

**Development of a Combined Multi-Sensor/Signal Processing
Architecture for Improved In-Situ Quantification of the
Charge Balance of Natural Waters**

by

Amy Violet Mueller

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in the Field of Environmental Chemistry

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

Author
Department of Civil and Environmental Engineering
May 4, 2012

Certified by.....
Harold F. Hemond
William E Leonhard Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by.....
Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

Development of a Combined Multi-Sensor/Signal Processing Architecture for Improved In-Situ Quantification of the Charge Balance of Natural Waters

by
Amy Violet Mueller

Submitted to the Department of Civil and Environmental Engineering
on May 4, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in the Field of Environmental Chemistry

Abstract

This thesis details the design, implementation, and testing of a new electrochemical instrument for the in situ measurement of both major and environmentally relevant minor ions in fresh waters, namely Na^+ , K^+ , Ca^{2+} , Mg^{2+} , NH_4^+ , Cl^- , NO_3^- , and SO_4^{2-} . The instrument is built on a hybrid multi-probe / signal processing architecture and is implemented using commercial sensor hardware (primarily ion-selective electrodes (ISEs)) paired with a novel neural network processor designed to take advantage of *a priori* chemical knowledge about the system. Adaptation of this architecture to in-situ conditions and quantification of relatively minor ions required overcoming a number of challenges, including: (1) lack of a standardized method for unsupervised recording of ISE equilibrium potential, (2) non-availability of commercial electrodes for some ion species, and (3) detection of ion concentrations that fall below the ISE linear response region and/or are confounded by the presence of relatively large quantities of interfering ions. As such, a methodology is proposed and validated for standardization of ISE potential readings, resulting in consistent measurements completed in <6.5 min., improving replicability, and facilitating simultaneous measurement of up to 12 ion channels. The sensor suite is then designed such that each ISE provides information about more than one analyte, and finally, the artificial neural network (ANN) architecture is optimized for use on environmental chemical data by including software constraints implementing known chemical relationships, i.e., the concept of charge balance and the total ion-conductivity relationship. Two experiments are conducted using environmentally-relevant data sets (one semi-synthetic, one created in the lab) to characterize the effectiveness of the proposed ANN architecture. Final results demonstrate over an order of magnitude decrease in relative error (as measured against use of ISEs as stand-alone sensors) without concentration-dependent error bias, including estimates for analytes for which no specific ISE exists (SO_4^{2-} , Mg^{2+} , HCO_3^-). Simultaneous un-biased quantification of all eight ions is achieved with $\sim 20\%$ error on most channels including NO_3^- (concentrations $\leq 100 \mu\text{M}$) and $\sim 50\%$ error for NH_4^+ ($\leq 100 \mu\text{M}$), however it is also demonstrated that errors of $\sim 10\%$ are achievable for N-species ions even at low concentration if slightly higher uncertainties on other channels can be tolerated.

Thesis Supervisor: Harold F. Hemond

Title: William E Leonhard Professor of Civil and Environmental Engineering

Acknowledgments



What we all conclude about science after realizing most of our advisors finished their PhD in fewer years than we took to complete undergrad. *Image updated from boasas.com*

I would first and foremost like to thank my advisor, Harry, for all of his advice, direction, and understanding over the years. It is, I believe, a unique and yet universal experience for each of his students to explore his seemingly boundless expertise, testing the waters with ever more difficult or unstudied questions and concepts, with the hope of one day catching up to his subtle and modest intuition for just about everything - and even five, six, or seven years into this process, it is each time somehow simultaneously maddening and reassuring to realize that he had a handle on the crux of the matter well before you even bothered to put the question to him. I consider myself extremely lucky to have had the honor to work with him on this and a number of other projects over the past 8 years, as I have most certainly become a more effectively inquisitive scientist and productive engineer under his guidance.

Similarly Professor Phil Gschwend has been instrumental in teaching me about how to conceptualize my work, distill it down to its meaningful essence, and present it to people in a way that they can understand and - perhaps most importantly - put into context within the vast space that environmental engineering covers. Phil has also taught me how to be a good audience member, how to ask good questions, and how to think about other people's research critically, and for this I am extremely grateful.

And to the whole of the Parsons Supergroup - especially those individuals who saw me through my first few years on this project - I owe all of you for the consistent source of great suggestions, patient ears when my setup was going wonky, and for providing a safe place to talk about research failures as well as successes. I still don't love public speaking, but being able to talk about my work to this wonderful group of people has bolstered me up to the point where I can buck up and get through it without the knock-kneed butterfly stomach I had 8 years ago at just the thought of giving talks.

I must also thank my parents, who probably thought I was crazy for going back to school *again* but have been nothing but supportive along the way, and my brothers, who have made being back in Boston a whole lot of fun. As a side note, Dad, thank you for

acclimating me to the sound of construction at such a young age - it has been key to the completion of a massive amount of this work and writing during the complete renovation of the room next door over the past 6 months. Next is a long list of friends - some who have known me for almost 20 years at this point and some found recently - who must be credited with helping me keep hold of my sanity in dire straits (and for humoring my rants about tiny details of my research with a straight face and a lot of patience). My sanity has also been secured through the cheerful and consistent guidance of Checka, the most wonderful of yoga instructors, who taught me to keep the big picture in mind and strive always for Focus, Communication, Patience, and Contentment. And finally Matt - thank you for adding adventures into the mix and reminding me that there is more life out there waiting to be experienced and humanity out there waiting to be helped. Olyan boldog vagyok hogy egymast vigyazunk et hogy te vagy *az enyem*.

There are also a number of researchers whose work has been inspirational to me, helped me to formulate this project from a wee inkling of an idea into something with a chance of actually seeing the light of day, and taught me volumes about the hardware, software, applications, and context of this project. Since I first stumbled on their trails, I have been actively chasing down just about anything they published and been the better scientist for it, so I wanted to take this opportunity to acknowledge their contributions (and those of their students) to my work: E. Bakker, E. Pretsch, M. del Valle, S. Alegret, Y. Vlasov, A. Legin, C. DiNatale, A. D'Amico, and W. Morf. Thank you for being shoulders I could stand on.

And finally, though they will probably never know how grateful I am, I want to thank the individuals out there who put in the time to catalog tables of seemingly esoteric chemical data or to painstakingly explain obscure Matlab functions in help forums so that I could stumble across it through desperate internet searches. Random folks: you saved me a whole lot of time, and I really appreciate it. If it weren't for you, who knows how much longer this thesis could have taken...

Contents

1	Introduction	19
1.1	Motivation	19
1.2	Problem statement	20
1.3	Summary of thesis chapters	21
2	Background	23
2.1	In-situ ready sensor technologies	24
2.1.1	Potentiometric sensors	26
2.1.2	Amperometric sensors	28
2.1.3	Optical sensors	29
2.1.4	In-situ sensors used for research	30
2.1.5	Commercially-available sensors	30
2.2	Chemometrics: signal processing for chemical applications	31
2.2.1	Linear (logarithmic) models	31
2.2.2	Principal components analysis	31
2.2.3	Partial least squares regression	32
2.2.4	Non-linear PLS and PCA	32
2.2.5	Time domain extensions	32
2.2.6	Machine learning algorithms - artificial neural networks	32
2.3	Development of sensor arrays: the “electronic tongue”	33
2.3.1	Electronic tongue systems	33
2.3.2	Innovative electronic tongue systems	34
2.3.3	Summary of signal processing algorithms, relative utility	37
2.3.4	Shortcomings of current chemical quantification systems	38
3	Hardware Setup	39
3.1	Sensor selection	39
3.2	ISE-to-PC isolation and filtering circuitry	42
3.3	Data acquisition hardware and software	46
4	Determination of Equilibrium Potential for Ion-Selective Electrodes	51
4.1	Introduction	52
4.2	Materials and Methods	54
4.2.1	Theory	54
4.2.2	Sensitivity Analysis	55
4.2.3	Experimental Setup	56
4.3	Results and Discussion	59

