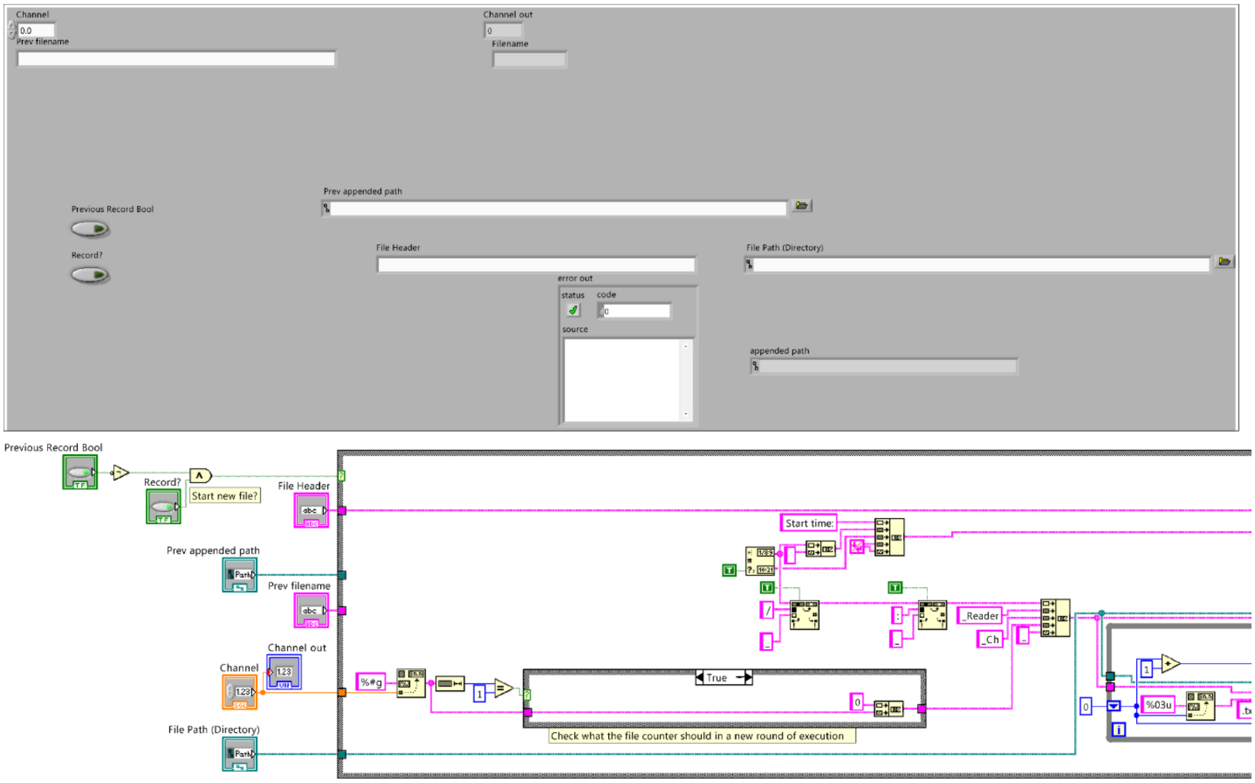


Figure 41 – File Maker sub-vi.

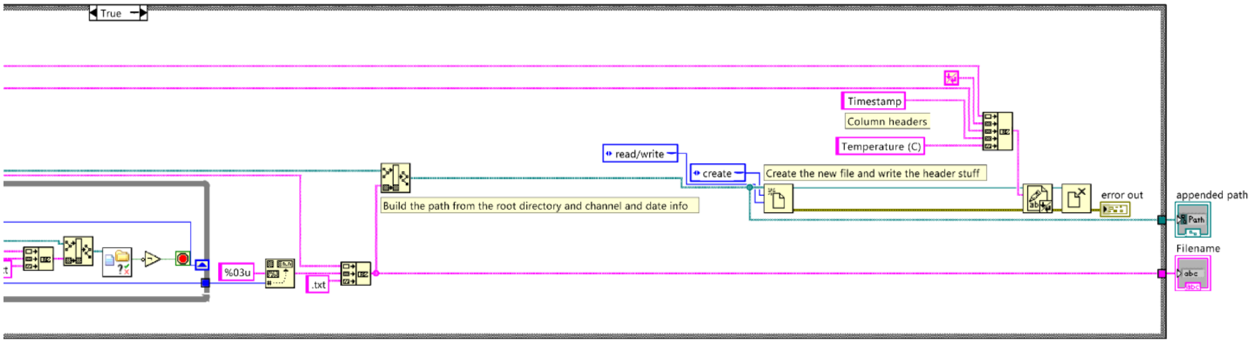

True
Timestamp
Column headers
Temperature (C)
read/write
create
Create the new file and write the header stuff
Build the path from the root directory and channel and date info
error out
appended path
Path
Filename
%03u
.txt

Figure 42 – File Maker sub-vi, part 2.

The “Append to File” sub-vi is described next, the block diagram of which is shown in Figure 43. This sub-vi takes the file-path generated by the “File Maker” sub-vi, a time-stamp which is generated when the thermocouple readings are taken, and a thermocouple reading. The thermocouple reading and timestamp are then appended to the file. This sub-vi only executes if the “Minimum time between file writes” is satisfied. This is regulated by the “Elapsed Time Run Once” express sub-vi, which is shown in Figure 44. This sub-vi is custom written, but mimics the functionality of the “Elapsed Time” express vi built into LabVIEW. The key to its functionality is that the design is open-loop. The present time is compared to the timestamp when the sub-vi was previously run. Shift registers are also key to this design, since they “remember” the value they held during their previous execution. If the time difference between the present timestamp and the timestamp saved when the last file-write was made, then the output is “true”, which signals the data to be appended to file. If the output is “false”, then the data is not saved to file, but the thermocouple readings are still displayed in the indicator. The data points are also plotted on the graph if “Acquire?” is enabled. The graph plotting strategy is outlined next.