4.3.1	Steady state determination	59
4.3.2	Linearity of electrode response	60
4.3.3	Rate of failure to declare equilibrium as a function of parameterization	62
4.3.4	Optimization against system constraints	62
4.3.5	Quantification of response time	64
4.4	Conclusions	66
4.5	Acknowledgments	69
5	ANN Design	71
5.1	Introduction	71
5.2	ANN parameters: function and values	72
5.3	ANN architecture with chemical constraints	73
5.3.1	Network architecture	75
5.3.2	Assigning weights to implement chemical constraints	76
5.3.3	Weight constraints with logarithmic targets	77
5.4	Conclusions	79
6	Proof-of-concept: 1-Anion Subsystem	81
6.1	Introduction	81
6.2	Experimental (materials and methods)	83
6.2.1	Electrode characterization	83
6.2.2	Water quality data: selection, filtering, pre-processing	83
6.2.3	Simulated electrode responses	85
6.2.4	ANN training and use	88
6.3	Results	90
6.4	Discussion	92
7	Full ionic set for environmental sampling	95
7.1	Introduction	95
7.2	Sample set creation	96
7.2.1	Statistical representation of environmental samples	96
7.2.2	Selection of training samples	99
7.2.3	Creation of ion-mix solutions	99
7.3	Sample measurement	100
7.4	Neural network training set	100
7.5	Neural network architectures tested	104
7.5.1	Internal ANN parameter space	104
7.5.2	External ANN parameter space (ANN architecture)	105
7.6	Results	106
7.6.1	ISE-only concentration prediction	106
7.6.2	ANN suite evaluation	109
7.7	Conclusions	123
8	Conclusions	125
8.1	Project summary and conclusions	125
8.2	Suggestions for future work	126
8.2.1	Design for prolonged and in-situ use	126
8.2.2	ANN optimization	127

8.2.3	The environmental matrix	127
A	Supplementary Materials for Chapter 7	129
A.1	Sample creation and characteristics	129
A.2	ISE calibrations and results	134
A.3	ANN evaluation and extended results	137
B	Matlab code	145
B.1	Equilibrium determination	145
B.2	Environmental PDF and sample creation	150
B.3	Creating ANN training set	164
B.4	ANN with chemical constraints	168

List of Figures

2-1	Ion selective electrode cell [1].	24
2-2	Binding free energy ($\text{kJ}\cdot\text{mol}^{-1}$) for alkali metals with a series of cryptand and 18-crown-6 ionophores, from [2].	26
2-3	CHEMFET cross section [3].	28
3-1	Photographs of lab setup for ISE sampling, including ISE hardware (orange), custom circuitry (yellow), data acquisition (teal), and PC for LabView interface (far left).	43
3-2	Wiring diagram for BNC interfaces from ISEs to isolation hardware.	44
3-3	Two-pole Butterworth filter.	45
3-4	Isolation input / low-pass filter circuit schematic.	47
3-5	Isolation input / low-pass filter PCB. Red traces are on the top layer of the PCB while green traces are on the bottom layer. Black encodes component outlines and text printed on the PCB screenprint layer.	48
3-6	LabView user interface.	48
3-7	LabView schematic for data acquisition.	49
4-1	Typical non-monotonic ISE response signal as seen in this study (left) and as elucidated by Lindner et al. [4] (right).	54
4-2	Range of time-series responses of ion selective electrodes to a single aqueous sample.	55
4-3	Mean percent change in determined steady state concentration for the tighter range of parameterizations relative to the baseline case of $\{0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec}\}$. Interior solid bars show the mean change (black: > 0 ; red: < 0) while exterior transparent bars show the mean absolute value of the change. Note that parameterization difference within this range can result in no more than 0.6% change in declared concentration.	57
4-4	Mean absolute change in determined steady state emf [mV] for a range of parameterizations relative to the baseline case of $\{0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec}\}$. Difference in bar heights indicates that emf declared for a specific time series may vary significantly with parameter choice.	60
4-5	Calibration curves for four ELIT ISEs in their respective salts (3σ error bars). Linear fits with near-Nernstian slopes (Nernstian slope at measurement temperature of 19°C is 57.9) and $R^2 > 0.99$ are found for concentrations down to $1\mu\text{M}$ in all cases (down to $0.25\mu\text{M}$ for K^+).	61

4-6	Effect of parameterization on equilibrium failure rate, λ_{failure} , for a sample period of ~ 6.5 minutes. Bars are subdivided by solution content (A) and probe (B) to demonstrate the range of characteristics affecting the response time of electrodes.	62
4-7	Summary of parameterization ‘goodness’ as judged by simultaneous minimization of RMSE and maximization of equilibrium success rate. Cross-hatching = poor results; * = good results for all probes; M/U = good results for the subset of selectivity matched (M) or un-matched (U) salt/ISE probe pairs.	63
4-8	Mean difference in determined response time [sec] relative to the $\{0.4\text{mV}\cdot\text{min}^{-1}, 30\text{ sec}\}$ baseline over a range of parameterizations; plot on the right shows results for a more constrained parameterization set (referenced to plot on left). Blue tones indicate that parameterization produces shorter response times than baseline while red tones indicate longer response times (note color-bar scale change from left to right). Note total difference of almost 3 minutes across parameterizations shown in plot on left as compared to a difference of less than 1 minute on the right.	64
4-9	Effect of electrode sensitivity and membrane type on mean response time for baseline parameter values.	65
4-10	Response time averaged over 9 electrodes for different salt solutions at a range of concentrations. Data are taken for parameter values $\{0.4\text{ mV}\cdot\text{min}^{-1}, 30\text{ sec}\}$	66
4-11	Response time of independent electrode channels as a function of NaCl or KCl concentration. Data are taken for parameter values $\{0.4\text{ mV}\cdot\text{min}^{-1}, 30\text{ sec}\}$	67
4-12	Response time of independent electrode channels as a function of CaCl ₂ or NH ₄ Cl concentration. Data are taken for parameter values $\{0.4\text{ mV}\cdot\text{min}^{-1}, 30\text{ sec}\}$	68
5-1	Prototypical neuron component of a neural network. <i>Figure courtesy of the MathWorks.</i>	72
5-2	Overview of neural network training. Figure from [5].	73
5-3	Matlab-formatted representation of three neural network architectures: a traditional structure (top), one with a single constraint layer (middle), and one with two constraint layers (bottom). (Note that the middle case is labeled EC for the Electrical Conductivity case but could represent either of the chemical constraints discussed here.) Weights and biases omitted from training in the non-traditional architectures are boxed in red, while nodes where Matlab applies the mapminmax (or inverse) function are highlighted in yellow.	75
6-1	Electrode response to a range of salts. (Top) Response of the ELIT Na ⁺ ISE to four cations. (Bottom) Response of 9 electrodes to different concentrations of NH ₄ Cl solution. Note that log-linear responses exist for most ISE/ion pairs, indicating that use of these electrodes in mixed-salt solutions will produce responses due partially to each of the ionic constituents.	86
6-2	Direct comparison of MA and TX data ‘fingerprints’.	87

6-3	Ionic characteristics of Massachusetts (left) and Texas (right) data selected for simulated data set.	87
6-4	Basic feedforward ANN structure used as the starting point for training. . .	88
6-5	Inclusion of charge balance constraint via addition of a non-trained output layer in the ANN. Hidden and output layers shown on the left refer to those included in the generic feedforward ANN structure shown in Fig. 6-4. . . .	89
6-6	Improvement in prediction of MA concentrations using a MA-trained ANN with both conductivity and charge balance constraints. Results are shown relative to use of ISEs as stand-alone (single analyte) sensors.	91
6-7	Concentration prediction results using a MA+TX trained ANN to process both MA and TX data.	93
7-1	One-dimensional probability distribution functions for representative environmental ions, created using archived USGS data for the five states listed. Density values are plotted at bin mid-points.	97
7-2	Cumulative distribution function for 8-D ion Joint PDF. Independent axis is a sorted bin index, with bins sorted by descending density contribution. . .	98
7-3	Mean response of divalent ISEs as a function of primary analyte concentration.	101
7-4	Mean response of monovalent ISEs as a function of primary analyte concentration.	103
7-5	Estimated EC based on ion makeup of water samples as compared to measured EC (temperature corrected and calibrated). Measurements from VWR meter were highly correlated with but consistently lower than those produced by the Amber meter; it is expected this is related to the built in temperature correction software which was disabled for these experiments but does not always completely disable correctly.	104
7-6	ISE-based predictions of ion concentrations (prediction vs. target) for NH_4^+ , NO_3^- , Na^+ , and Cl^- . Limit of detection (LOD) is plotted as a vertical dotted line.	107
7-7	ISE-based predictions of ion concentrations (prediction vs. target) for K^+ , Ca^{2+} , hardness, and SO_4^{2-} . Limit of detection (LOD) is plotted as a vertical dotted line.	108
7-8	Total NRMSE as a function of ANN architecture. Horizontal axis shows output configuration while colored bars represent options for data used in training (mix data only / mix data plus single-salt standards) and data normalization (none / log).	114
7-9	Relative percent error for optimal ANN predictions as a function of analyte concentration. Results are shown for mix data only.	116
7-10	Scatter plots of nitrogen ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.	117
7-11	Scatter plots of Na^+ and Cl^- ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.	118

7-12	Scatter plots of K^+ and Ca^{2+} ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.	119
7-13	Scatter plots of Mg^{2+} and SO_4^{2-} ion concentrations predicted using the optimal ANN as a function of target concentration. (Note most ISE predictions do not fit on graph at this scale.) One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.	120
7-14	Scatter plots of carbonate system ion concentrations predicted using the optimal ANN as a function of target concentration. Note that there is significant bias in these estimates at increasingly small concentrations; it is expected that improvement in sulfate concentrations (which have the similar magnitude contribution as carbonate concentrations in the charge balance equation) would further reduce the uncertainty in these low-concentration predictions, while the same can be stated for further simultaneous improvement of bicarbonate and chloride predictions. One-to-one line shown in red; regression of estimates against targets (concentration data - note low concentrations do not contribute significantly to this fit) and 95% confidence interval on the linear fit shown in black.	121
7-15	Scatter plot of constraint predictions of optimal ANN as a function of target value. One-to-one line shown in red; regression of estimates against targets (data before log-transformation) and 95% confidence interval on the linear fit shown in black.	122
A-1	Ion concentrations in 75 training samples, plotted against sample number. Recall that 'low' and 'high' nitrogen conditions were imposed on, respectively, samples 26-50 and samples 51-75.	130
A-2	Response of ELIT ISEs to each of five single-salt calibration standards. . .	134
A-3	Response of glass and divalent cation ISEs to each of five single-salt calibration standards.	135
A-4	Response of anion ISEs to each of five single-salt calibration standards. . .	136
A-5	Relative error (as %) for ISE-based predictions of ion concentrations. . . .	141
A-6	Correlation of net goodness parameters with error calculations for nitrogen ions.	142
A-7	Scatter plots of ion concentrations predicted using the optimal ANN (chosen using the MRE metric) as a function of target concentration.	143

List of Tables

2.1	Sensor types used for measurements of relevant ions.	25
3.1	Overview of commercially-available sensors of interest for this application (accurate as of 2009; manufacturers such as WPI and YSI released some additional ISE-based instrumentation in 2010-11). Analytes listed are not comprehensive and are intended to be representative of quantities of interest for this application.	41
3.2	Sensor hardware incorporated into the sensor suite.	42
3.3	Physical layout of inputs to LPF Stage from Coax Plug Wall.	44
3.4	Physical layout of outputs from LPF Stage to Data Acquisition I/O Pins. Note: * value connected to all ‘-’ inputs for used analog input ports.	45
4.1	Single-salt standards used for electrode characterization, producing a total of 52 standard salt solutions.	58
4.2	Ion selective electrode hardware; information on membrane composition and published detection limit (LOD) as given by manufacturers where available.	59
5.1	Neural network characteristics and parameters.	74
5.2	Charge balance and conductivity constraint multipliers used for calculation of non-trained neuron weights. Conductance values adapted from [6, 7]. *Note that the charge balance constraint has been formulated such that the balance of all other ions is trained to the net contribution from H^+ and OH^- as explained further in the text.	78
6.1	Experimental Characterization of ISE limits of detection (LOD). Primary refers to the ‘named’ ion of selectivity (e.g., Ca^{2+} for the ELIT Ca^{2+} ISE), while secondary, tertiary, and quaternary (analyte indicated in parenthesis after the LOD value) are ordered by response magnitude ($mV \cdot M^{-1}$) and not the LOD value. Note that Cl^- was the only cation in this study (excepting OH^- at low, fairly constant concentration) and thus does not have data for response to non-primary ions.	84
6.2	Data from USGS sites in Massachusetts [8].	84
6.3	Data from USGS sites in Texas [8].	85
6.4	Mean change in parameters, given as (simulated value - recorded value) for chloride, conductivity, and hardness (percentage change relative to the mean of measurements is given in parentheses), resulting from the creation of the semi-synthetic data set from actual ionic data measured by the USGS.	85
6.7	NRMSE comparison of MA+TX-trained ANN applied to (a) MA data and (b) TX data.	91

6.5	NRMSE of analyte concentration predictions for MA data made using ISEs only (as stand-alone single-analyte sensors) and by using ISEs processed with the optimal ANN (minimization of NRMSE for these five analytes). Best predictions for each analyte are identified with bold font.	92
6.6	NRMSE comparison of MA-trained ANN results when applied to (a) MA data and (b) TX data. Results are compared to (c) NRMSE for ISEs used as single-analyte sensors on TX data, and degradation in estimation performance is represented by (d) the ratio of NRMSE for TX data to NRMSE for MA data.	92
7.1	Approximate concentration ranges for ions of interest in New England waters ($\log_{10}[\text{M}]$).	98
7.2	Salt solutions used in creation of ion mix samples (all standards at 100mM except for $\text{Ca}(\text{OH})_2$ and MgCO_3 which were 20mM and 1.2mM respectively).	99
7.3	Required (for calculation of chemical constraints) and additional (quantities that can be calculated or inferred given provided information) target outputs for the neural network architecture.	102
7.4	Range of parameters explored for design of neural networks.	105
7.5	Formulae for metrics used to rank ANN results, including MSE (mean squared error), NRMSE (normalized root mean squared error), MRE (mean relative error).	105
7.6	Range of parameters explored for design of neural network architecture.	106
7.7	Errors for ISE-based ion concentration predictions.	109
7.8	Parameterizations for best ANN (chosen using NRMSE metric) as a function of ‘External’ architecture.	110
7.9	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to concentration values of mix data ; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	112
7.10	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to concentration values of mix and single-salt data ; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	112
7.11	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to logarithm-transformed mix data ; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	113

7.12	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (<i>EW</i>). Architectures trained to logarithm-transformed mix and single-salt data ; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	113
7.13	Parameterization of the linear regression of ANN-predicted concentrations against target concentrations; in nearly all cases slope is statistically indistinguishable from 1 and intercept is statistically indistinguishable from 0.	115
7.14	Ion concentration prediction errors for optimal ANN compared to results using ISEs as stand-alone sensors.	115
A.1	Delimitation of bin edges (as $\log_{10}(M)$) for PDF based on USGS-recorded environmental samples.	129
A.2	Ion concentrations (μM , except for alkalinity which is given in mM) for chosen training sample mixtures. Recall that ‘low’ and ‘high’ nitrogen conditions were imposed on, respectively, samples 26-50 and samples 51-75; in some cases, equilibrium pH will lead to decrease in NH_4^+ as equilibrium shifts toward NH_3 . Calculations are given for alkalinity, pH, and carbonate species, for cases of (1) full equilibration with atmospheric CO_2 (392 ppm, recorded at Mauna Loa Dec. 2011) or (2) limited carbon exchange (carbonate system limited by standards used for sample creation).	131
A.3	Most informative calibration curve for prediction of target ions directly using ISEs as stand-alone sensors	137
A.4	Pairwise correlation coefficients between net (whole data set) goodness metrics and those calculated individually for nitrogen ions. Methods are: MSE (mean squared error), NRMSE (normalized root mean squared error), MRE (mean of absolute value of relative error).	137
A.5	Parameterizations for best ANN (chosen using MRE metric) as a function of ‘External’ architecture.	138
A.6	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (<i>EW</i>). Architectures trained to concentration values of mix data ; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	139
A.7	Concentration NRMSE and (<i>MRE</i>) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (<i>EW</i>). Architectures trained to concentration values of mix and single-salt data ; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.	139

A.8 Concentration NRMSE and (*MRE*) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns. 140

A.9 Concentration NRMSE and (*MRE*) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix and single-salt data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns. 140

Chapter 1

Introduction

This thesis details the development of a novel architecture for *in-situ* measurement of the ions that constitute the major charge balance of natural waters. Accurate knowledge of these analyte concentrations can provide critical information needed to (1) diagnose the cause of changes in indicator measurements such as pH or conductivity (traditionally used to monitor for threats to ecosystem health); (2) identify water provenance, in terms of geological or anthropogenic source (e.g., inputs from water treatment plants, agriculture, etc.); and (3) provide a more complete scientific understanding of aquatic ecosystems, including inorganic nutrient cycling and quantification of system alkalinity. Currently, however, measurement of these ions is accomplished primarily by lab-based analysis of physical grab samples which, due to both cost and logistical constraints, limits sample collection to low spatial and temporal resolution. In contrast, an in-situ methodology which can be deployed *in the field* to collect measurements *in real time* would promote sampling at high spatial and temporal resolution, improve the quality and scope of data provided to scientists and environmental managers, and do so while simultaneously reducing the financial burden of data collection. It is these scientific and environmental motivations that drive this work, while the logistical and financial constraints provide a context in which to situate instrument development.