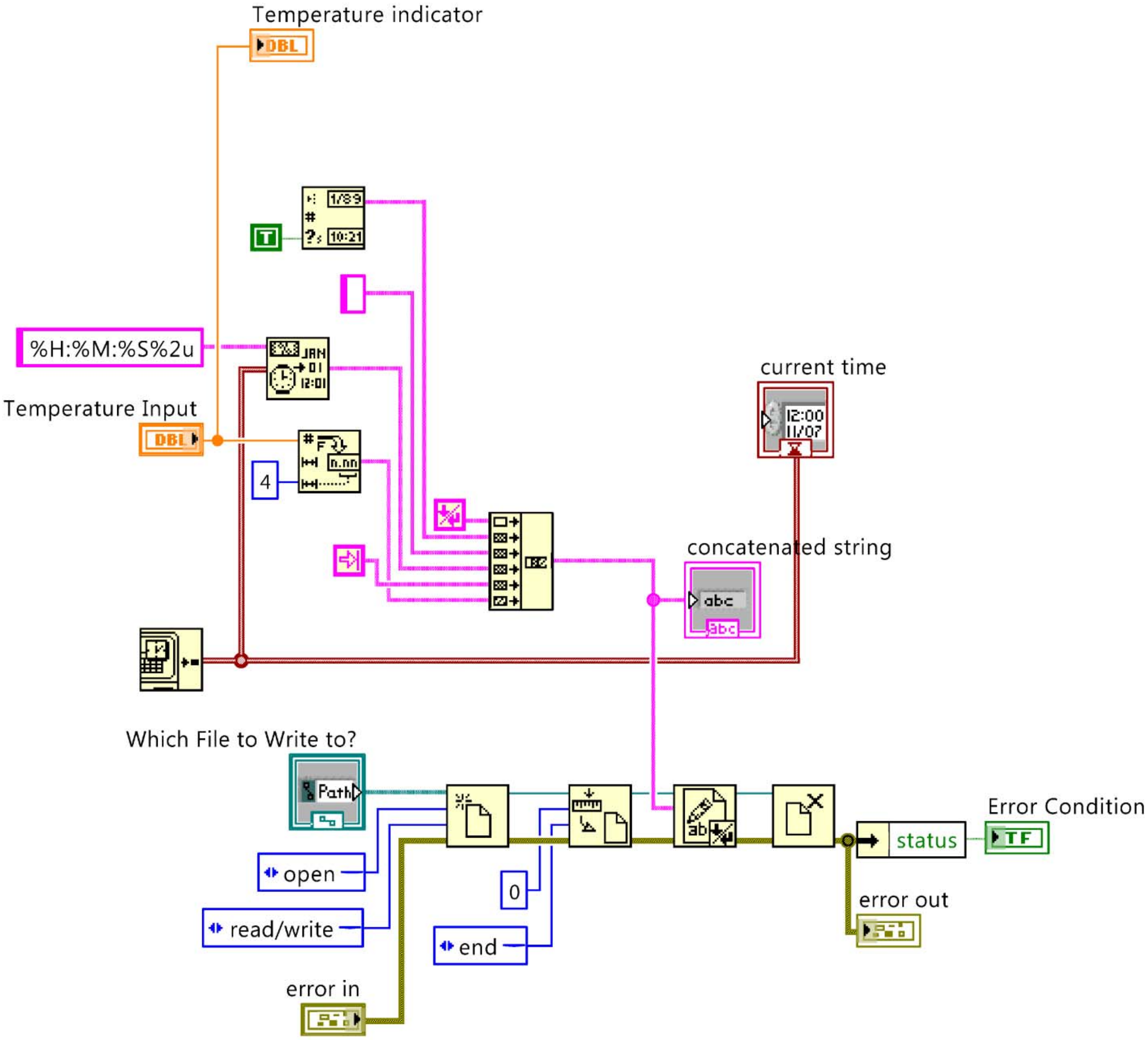


Figure 43 – Append to File sub-vi.

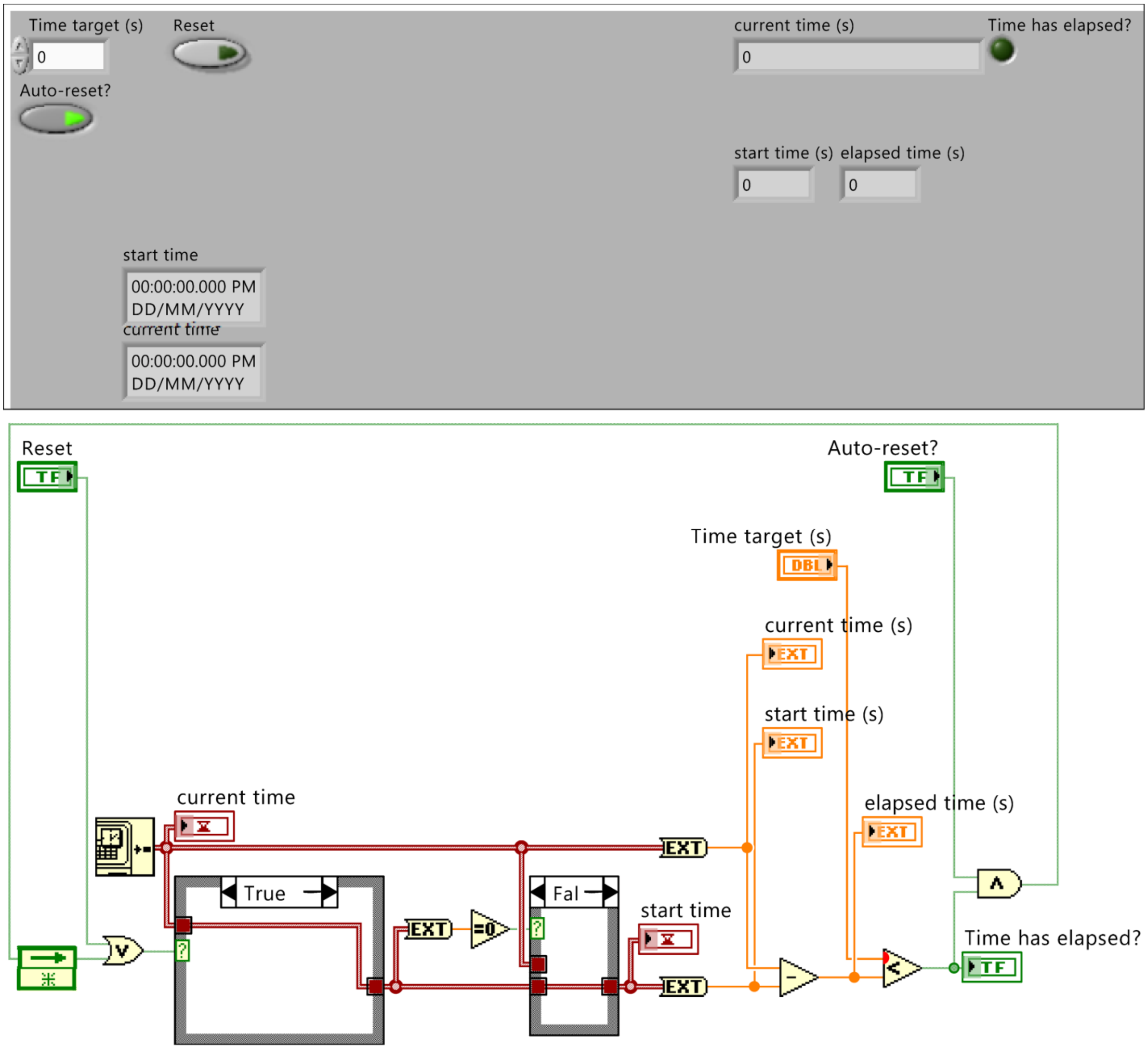
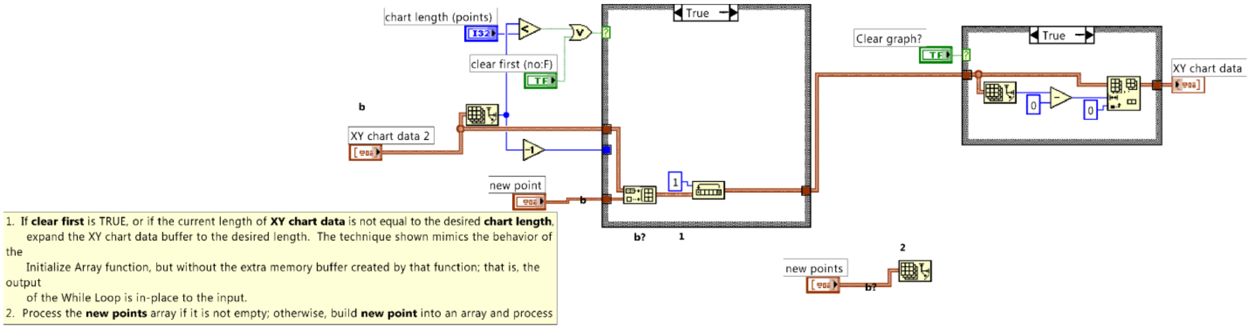