1.1 Motivation

Constituents of the major charge balance of natural waters play a key role in, or are derived through, natural processes including rock weathering, runoff from land surfaces during precipitation events, and nutrient cycling (by bacteria, algae, etc.). Concentrations of ions in surface waters subsequently have direct and indirect effects on the health of macroscale flora and fauna in aquatic ecosystems; phenomena such as eutrophication (increased nutrient loading) and acidification can lead to abrupt changes in food cycles, death of individuals, succession of heartier species, or disruption of natural breeding cycles. Currently anthropogenic influences on natural systems are widespread and exist at many scales, as both point and spatially-distributed sources. For example, the following all contribute to alteration of natural charge balance levels and/or cycling: (1) nitrogen and phosphate runoff from agriculture, (2) direct nutrient inputs via wastewater treatment effluent, (3) altering of water temperature and ionic/dissolved constituents via use as process water at power plants and factories, and (4) acid rain.

Because of the ubiquity of human influences and potential consequences of such alterations to the natural environment, interest in monitoring ion and nutrient levels in the

environment is widespread. Many governmental (US Geologic Survey, US EPA, MA Dept. of Environmental Protection, MA Dept. of Fish and Game, etc.), non-profit (Mystic River Watershed Association, CA Clean Water Team / Surface Water Ambient Monitoring Program (SWAMP)), and academic institutions undertake water quality sampling at hundreds of locations around the country at scales from hourly to annually. Collected data are used to assess drinking water quality, recreational water quality, and the health of natural ecosystems, as well as to support scientific studies of the natural cycling (of nutrients, micro-, and macro-organisms) in both affected and relatively pristine ecosystems. In addition, more targeted sampling, managed by any of these institutions, attempts to trace sources of contamination to ecosystems in which problems have already been identified. Conversely, commercial businesses (e.g., utilities, manufacturing facilities) that use or discharge to public water bodies typically monitor outfalls to catch potentially damaging conditions before the ecosystem is substantially affected; these sampling efforts may be in addition to or cooperative with those implemented by state or federal agencies charged with regulating such water use. In many of these contexts, however, management decisions are based on a very small number of samples that may or may not accurately represent the true status of the ecosystem. In these cases, the capability to take in-situ measurements in real time will provide practitioners with increased spatial and temporal resolution in their data along with the capability to adaptively map out characteristics within the ecosystem while in the field. This will improve ability to pinpoint areas of particular concern, allow more optimal placement of physical sampling stations, and lead to improved confidence and prediction capabilities in environmental assessments. The diversity and importance of all of these areas of work and research speaks strongly to the need for the type of instrument targeted in this doctoral work.

1.2 Problem statement

The proposed purpose of this thesis is thus the development of an **in-situ** sensor array for measurement of the ions making up the major charge balance of natural waters. This project specifically targets the ability to quantify concentrations of these constituents *at environmental levels, in the environmental matrix, and with no sample pre-processing*. This research aims to take advantage of identified areas where other similar technologies have fallen short to improve overall system function and extend utility of the technology to in-situ environmental applications. The proposed instrument is thus envisioned as: a hybridized sensor suite with additional probes for environmental variables (pH, temperature, conductivity), coupled with a machine-learning type non-linear multivariate signal processing scheme (e.g., ANN). Alternative ANN architectures and training methodologies, e.g., use of known chemical constraints, are explored. Ultimately, the goal is the rapid and accurate measurement of concentrations in the field to improve upon current options and to reduce the constraints and burdens associated with traditional sampling campaigns methodologies.

The analyte set of interest, $\{\text{Ca}^{2+}, \text{Mg}^{2+}, \text{K}^+, \text{Na}^+, \text{NH}_4^+, \text{Cl}^-, \text{NO}_3^-, \text{SO}_4^{2-}\}$, has been chosen to facilitate measurements relevant to multiple goals of both environmental engineering and science, including quantification of nutrient levels, computation of alkalinity values, and improved understanding of the nitrogen cycle. Note that, while not explicitly included in the analyte set, the pH and carbonate systems also play an important role in the chemistry of these applications, and they will also be quantified through specific sensors and/or interferences on other channels. This set also includes all major anion

and cation species typically found in freshwater systems, allowing verification of estimated concentrations through application of electroneutrality ($\sum n \cdot \text{anion}^{n-} = \sum n \cdot \text{cation}^{n+}$) and conductivity measurements, discussed in more detail in Chapters 6, 7 and 5.

Successful implementation of an in-situ sensor suite of the type described above is expected to contribute the following to the current state of technology and innovation:

- Improved in-situ measurements compared to current single and multi-probe technologies
- Reduced cost and time required for environmental sampling campaigns
- Use of chemical principles (electroneutrality) and measurements (conductivity) to constrain quantification of analyte concentration and extend utility of current commercial hardware
- Application of extended machine learning techniques, e.g. ANNs with feedback, to environmental problems

1.3 Summary of thesis chapters

A brief summary of the contents of each chapter, along with its relation to the overarching theme, is provided here.

Chap. 2: Provides background on the hardware and software methods relevant to this thesis as well as describing other related work in the field.

Chap. 3: Covers the design and build-out of electronics and physical hardware systems used to obtain accurate, minimally-noisy measurements from the ISE sensor array.

Chap. 4: Details creation and evaluation of an algorithm for determining equilibrium measurements from ISEs whose response time is on the order of minutes and whose voltage is, at least in part, controlled by slow diffusion processes at the sensor membrane. This chapter focuses on measurement *precision*, i.e., repeatability, for ISEs.

Chap. 5: Provides an introduction to neural network techniques and describes the method used to integrate chemical knowledge into the ANN framework.

Chap. 6: Explores application of the proposed ANN architecture to a synthetic data set, created by combining calibration data for a subset of the proposed ISE suite (restricted to a single anion for explicit use of chemical constraints) with historical USGS data for fresh waters in New England. This and the next chapter extend the work of Chapter 4 to improve *accuracy* of ultimate concentration predictions.

Chap. 7: Builds upon the work of the previous chapter by applying the entire proposed suite of ISEs to measurement of a range of environmentally-relevant samples to verify utility of the proposed method for quantification of the full ion set at environmental levels and in environmental mixes.

In addition, several appendices provide supporting data, code, etc.; references to specific materials available in the appendices are given in the individual chapters.

Chapter 2

Background

It was the discovery of the hydrogen electrode (Le Blanc, 1893) and the pH electrode (Cremer, 1906) [9] that spurred the development of electrochemical systems which would improve the speed, accuracy, and ease of chemical measurements relative to those possible with the best mechanical technologies of that era. Development of in-situ electrochemical sensors is today providing similar benefits (improved ease of use, increased speed) relative to current lab-based instrumentation. Work in the 1960's and 1970's introduced the concept of the Ion Selective Electrode (ISE) and the corresponding optical sensors (optodes/optrodes), along with the birth of the ISFET (ion selective field effect transistor), the semi-conductor cousin of the ISE. Continued improvement of the selectivity, stability, and longevity of these devices has brought them into widespread use in medical, biological, and chemical applications; however, the applicability of these sensors for in-situ environmental purposes has to date been limited because of two major hurdles. In-situ environmental sampling requires resolution of relatively low concentrations (down to μM levels for most nutrient species, from 10 μM –10 mM for most other major ions) against a complex background that frequently contains interfering ions at levels that defeat current technology. In addition to ongoing research into more effective selectivity mechanisms, efforts evolving over the past 10–20 years have also focused on two techniques for overcoming these challenges using currently available devices: (1) compilation of sensor arrays to better quantify analytes of interest in these complex solutions, and (2) use of signal processing techniques to untangle, and possibly even take advantage of information available in, interferences. The state of research in these areas will be detailed further in this chapter.

Significantly, implementation of appropriate signal processing methods can allow use of commercially-available sensors well outside of their intended application areas, shortening the development timeline relative to novel hardware development, and providing tools for scientists, managers, and decision makers who are looking for improved ways to harvest data from the environment. These techniques are not analyte specific and, once developed, have the potential to provide solutions to various other problems in environmental chemistry as well.

It is in this context that I present my doctoral work, focused on the combined use of these two techniques to extend utility of current technologies to the application of in-situ environmental measurements. Such a project requires use of current commercial sensor technology as well as novel signal processing algorithm development, both of which will be extensively covered in this thesis. In this section, however, I will briefly present an introduction to several important topic areas: sensor technologies applicable to in-situ problems

(2.1), signal processing methodologies appropriate for this application (2.2), and the current state of sensor array technologies and research (2.3). For those wishing to delve into any of these topics in more detail, I refer you to the many available reviews and primers on these topics that have been published in the last 10-20 years:

- History and development of compact ion sensors (ISEs/optodes): [9, 10, 11]
- Ion selective electrodes: [1, 12, 13, 14, 11, 15, 16, 17, 18, 3, 19]
- Electrochemical sensors (more broadly): [20, 21, 22]
- IUPAC official recommendations and reviews: [23, 24, 25]
- Chemometrics (including neural networks): [26, 27, 28, 29, 30, 31]
- Electronic tongues, sensor arrays: [32, 33, 34, 35], and from 2010 alone: [36, 37, 38, 39]
- Sensors for environmental applications: [40, 41, 42, 43, 17]
- Context and requirements for environmental sensor systems: [44, 45]

2.1 In-situ ready sensor technologies

The ability to perform chemical measurements in-situ decreases the time required for field campaigns, reduces sampling cost, eliminates error due to sample contamination and potentially increases measurement quality and resolution in both space and time. As such, in-situ use is the primary motivation for the development of the sensor proposed here. Following is a review of relevant sensor technologies that can be used in-situ, along with an exploration of commercially available units, both of which are key to the design of the desired instrumentation architecture. An overview of the types of sensors detailed here is provided in Table 2.1. Note that, while this thesis focuses on oxic applications, the eventual extension to anoxic environments will require acquisition of information about the redox state of the system, e.g., ORP, pO_2 , pCO_2 , or dissolved iron concentrations; as such, sensors for several gaseous species are included but will not be discussed extensively in this document.

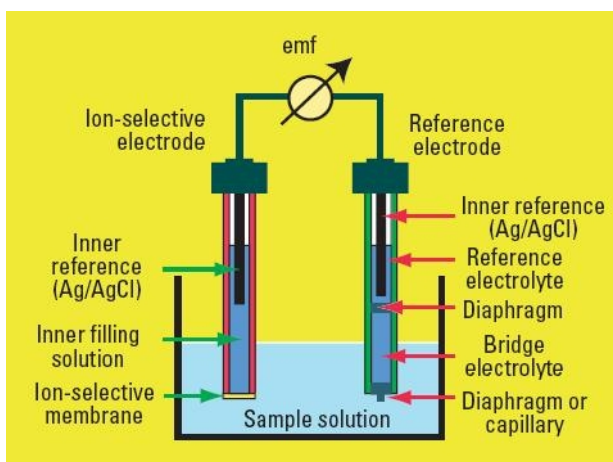


Figure 2-1: Ion selective electrode cell [1].

Table 2.1: Sensor types used for measurements of relevant ions.

Sensor Type	Implementation	Analytes Measured	Ref.
<i>Potentiometric</i>	Ion-Selective Electrode (ISE)	pH, inorganic ions, metals, organic ions, gasses (NO, pCO ₂), etc.	[46]
	ISFET	H ⁺ , OH ⁻	
	CHEMFET	similar to range for ISEs	[3, 47]
<i>Amperometric</i>	Clark-type electrode	N ₂ O, O ₂ , H ₂ S	[48, 43]
<i>Voltammetric</i>	LSV (linear sweep voltammetry)	sulfur species	[49]
	LSV, SWV (square wave voltammetry), etc.	redox-active species (e.g., O ₂ , Fe ²⁺ , Mn ²⁺ , H ₂ S)	[50]
	(varied)	O ₂ , Zn, Cd, Pb, Cu	[43] (review)
<i>Polarographic</i>	DPP (differential pulse polarography)	sulfur species	[49]
<i>Optical</i>	ISUS (in-situ UV spectrophotometry)	NO ₃ ⁻ , NO ₂ ⁻ , Br ⁻ , HS ⁻	[51, 52]
	Optode (fluorescence)	NO ₃ ⁻ / NO ₂ ⁻ , Cl ⁻ , CO ₂ (aq), F ⁻ , H ₂ PO ₄ ⁻	[53, 54, 55, 56]
	Optode (fluorescence - based detection)	lifetime O ₂	[57]
	Optode (absorbance)	K ⁺	[58, 59]
	Optode (general)	similar to range for ISEs, including Cu ²⁺ , Co ³⁺ , Pb ²⁺ , Ni ²⁺ , Fe ³⁺ , Ca ²⁺ , K ⁺ , O ₂	[60, 10, 61]

2.1.1 Potentiometric sensors

The most abundant type of portable, low-power electrochemical sensors is potentiometric. These sensors are typically based on measurement of the voltage potential of an ion-selective electrode (constructed using an ion-selective membrane) relative to a reference electrode placed in the same solution (see Fig. 2-1). In the absence of interfering ions, the response potential E of an ISE to the analyte of interest generally follows the Nernst Equation:

$$E = E^{\circ} - \frac{RT}{zF} \ln \frac{a_{red}}{a_{ox}} \quad (2.1)$$

where E° is the standard electrode potential, z is the charge of species, T is the temperature in degrees Kelvin, R is the universal gas constant, and F is Faraday's constant. At standard temperature and pressure, this corresponds to approximately $59.1mV$ change per order of magnitude change in concentration (when $z = \pm 1$). The inclusion of interfering ions severely complicates the mathematics, however, which will be discussed further in Section 2.2.

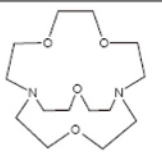
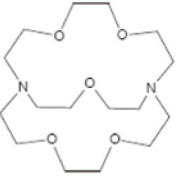
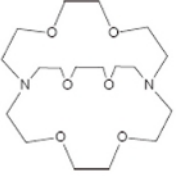
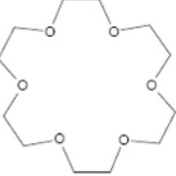
Ionophore	Family	$-\Delta G^{\circ} \text{ (kJ mol}^{-1}\text{)}$				
		Li ⁺	Na ⁺	K ⁺	Rb ⁺	Cs ⁺
	Cryptand	69.5	-	-	-	-
	Cryptand	41.8	74.1	64.0	53.1	-
	Cryptand	-	60.2	75.3	70.3	43.1
	18-crown-6	-	25.1	34.7	30.6	27.2

Figure 2-2: Binding free energy ($\text{kJ}\cdot\text{mol}^{-1}$) for alkali metals with a series of cryptand and 18-crown-6 ionophores, from [2].

Ion-selective membranes may be chalcogenide or oxide glass, crystalline materials, or a polymeric membrane (e.g. PVC), doped with a plasticizer and an ionophore (supplying the specificity of the membrane). The popularity of polymeric membranes has grown with the development of ionophores of increased specificity, however it must be kept in mind that the binding free energy of many ionophores is similar across analytes with similar charge and volume (see Fig. 2-2) and thus there exists an inherent limit for improvements in ionophore specificity.

Fig. 2-1 shows an electrode with inner reference electrolytes. More recently, all-solid-state ISEs have also been developed that couple the membrane directly to the electric circuit or provide an internal ‘gel’ type reference to remove the need for the internal reference solution. Initial issues with stability have been overcome, making solid-state PVC ISEs a viable option for environmental work at this time. Main benefits of ISEs are low cost, ease of use, and wide availability. Issues that continue to confront ISEs are selectivity, sensor drift, need for lower detection limits, and lifetime/re-conditioning limitations. Significantly, detection limits could be lowered if one were able to take advantage of information contained in the non-linear portion of the response curve, and different proposals for how to do so have been put forth, e.g., [62]. It is also important to note that active research in the area of ISEs has identified cross-membrane voltage gradients as the major roadblock to significantly lower detection limits, and methodologies for eliminating these gradients have proven successful at the research level [63, 1, 14, 13, 64] but have not yet been commercialized. Because ion selective electrodes have been researched for many years at this point, extensive material covering their function and time response is available (see [4, 65] for two key classic sources) as well as discussion of the mathematics [66, 67, 68] and general construction/function [69, 70, 12, 1, 18].