Figure 44 – Elapsed Time Run Once express sub-vi.

The thermocouple data points are generated at irregularly-spaced time intervals, requiring the use of an XY-graph. The input to this graph is very important. Each time a temperature data point is taken, it is combined with the corresponding time-stamp to create a data point. For each channel, each data point is placed in a buffer, which essentially combines the data points into an array. The buffer sub-vi is shown in Figure 45. The buffer outputs from each channel are then combined and used as input for the graph. If "Acquire?" is enabled for a channel, new points can be added to the buffer, else the buffer remains the same. If the "Clear graph?" button is enabled, then any channel with "Acquire?" enabled will have its buffer cleared. The buffer design is not optimal. The buffer starts with one data point and then grows until a maximum value (set at 5000 data points) is reached. Once the buffer is full, the oldest data point is removed and the newest data point is appended to keep the buffer size constant. The problem with this design is that it requires LabVIEW to repeatedly request larger and larger memory blocks, which requires a lot of overhead and can crash the computer. This problem is minimized by keeping the buffer at a modest size. A better design would be to allocate the entire buffer into memory during the initial program execution. This approach is used in all the vi's in this Appendix.



1. If **clear first** is TRUE, or if the current length of **XY chart data** is not equal to the desired **chart length**,
    expand the XY chart data buffer to the desired length. The technique shown mimics the behavior of the
    Initialize Array function, but without the extra memory buffer created by that function; that is, the output
    of the While Loop is in-place to the input.
2. Process the **new points** array if it is not empty; otherwise, build **new point** into an array and process

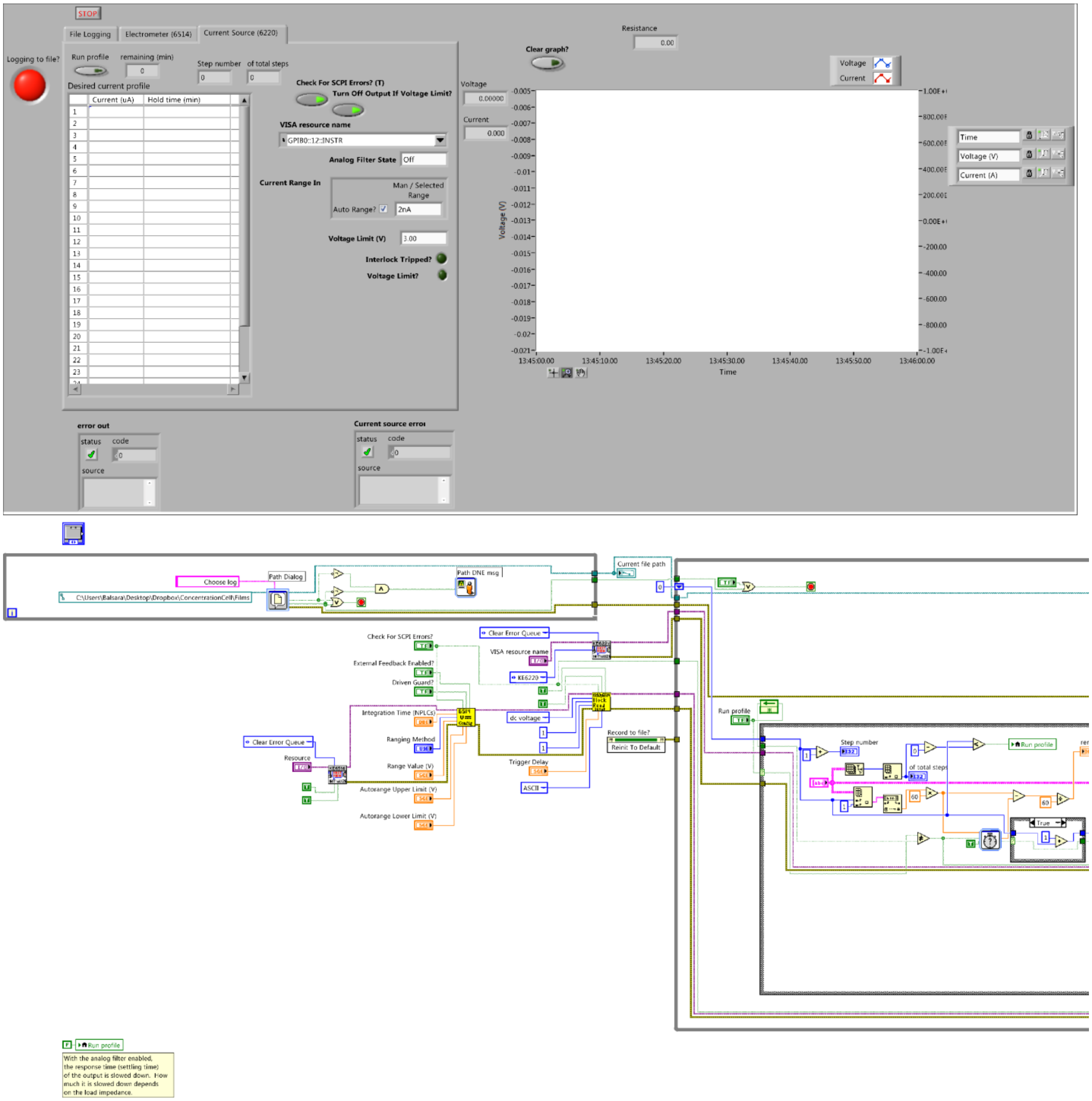
Figure 45 – Chart buffer sub-vi.