The semiconductor cousins of the ISE, the ISFET and CHEMFET, are potentiometric sensors also widely used in medical, biological, and environmental applications. The ISFET, developed in the mid-1960s, is generally defined as a field-effect transistor (FET) for which the gate-source potential is pH dependent (i.e., depends on $[H^+]$ or $[OH^-]$). By extension, a CHEMFET is a FET constructed to be sensitive to an analyte other than H^+ or OH^- . This is achieved by layering an ion-selective membrane (often identical to those developed for ISEs) onto the gate channel of an ISFET device [71]; Fig. 2-3 shows a cutaway of this architecture. The membrane acts as an analyte-to-pH transduction unit, introducing H^+ / OH^- molecules to the ISFET gate in proportion to the number of bound molecules of interest on the external side of the membrane. Detection of analyte concentration is then done by extension of the ISFET principles. More discussion of ISFET/CHEMFET functionality can be found in [3, 19].

Finally, a novel extension of the membranes used in traditional ISEs has recently been described wherein they are combined with voltammetric electrodes to produce ‘V-ISEs’ (voltammetric ion-selective electrodes). Because voltammetric response is controlled by the ions reaching the working electrode and ion-selective membranes theoretically control this flux, researchers were able to measure a number of inorganic cations via voltammetric techniques using working electrodes coated with ion-selective membranes [72]. This technique has not yet been explored for in-situ use.

The key characteristic to recognize with respect to ion-selective electrodes is, for the analytes targeted in this thesis, that perfect selectivity is made nearly impossible by the similar sizes and charges of dissolved constituents of fresh waters. This means, however, that each electrode actually simultaneously provides information about several target analytes, making ISEs ideal candidates for use in multi-sensor arrays and in combination with

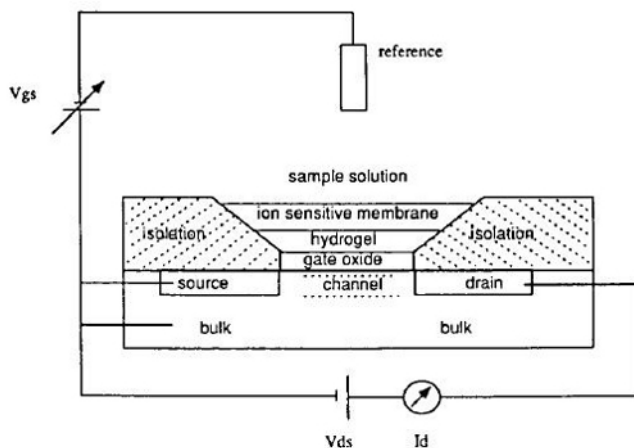


Figure 2-3: CHEMFET cross section [3].

downstream signal processing modules.

2.1.2 Amperometric sensors

Amperometric sensors, broadly defined by the IUPAC [73] as “a detection method in which the current is proportional to the concentration of the species generating the current,” encompass a wide variety of electrochemical sensors. Electrodes used in amperometric methods are generally composed of noble metals, in contrast to the materials listed above for potentiometric electrodes. The most familiar uses of amperometric techniques are the Clark cell electrode for dissolved oxygen measurement and voltammetric methods for measuring redox-active species, but a more complete overview of relevant amperometric, voltammetric, and polarographic applications is included in Table 2.1.

In the Clark cell, the applied voltage is constant, calibrated to the reduction potential of O_2 . Generally, existing amperometric sensors for analytes other than oxygen are based on the Clark cell electrode, i.e., signal transduction from the analyte of interest to O_2 occurs at the sensor membrane or in the inner solution, after which the O_2 signal is measured using a traditional Clark cell electrode. Two examples of such sensors, for N_2O and H_2S , are referenced in Table 2.1. The required consumption of the analyte of interest is a constraint to this method, usually requiring stirring to assure that diffusion through the membrane is the limiting factor. Recent work on the miniaturization of Clark cell electrodes has, however, significantly addressed this issue.

In voltammetric methods, by contrast, current is measured as a function of applied voltage (between the working and reference electrodes) while the voltage is varied across some range according to a given function. Many voltammetric techniques exist, using different series of applied voltages, including linear sweep (LSV), staircase, differential pulse, cyclic, and anodic/cathodic stripping voltammetry, with different mathematical techniques used to transform measured current values into concentrations [22]. In order to limit currents from the reference electrode (to improve stability of applied voltage and measurement of current signals), a third electrode (the counter electrode) is required for voltammetric methods. The choice of electrode material determines the range of reduction potentials that can be measured; some typical electrode choices are Hg, Pt, and C, though Au, Ir, Pd, Re, and Rh have also been used [74, 75]. Voltammetry is a popular technique because it allows the mea-

surement of several analytes with a single instrument, and it is traditionally used to identify redox-active species as a function of their known reduction potentials [50, 76, 77]. Recently, however, voltammetric techniques have been applied for the in-situ quantification of in-situ sulfur [78, 49, 76, 50], measurement of nitrogen species [79], the classification of beverages [74, 75, 80], and the identification of complex liquid media (amino acids in feed samples [81], heavy metals and organic acids [82], drinking water quality [83]). Voltammetric microelectrodes have also been used to measure O_2 and H_2S in microbial mat communities [84, 85, 86]. In some cases, multiple (4–6) working electrodes are used to obtain more information about the mixture [74, 75]. The major disadvantages of voltammetric techniques for in-situ applications are the instability of electrode drift and the required supporting equipment (in addition to the variable voltage supply and other electronics, these methods also often rely on FIA (flow injection analysis) systems where samples are pumped into a reagent/buffered sample stream that flows past the electrodes, for example).

Polarographic techniques, which use similar sequences of applied voltage, instead require an electrode with a continually-renewed surface (as opposed to the standard metal electrodes used in voltammetry). Traditionally, the dropping mercury electrode (DME) has been used in polarography, however pressures to “green” analytical chemistry may result in a phase-out of Hg-containing electrodes and are spurring research into alternative electrode options. Benefits of polarographic techniques include high precision and lower detection limits, reportedly by one or more orders of magnitude compared to voltammetric techniques. Disadvantages are similar to those for voltammetry, with the addition of use of a mercury electrode being a major limitation for in-situ applications.

2.1.3 Optical sensors

Several optical techniques are used for in-situ applications, including in-situ UV spectrophotometry [51, 52, 87] and optical electrodes (optodes/optrodes) based on absorbance, fluorescence, and “lifetime-based detection” techniques. The benefits of in-situ spectrophotometry include good detection limits (e.g. $0.2 \mu M$ for nitrate [52]) and high accuracy; however, these systems are very sensitive to temperature, limited to the detection of UV-absorbing analytes (nitrate, nitrite, and sulfite of the analytes of possible interest here), and can see substantial interference from DOC (dissolved organic carbon) and halides in natural waters. Additional analytes can be measured by first chemically manipulating the sample, but this requires the introduction of reagent supply and waste storage issues.

Alternately, optodes offer the convenience of ISEs with better stability (as function is based on equilibration of the entire membrane rather than just the interfaces [61]), easy miniaturization (e.g., through use of fiber optics [88, 89, 60]), and generally lower detection limits (by one to several orders of magnitude). Optode membranes are, in many cases, identical to ISE membranes with the addition of either chromoionophore or fluorophore to the ionophore-doped polymeric/plasticizer membrane. The ion-concentration signal is thus transduced into a pH-dependent change in absorbance spectrum or a quenching of the fluorescence signal (reviews [10, 61, 11]) which may be detected optically. Typically the intensity (or loss of intensity) is considered the signal, as it is directly proportional to the concentration of the analyte (note: this is in direct contrast to the logarithmic relationship for ISEs). A reference chromoionophore, fluorophore, or luminophore may be incorporated into the membrane as well, and improved resistance to sensor drift and sensitivity to environmental changes has been demonstrated using this technique. Optical sensor arrays have been produced for taste sensing (using absorbance [58] and fluorescence [53]) and fiber

optic arrays proposed as universal sensing platforms [88]. Finally, “lifetime based detection” measurements have been introduced using fluorescence sensors. In this scheme, either the lifetime (time until the luminescence dies out) or the phase shift of the luminescence (given modulated excitation) are measured. This information can be used to infer analyte concentration, and this methodology reduces sensitivity to drift and bio-fouling in the case of in-situ installations [57]. Benefits of optode technologies echo those for ISEs: they are relatively inexpensive, easy-to-use, and somewhat available commercially. Drawbacks are that fewer analytes have corresponding optodes produced commercially (relative to ISE selection), they still suffer significant temperature effects, and the measurement range is generally smaller than for ISEs (though the lower limit of the window can be adjusted appropriately based on the application) [90]. Ongoing work in this field has, however, produced optodes with significantly increased ranges during the last ten years [54], partially addressing these concerns.