Keithley Electrometer and Current Control Program

Two different Keithley programs were used in this Thesis work, both of which serve the same functionality. In both cases, the vi interfaces with two different Keithley instruments. In both cases, one of those instruments is the Keithley 6220 Current Source. In the vi demonstrated here, the second instrument is a Keithley 6514 Electrometer. In the other vi, the second instrument is a Keithley 2182A NanoVoltmeter. The burden of choosing the appropriate program or instrument to use rests with the user, although the recommended course of action is to consult the Keithley manuals, Keithley tech support, and other users who can help determine the appropriate constraints, controls and requirements for the experiment at hand.

The vi based on the 6514 Electrometer is shown in Figure 46 and is continued on Figure 47. The file logging and data graphing strategy is the same as that used for the Thermocouple Reader program in the previous section, although the individual sub-vi's differ slightly in their names and implementation. The Keithley instruments are connected together using GPIB interconnects. The LabVIEW computer connects via USB to a Keithley KUSB-488B USB to GPIB converter, which communicates to the GPIB ports of the instruments. The portions of the program that communicate with the Keithley instruments are taken directly from Keithley LabVIEW drivers, which at the time of this writing were available from both National Instruments and Keithley. When the program is executed, it starts to read from the electrometer. The voltage reading is shown in the "Voltage" indicator, and it is also plotted on the graph. The current is also "read" at each time point. However, the program was designed so that the 6220 current source was not necessarily needed. That is, the program assumes the instrument is "off", and then reports a value of "0" for the current. This is true in most cases, although it could be a "bug" under other circumstances. The program is also designed to apply "current-profiles" to the sample, the input of which is available on the "Current Source (6220)" tab. The user can input a set of current/hold-time pairings, which are then executed in series when the "Run profile" button is pressed. The "Elapsed Time Run Once" sub-vi, as described previously, is used to regulate the period of time that a given current level is applied. After all steps are completed, the current source is turned off. At all points during the profile the current reading is taken from the 6220 unit and subsequently graphed, and, if appropriate, it is logged to file.

The electrometer controls must be set before the program is executed and cannot be changed in real time. Not all of the controls worked, for reasons I was unable to determine, and the Keithley tech support did not help me resolve those issues. Occassionally, a GPIB connection error will arise. This usually happens when talking to the electrometer, although that statement is likely biased by the fact that the electrometer is almost always in use for these experiments, whereas the current source is not necessarily so. The GPIB error can occur within minutes, or may take up to 24 hours. It invariably crashes LabVIEW. I was not able to resolve this error, and its presence will prevent the long-term usage of this program. For the purposes of monitoring concentration cells, however, it was adequate. Similarly, not all of the inputs for the current source are actually applied. It is up to the user to ensure that the program is performing as intended. It would be fruitful to check for updated Keithley-LabVIEW drivers, since they are updated and modernized over time. Keithley tech support may also provide assistance.



VISA resource name: GPIB0::12::INSTR

Analog Filter State: Off

Current Range In: Auto Range? ☑ Man / Selected Range: 2nA

Voltage Limit (V): 3.00

Voltage: 0.00000

Current: 0.000

Resistance: 0.00

C:\Users\Balsara\Desktop\Dropbox\ConcentrationCell\Films

With the analog filter enabled, the response time (settling time) of the output is slowed down. How much it is slowed down depends on the load impedance.

Figure 46 – Keithley 6514 Electrometer and 6220 Current Source vi.

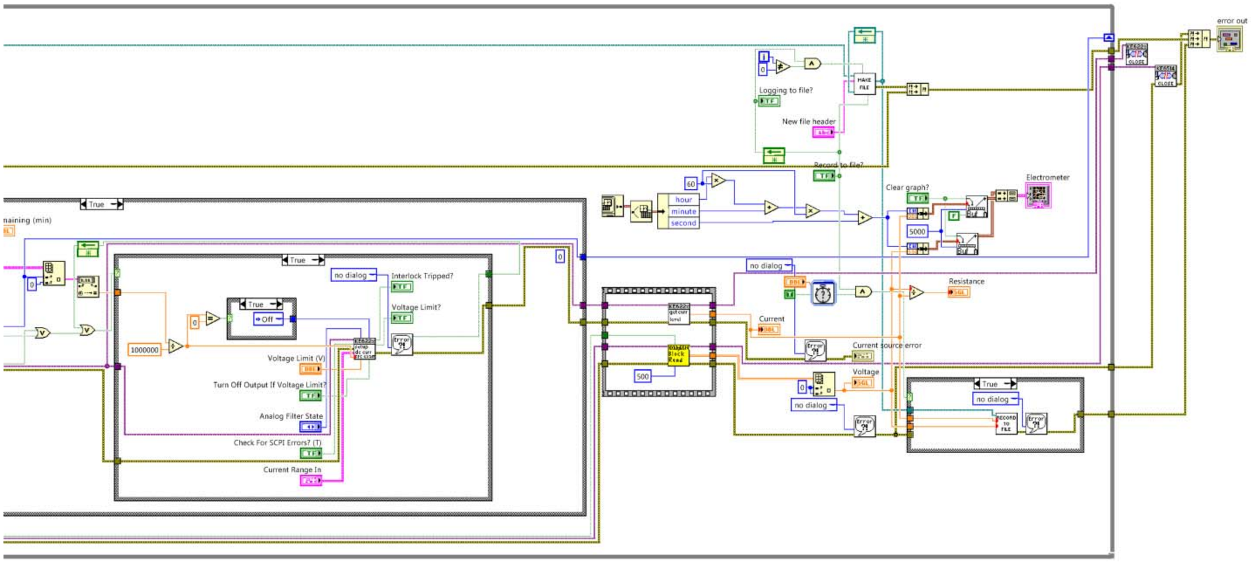


Figure 47 – Keithley 6514 Electrometer and 6220 Current Source vi, part 2.