2.1.4 In-situ sensors used for research

Several sensors have been proposed in recent years for in-situ measurement of ions, many of which are calibrated using traditional methods and models. While these sensors typically report only a single, or a few, ion concentrations, they have in many cases been able to improve detection on this single channel by taking advantage of technique specificity, knowledge of the background matrix, etc. Optical techniques such as spectrometry have been applied for quantification of nitrate in sea water [52, 51]. In limited cases, extensive lab calibration against known environmental backgrounds has allowed adaptation of ISEs for stand-alone use in environmental monitoring, including long-term (2 months) monitoring of river nitrate concentrations [91] at levels of 40-150 μM and surface water measurements of NH_4^+ and NO_3^- [92]. Microelectrode ISEs have successfully quantified Ca^{2+} and CO_3^{2-} in lake pore waters [93] while voltammetric microelectrodes have been used to measure a range of species in marine sediments [77, 94, 49, 76, 43, 95, 9] and microbial mats [96, 84, 85, 97, 98, 86]. The most extreme example is the use of a suite of ISEs on the Phoenix lander for in-situ investigation of ion concentrations on Mars [99, 100, 101, 102].

2.1.5 Commercially-available sensors

The number of sensors available commercially is substantially fewer than those described in the literature. In 1999, H. Weetall commented that “commercialization of chemical sensor and biosensor technologies has continued to lag behind the research by several years. The reasons are many. There have always been cost considerations, this includes the poor integration of many biosensors and chemical sensors into easy to use systems. Additional concerns are stability and sensitivity issues, quality assurance and competitive technologies. These issues have been identified not only as concerns, but as the major risks associated with development of chemical sensor and biosensor systems” [103]. This statement applies equally to the marketplace of today. It is, however, of great interest to produce a system using widely available sensors in order to promote feasibility, availability, and system lifetime. As such, a more complete discussion of currently commercially-available hardware and its utility for this project is provided in Chapter 3.

2.2 Chemometrics: signal processing for chemical applications

The use of multivariate signal processing techniques has become widespread in chemical applications during the last century, with use of many techniques detailed in [31, 26]. This section will provide a brief introduction to the methods most frequently used for sensor array applications, along with significant results from the literature.

2.2.1 Linear (logarithmic) models

It is important to note that many chemometric methodologies have been motivated by the use of single ISEs, or arrays of ISEs, in cases where it is necessary to retrieve single ion concentrations from a signal disturbed by the effects of multiple interfering ions. The model for such a situation uses selectivity coefficients which are calculated by the Nicolsky-Eisemann Equation as shown in Eq. 2.2:

$$\log K_{ij}^{pot} = (E_j - E_i) \left(\frac{z_i F}{2.3 RT} \right) + \log(a_i * a_j^{\frac{-z_i}{z_j}}) \quad (2.2)$$

where subscript i refers to the primary ion and subscript j refers to an interfering ion, E_i and E_j are taken at the same concentration, and all other parameters are identical to those in Eq. 2.1. For a suite of interfering ions, this equation can be rearranged as follows (Eq. 2.3):

$$a_i = 10^{\frac{E_i - E^o}{s}} - \sum K_{ij}^{pot} a_j^{\frac{z_i}{z_j}} \quad (2.3)$$

where s is the Nernstian slope of the calibration curve and all other parameters are as previously defined. Because of the form of this equation, Singular Value Decomposition (SVD) was initially considered for these applications. Recent work has, however, made clear the limitations of such tightly constrained linear models. Application of this equation requires (1) exact knowledge of the concentrations of all interfering ions and (2) knowledge of the corresponding selectivity coefficient *at those concentrations*. It has also been demonstrated that these selectivity coefficients are strongly dependent on temperature [104] and that the full equations taking these effects into account have a more complex form than given here [105]. Because a full characterization of the required selectivity coefficient matrix may be difficult to obtain, or may take an infeasible amount of time to determine experimentally, methodologies have been explored that do not require the full electrode characterization in order to produce accurate results.

2.2.2 Principal components analysis

Principal Components Analysis (PCA) is a linear method that factorizes input matrix \mathbf{M} (of m variables) into a square mixing matrix \mathbf{A} and a rectangular matrix \mathbf{P} of uncorrelated signals and rank p , where $p \leq m$, that is:

$$\mathbf{M} = \mathbf{A}\mathbf{P} \quad (2.4)$$

where the rows of \mathbf{P} (“factors”) are specified such that the first factor explains the largest fraction of the variance in the original data matrix \mathbf{M} and subsequent factors $\{2\dots p\}$ explain diminishing fractions of the variance. PCA performs well on over-constrained systems and

is useful for discovery of underlying structure in data. It has proved useful for classification problems but is limited in its application to quantification. Primarily, the variance-maximization scheme produces p uncorrelated signals that are not necessarily (and often not at all) correlated to the quantities of interest for the researcher (which may not themselves be statistically independent). A more complete primer is given in [31].

2.2.3 Partial least squares regression

Partial Least Squares (PLS) regression (also known as *Projection to Latent Structures*) is a partially-supervised methodology used for prediction of “resultants” from “inputs” and extends the multiple linear regression model for application to imperfectly constrained (often over-constrained) systems. While PCA minimizes correlation between factors, PLS maximizes correlation between input and resultant variables. PLS requires a set of training data (corresponding input and resultant matrices), and the resulting model must be evaluated on an independent validation set. PLS has been used extensively in the field of chemistry, including application to sensor array problems, as will be discussed further below.

2.2.4 Non-linear PLS and PCA

Non-linear extensions of PLS (NL-PLS) and PCA (NL-PCA) exist which take advantage of prior knowledge of the non-linear interdependence of system variables. For application to ion selective electrodes, these relationships have often been derived from the Nernst and Nikolski-Eisenman equations, with some success (e.g., the Hydrion-10, listed in Table 3.1, works on these principles). In general, however, theoretical knowledge of the system non-linearities is limited or difficult to determine, restricting the power of these methods.

2.2.5 Time domain extensions

In addition to analysis of ‘steady state’ sensor signals, many techniques have been developed to take advantage of the information present in the time-series data returned by a sensor after immersion in a new sample. (These same techniques can be applied to spectral responses as a function of wavelength.) Pre-processing methodologies compress information from the time domain (typically by representing it in the frequency domain), which reduces data size and allows it to be further post-processed by any of the algorithms described above. Such techniques relevant to the work presented here include the Fourier Transform (FFT), wavelet transform, and multi-dimensional PLS (NL-PLS2 [106]).

2.2.6 Machine learning algorithms - artificial neural networks

A great number of machine learning algorithms have been investigated and proven in the field of artificial intelligence (AI). Many of these have promise in chemical sensor suite applications due to their capabilities to approximate non-linear functions and to parse out underlying structures from example data rather than a given mathematical model, including artificial neural networks, support vector machines, and boosting. A single representative methodology, the artificial neural network, is discussed here as it has already been introduced and proven in the field of chemometrics (for in-depth discussion see [107, 30]). Only a cursory introduction is provided here, while specific functionality and implementation details are covered further in Chapter 5.

Artificial neural networks (ANNs) offer an alternative to PCA and PLA, used independently or to post-process data initially evaluated with PCA or PLS algorithms. Neural networks provide a non-linear multivariate methodology to perform estimation *without the need for knowledge of the nature of the system non-linearities, interferences, or noise*. Like PLS, the methodology is adaptive, requiring both training and validation data sets (for algorithm training and for independent evaluation of the “goodness” of the resulting network). Generally a larger training set will produce more accurate results, however it is necessary to consider (1) time required to create and process training samples and (2) time required for convergence of the model. A good training set should be large enough to capture variation expected in the applications of interest with minimal burden for development of the training data set. The validation set should also span the sample space of interest but must be independent of the corresponding training set (i.e., must not contain identical sample points) in order to provide a useful measure of the generality of the ANN predictive capabilities. ANNs have many tunable parameters, and numerous training algorithms have been developed, of which Bayesian regularization (back propagation) has been preferred in chemical applications. Daponte and Grimaldi [108] provide an excellent overview of ANNs as used in measurement systems generally, while ANNs applied to chemical systems are covered in [107, 109, 110]. A wide variety of alternative network architectures and training algorithms have been developed in AI-related fields, providing ample opportunity for expansion of current ANN uses in chemical applications. Because, unlike PCA, PLS, and their non-linear versions, ANNs are not based on a simple matrix multiplication model, the underlying structure is discussed in more detail here and in Chapter 5.