Temperature Control and VMP3 Triggering Program

The Bio-Logic VMP3 potentiostat was regularly used to acquire conductivity measurements for the work in this thesis. The VMP3 does not have any advanced temperature control functionality. Although it can be programmed to send analog temperature setpoint signals to custom-designed temperature controllers, there is no temperature feedback into the VMP3 control program. The problem with that design is that there is no way to ensure that a stable temperature is achieved, and problems such as missed or incorrect setpoints go undetected. To address this problem, the following program was aimed with the specific goal of interfacing the VMP3 with LabVIEW temperature control, as shown in Figure 48 and is continued in Figure 49. This setup involves two computers: one to control the VMP3 using Bio-Logic's EC-Lab software, and one to run LabVIEW. The VMP3 must be run with a specific program design that signals LabVIEW. Signaling between the VMP3 and LabVIEW is achieved with an intermediate signaling device, a National Instruments USB-6210. The VMP3 auxiliary ports are hard-wired to the signaler, and communication is achieved through standard transistor-transistor-logic (TTL) signaling in which a low-voltage reading (zero to 0.8 V) is considered "off", and a high-voltage reading (2 to 5 V) is considered "on". At present, it is necessarily to send and receive signals from all of the VMP3 channels of interest simultaneously. To accomplish this, the conductors from up to 8 VMP3 channels that send/receive TTL signals are short-circuited together, and all of them are connected to one channel of the USB-6210 signaler. The signaler can accommodate up to 4 separate channels. Temperature control is performed with ethernet-connected Watlow EZ-Zone PM controllers. The LabVIEW program interfaces with the Watlow program to perform temperature control steps. At the end of each step, LabVIEW signals the VMP3 (via the signaler), and then waits for a return signal. Initially, the VMP3 is waiting for a signal. Once the signal is received, it performs the proscribed measurement(s) and then sends a return signal. The LabVIEW program, upon receiving the signal, then advances to the next temperature step and the process repeats.

It is important to disconnect the shorted auxiliary cables from "bad" VMP3 channels. Channels with errors or channels with no experiment running are considered "bad". The problem is that the default state for the auxiliary channels is "off", which is apparently an internal short-circuit. The system has some resilience to faults. For instance, if 8 channels are running and 3 have problems, the system will continue. However, 4 faults will stop all 8 channels. The issue is that the short-circuits on the "off" channels interfere with the signaling process.

The VMP3 program must follow this design, and all channels for the experiment must be grouped together:

Wait for trigger with falling-edge (at the end of the signal, all connected channels advance simultaneously)

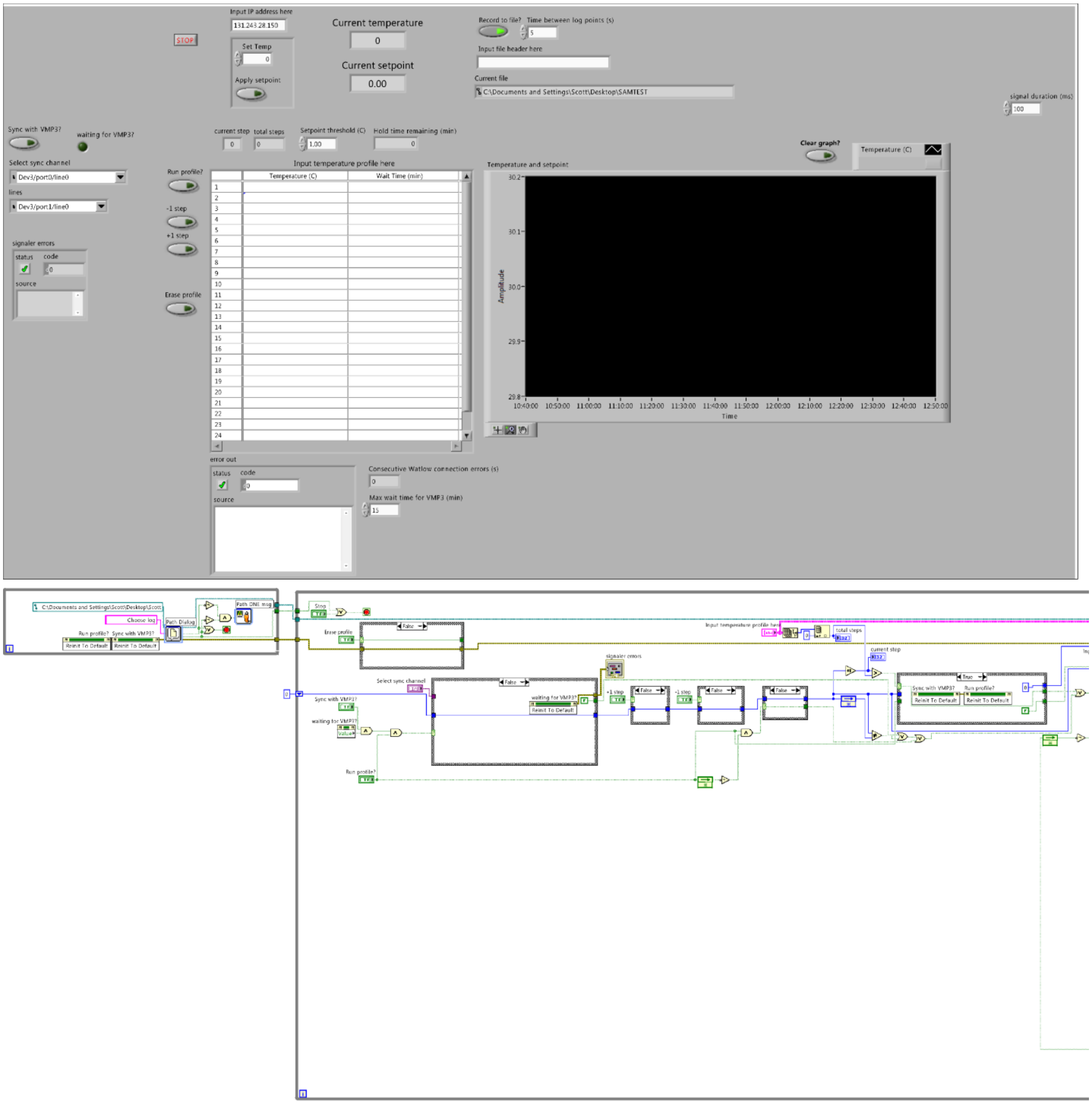
Perform measurements. Since the channels are grouped, each channel will perform the same experiment. For example, run 3 impedance measurements and perform a voltage measurement for some period of time. The time period should not matter as long as it is not so short as to immediately signal back to LabVIEW.

Short Wait period (ie: 0.01 s). This is necessary to reset the VMP3 signal level and is considered a bug in the VMP3 software. Bio-Logic is aware of this problem but it is low-priority for them. Trigger out with rising edge. The duration should be long (ie: 5 s). Since all of the channels are shorted together, it is necessary for them to signal at the same time so that the trigger voltage is large enough for the signaler to detect an "on". The extended signaling time is to compensate for small differences in the time it takes each channel to perform measurements.
A wait period that is considerably longer than the trigger out in step 4. For example, 11 s. Otherwise the VMP3 will immediately advance to the next step. Since the next step is to wait for a falling-edge trigger, if one or more of the channels take slightly longer than the others, that condition can keep the signal high and then at its conclusion it will trigger the other, finished channels that have returned to step 1 already. Once this occurs, a runaway condition can happen in which all of the channels continue to take measurements at random times and get out of sync.
Loop back to step 1. This step can be repeated indefinitely since the temperature control is happening in LabVIEW. The LabVIEW program can also have an arbitrarily large number of temperature steps.

The LabVIEW program in Figure 48 requires a few important inputs. The Watlow internet-protocol (IP) address must be set. The Watlow IP address can be permanently set if a static IP address is programmed and is reserved with the internet technologies (IT) department. Alternatively, the IP address can be assigned dynamically and the user can manually determine the IP address from the Watlow front panel. See the Watlow manual for details. When the program first starts, it asks the user for a directory in which to log the temperature history (using the same strategy as the "Thermocouple Reader" vi. If the IP address is correct (or, at least, it is an IP address corresponding to a compatible device), the temperature will immediately read out on the "Current temperature" indicator and the current setpoint, as read from the controller, is output into the "Current setpoint" indicator. If there is a connectivity problem, such as the controller being turned off, being disconnected from the internet, having an incorrect IP address, or having network problems, the program will continue to run, but no control or logging will occur and the "Consecutive Watlow connection errors (s)" indicator will periodically update. The program has a lengthy communication time-out built in, so this can be an extended period of time between updates. To run a temperature profile, input a list of temperature setpoints and wait times in the "Input temperature profile here" list and adjust the "Setpoint threshold (C)" to the desired value. If it is desired to sync with the VMP3, choose the appropriate channel for communicating with the signaler. It is possible to have up to 4 channels, but typically channel 0 is the only one in use. It is necessary to choose the same channel on both the "Select sync channel" and the control beneath it, "lines" to send/receive on the same channel. Next, enable "Run profile?" and the profile will start.

LabVIEW will apply the setpoint given in the first table entry. The process temperature is monitored continuously. If the temperature reading is within +/- the value of "Setpoint threshold", the countdown timer will begin, else it is reset (including if it drifts or overshoots out of range). Once the process has been within the proscribed temperature zone for the period in "wait time", the program will advance to the next step. If "Sync with VMP3" is enabled, the program first sends a signal out to the VMP3 and waits for the return signal before continuing. During the temperature setting portion, it is common to have network errors that stall the Watlow communications. The program ignores these errors at first, but if they persist for more than some period of time (coded as 60 s in Figure 49), the wait time is reset. This period can be bumped up if there are a lot of network errors. The controller does not spontaneously go out of control under normal circumstances.

The left side of the block diagram in Figure 48 contains multiple conditions under which the wait time should be reset during temperature profile execution. This program was also designed with the “open-loop” strategy, such that all of the sub-vi’s are executed once on each outer-loop execution. The plan was to modularize the entire program and extend it to multiple Watlow controllers. For example, 4 controllers could control 4 temperature stages and talk to 4 different channels on the signaler. That goal was not achieved, although it could be with relatively minimal modification of the code. Essentially the entire inside portion can be turned into a sub-vi. The problem with that approach is that it would have more than the allowed number of inputs. In practice, that approach is unlikely to be necessary, but the “open-loop” design simplifies the code in many ways. For instance, the entire temperature profile execution portion is performed within a sub-vi, as shown in Figure 50. This sub-vi includes three temperature-control-related steps: a temperature read, a setpoint read, and an optional setpoint write. The Watlow communicates through the ethernet ModBUS protocol and a native LabVIEW library is used for that step. The controller essentially has a large number of registers, some of which have read access, some of which can be write access, and some which have both. The “read setpoint” and “read temperature” sub-vi’s simply read values out of the appropriate register, and the “write setpoint” sub-vi simply writes a new setpoint into the appropriate register. Otherwise the native behavior of the control is preserved. These actions are equivalent to reading the front panel of the controller and adjusting the setpoint manually.


STOP

Input IP address here: 131.243.28.150

Set Temp: 0

Apply setpoint

Current temperature: 0

Current setpoint: 0.00

Record to file?

Time between log points (s): 5

Input file header here:

Current file: C:\Documents and Settings\Scott\Desktop\SAMTEST

signal duration (ms): 100

Sync with VMP3?

waiting for VMP3?

Select sync channel: Dev3/port0/line0

lines: Dev3/port1/line0

signaler errors: status ✓ code 0 source

Run profile?

-1 step

+1 step

Erase profile

current step: 0

total steps: 0

Setpoint threshold (C): 1.00

Hold time remaining (min): 0

Input temperature profile here

	Temperature (C)	Wait Time (min)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

error out: status ✓ code 0 source

Consecutive Watlow connection errors (s): 0

Max wait time for VMP3 (min): 15

Clear graph?

Figure 48 – Watlow temperature control and VMP3 signaling vi.

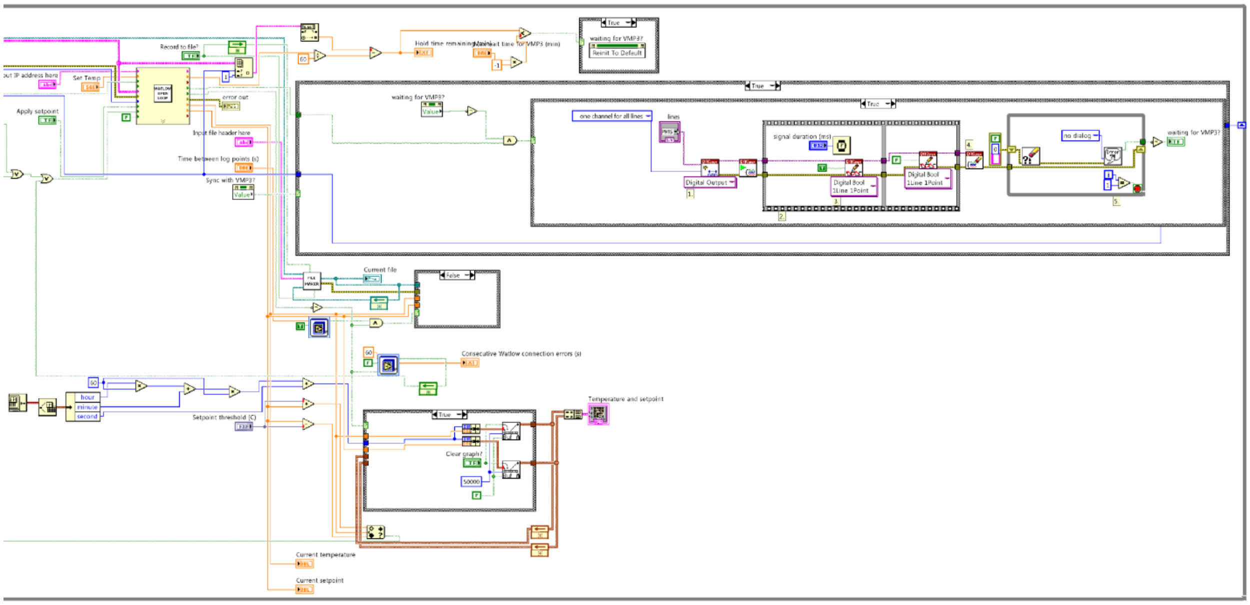


Figure 49 – Watlow temperature control and VMP3 signaling vi, part 2.

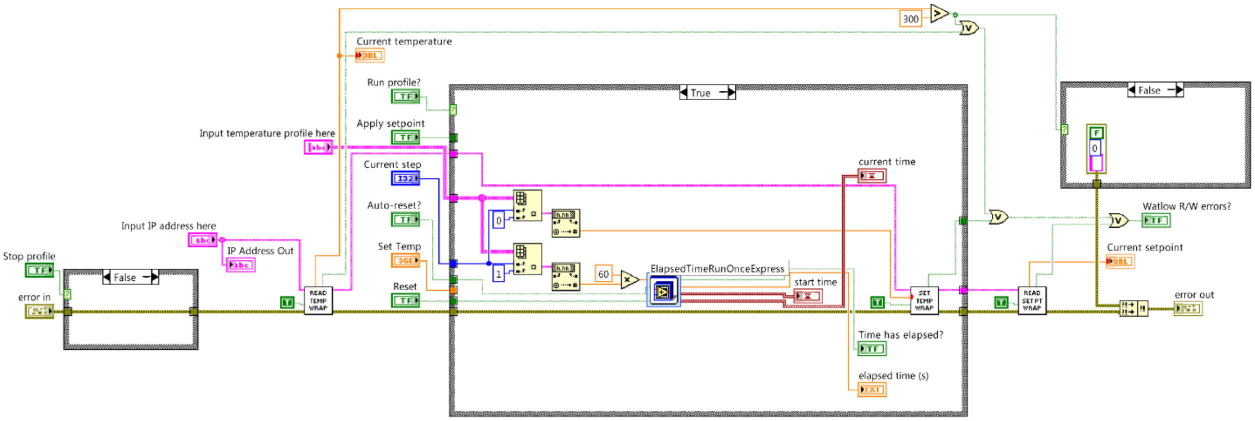
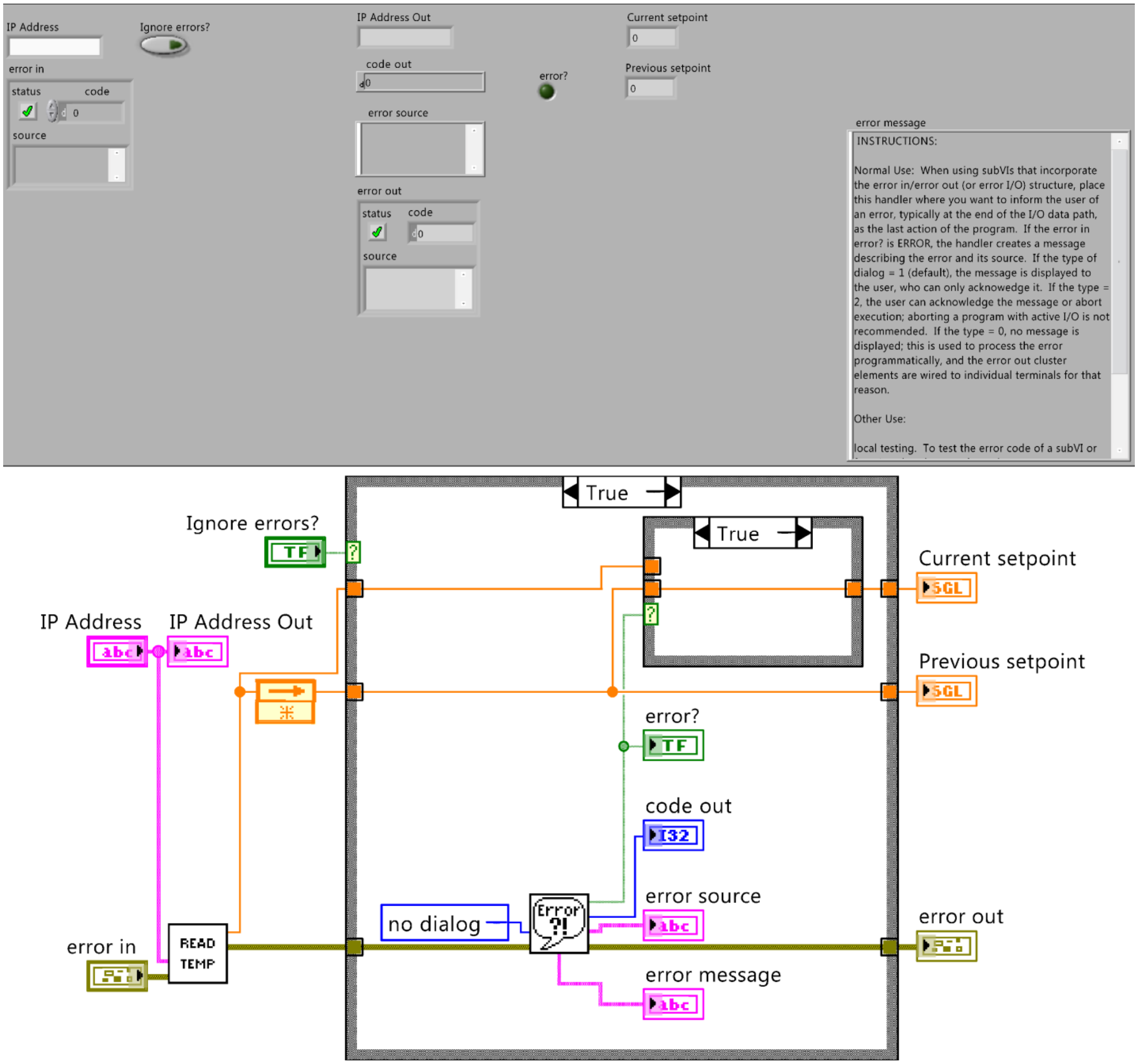


Figure 50 – Watlow temperature profile sub-vi.

The temperature read and control actions are individually confined within “wrappers”. That was my approach to handle errors that are generated by each sub-vi, although a better approach would be to combine all temperature-related controls into one or a few sub-vi’s. The example in Figure 51 performs a temperature read by executing the sub-vi shown in Figure 52. The purpose of the “wrapper” is to detect whether an error occurs, such as those that arise due to connectivity issues. In general, the errors are “ignored”, but noted. That is, if an error occurs the program execution continues, but the fact that an error occured is transmitted out to the outer vi’s. This approach allows temporary connection issues to occur without halting the program, but the outer control can log the duration of the errors and make intelligent choices about how to respond.



error message

INSTRUCTIONS:

Normal Use: When using subVIs that incorporate the error in/error out (or error I/O) structure, place this handler where you want to inform the user of an error, typically at the end of the I/O data path, as the last action of the program. If the error in error? is ERROR, the handler creates a message describing the error and its source. If the type of dialog = 1 (default), the message is displayed to the user, who can only acknowedge it. If the type = 2, the user can acknowledge the message or abort execution; aborting a program with active I/O is not recommended. If the type = 0, no message is displayed; this is used to process the error programmatically, and the error out cluster elements are wired to individual terminals for that reason.

Other Use:

local testing. To test the error code of a subVI or

Figure 51 – Read temperature wrapper sub-vi.

The read temperature sub-vi is shown in Figure 52. This sub-vi interfaces with the LabVIEW modbus-over-ethernet library to read the register(s) that correspond to the current temperature reading from the controller. Similarly, Figure 53 shows the set-temperature sub-vi. The important functionality in the set-temperature sub-vi is that it writes to a register in the controller.

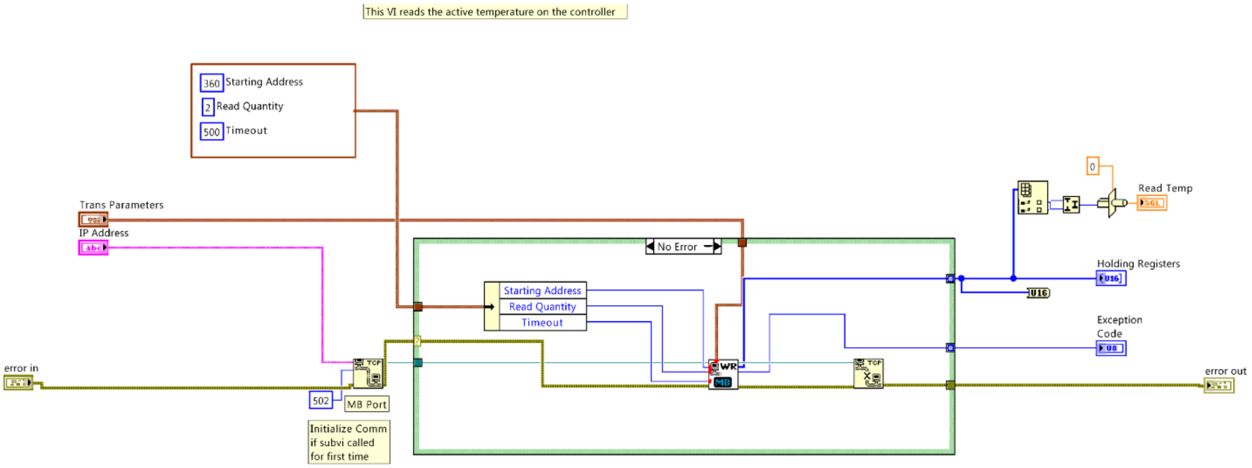


Figure 52 – Watlow read temperature sub-vi.

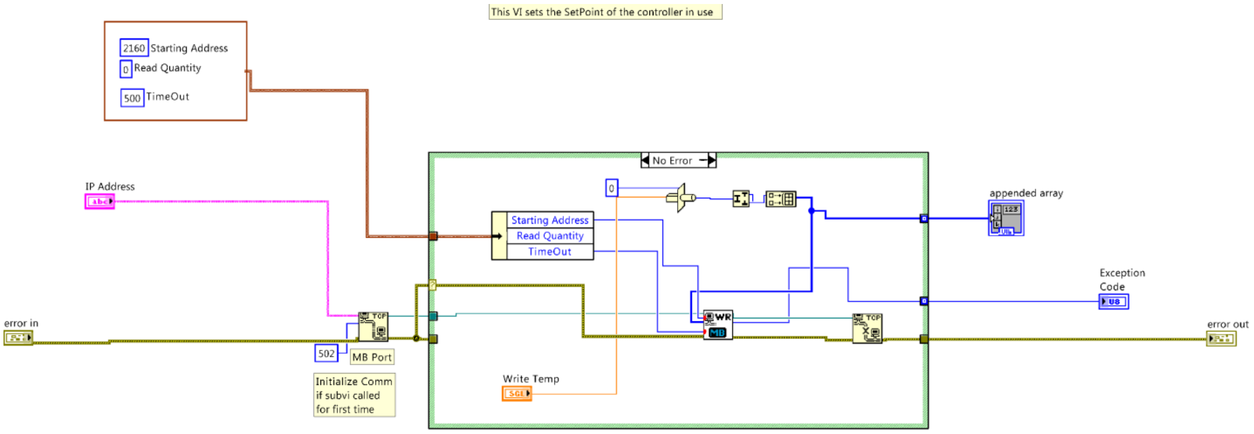


Figure 53 – Watlow set temperature sub-vi.

This program was used to obtain the conductivity/temperature profiles in Chapter 3. The program is fairly robust, but signaling errors are problematic at times. For instance, if there is a connection issue between LabVIEW and the signaler at the time the VMP3 signals back, then there is no way to know whether the VMP3 missed the signal from the signaler or if the signaler missed the signal from the VMP3. In the present design, if LabVIEW is waiting for a return signal for a longer time than the control setting in "Wait for VMP3", then it will assume there was a communication problem and it will send a signal back to the VMP3. Also, the VMP3 channels occassionally trigger themselves. This usually happens about once every 12 to 24 hours, and it may be due to interference acting upon the conductors in the VMP3/signaler connection. This hypothesized interference could arise either from the other conductors on the

VMP3 or from external fields. It is advisable to remove the unused conductors from the interconnects to minimize this possibility.