2.3 Development of sensor arrays: the “electronic tongue”

In 2006, Gunnlaugsson et al. stated “the criteria behind chemical sensing have involved the design of small single molecules that specifically recognize a single ion or a molecular species *in a competitive media* in a reversible manner and *in a given concentration range*” [111] (italics added). Unfortunately, these constraints are often the key to use of these sensors, due to the extreme difficulty of attaining perfect specificity. This is particularly true for ion selective electrodes, which are economical to produce and easy to use; however, their low cost and low power characteristics make them prime candidates for adaptation to use in environmental applications. This has led, over the past 20 years, to the development of sensor arrays, whereby the use of a number of sensor channels provides more information about the solution composition and improves predictions for the analytes of interest. These systems take advantage of substantially uncorrelated interferences on each sensor and thus can benefit from the use of non-selective or cross-selective electrodes in addition to the traditional ion-selective electrodes. Use of sensor array systems can be broadly categorized into those applied to classification and quantification problems, motivated by differing (often industrial) applications, and these are each discussed in more detail below.

2.3.1 Electronic tongue systems

Research into sensor arrays was initially driven by demand for a sensor to identify complex gas mixtures for applications in the food and cosmetics industries (for the classification of beers, meats, cheeses, perfumes, soaps, etc.) which led to the development of the first multi-sensor systems in the mid-1960s. Improved “smart” systems that mimicked the human sense of smell were introduced in the early 1980s, dubbed “electronic noses” [112, 113]; the

“intelligence” of these systems was derived from multivariate signal processing techniques such as PCA and PLS. Demand for similar identification systems for the determination of complex liquid media has come from several sectors, for application to food, medical, biological, and environmental problems. By analogy, these sensor arrays have been given the name “electronic tongue.” Following is a discussion of such systems developed to date, including details of system type (classification vs. quantification), intended analytes, and primary signal processing methodologies used.

The liquid-media sensor array was pioneered by Otto and Thomas, who developed an array of five electrodes (three selective and two cross-selective) and used PLS for the simultaneous quantification of calcium, magnesium, sodium, and potassium at physiological levels [114]. Soon after, they suggested and implemented use of ANNs in place of PLS [115, 109]. Since this time, researchers have expanded upon these original ideas by extending the hardware types, software used, or target applications.

The first such expansions were ISE-array systems using PCA, PLS, or ANNs for recognition of foodstuffs, including classification of, e.g., brand or age of wine [116, 117, 118], vinegar [119], water [116, 117, 120, 121, 122], fruit juices [80, 120, 123, 124, 125, 126], milk [75, 124, 126], soft drinks [121, 123], beer [121, 127, 128], tea [120, 123], and coffee [123]. Recognition of medical conditions based on ISE measurements was also attempted [129]. In most cases, subclass separation was achieved with success above 80-90%. Similar voltammetric systems with several working electrodes were proposed for classification of milk [130] and drinking water [83]. Section 2.3.3 discusses improvements suggested or used by researchers in cases where separation was incomplete.

Quantification by ISE-array was subsequently introduced for heavy metals [131, 132, 133, 106, 134, 135, 136], inorganic pollutants in modeled groundwater (Mn(II), Fe(III), Ca^{2+} , Mg^{2+} , Na^+ , Cl^- , and SO_4^{2-}) [133, 137], surfactants [138], total ions [121], components of wine and bottled water [139, 116, 118], small sets of inorganic ions [110, 140, 141, 142, 143, 93, 144, 145, 146, 147, 148, 149, 150], and biological liquids (Ca^{2+} , Mg^{2+} , Na^+ , HCO_3^- , Cl^- , H^+ , and HPO_4^{2-}) [151, 114]. Errors can be relatively low (<20%), although the best results are generally seen in biological applications (errors <5% achievable) where the background matrix is known and fairly constant. Generally, these systems have been comprised of potentiometric sensors (ISEs with inner reference solutions or of the all-solid-state variety) with cross-selective electrodes used to improve predictive ability. For example, inclusion of cross-selective electrodes allowed the determination of Mn(II), Zn(II), and Fe(III) by an electronic tongue (using ANNs) **even when none of the included electrodes were specific to these species** [133]. Training samples for calibration of the multivariate data processor are nearly always comprised of lab-created standard solutions. All systems listed above used ANNs to process the matrix of sensor signals, although identical PLS systems were often developed for side-by-side comparison [116, 118, 131, 137, 121, 152]. In cases where PLS (or NL-PLS) was compared to ANN for the same data sets, no definitive trend is visible: PLS produced better results in [121], similar results in [116, 118], and worse results in [131, 137, 121], and artificial neural networks have thus become the de facto algorithms of choice for electronic tongue applications in recent years.

2.3.2 Innovative electronic tongue systems

Expansions upon this initial idea have come in many forms. I will discuss here the contributions that I feel have been the most important for the field in the past 10–15 years: use of time domain information in signal processing, use of novel or combined hardware

architectures, and a move toward in-situ deployment of electronic tongues.

Novel signal processing methods

In addition to the well-tested algorithms such as PLS and ANNs, researchers are exploring application of a wide variety of other signal processing methods to the field of chemical sensing. Mixes of K^+ and NH_4^+ were analyzed using Bayesian methods based on the Nernst and Nikolski-Eisenman equations [153, 154], however this method requires prior knowledge of the number of ions in solution and their charge. Case-based reasoning (CBR), a method approximating a weighted nearest neighbor scheme, was applied to a 4-electrode array (pH, Ca^{2+} , NO_3^- , conductivity) for classification of ‘fertigation’ type samples [155] with inconclusive results.

Application of methods taking advantage of three-dimensional data, discussed in Section 2.2.5, have also been applied to chemical sensor suites with more promising results. A combined wavelet plus ANN method was applied to a voltammetric system for quantification of amino acids [81]. FFT pre-processing combined with ANN analysis has been shown successful for analysis of metals [148, 135] and cations [149, 150, 136] using ISEs. In the final case, use of the FFT stage was shown to improve estimates of NH_4^+ from 38% to 11% error and estimates of NO_3^- from 36% to 11% error. Three-dimensional PLS (NL-PLS2) has also been used with an ISE array to quantify metals [156] with (4.2, 0.09, 0.9) μM mean errors for (Cd^{2+} , Cu^{2+} , and Pb^{2+}).

New sensor hardware types

New hardware alternatives to traditional ISEs have been tried in electronic tongues for a number of applications. ISFET / miniaturized integrated thin-film arrays of multiple ISEs have been used for both qualitative analysis of water [122] and quantitative analysis of metal concentrations [134]. Screen printed ISEs with PLS data processing have recently (2008) been demonstrated for quantification of NO_3^- and NH_4^+ in fish tanks within 5% for levels down to $\sim 50\mu M$ (water is actively pH managed for fish health) [144]. Micromachining of silicon wafers and new LAPS (light-addressable potentiometric sensors) techniques are further promoting the minaturization of hardware, which has promise for improving stability and response time. EIS (electrochemical impedance spectroscopy) has been explored for quantification of K^+ , Na^+ , and NH_4^+ [141], and chronocoulometry (cyclic voltammetry, which calculates responses based on charge transfer rather than current) has been attempted for calculation of NO_3^- concentrations [157]. Capillary electrophoresis has been coupled with potentiometric detectors for Na^+ , K^+ , Ca^{2+} , and NH_4^+ in the $\sim 10\mu M$ range in an effort to move away from the high power requirements of lamps required for the corresponding optical techniques [158].

Combined hardware architectures:

Many creative and effective electronic tongues have also been constructed through combinations of existing technologies. Chemical oxygen demand (COD) was estimated with $\sim 1.5\%$ relative error using ANNs to process information from a UV spectrum, a temperature sensor, and a conductivity sensor [87]. Classification of fermented milk (*kefir*) was accomplished using ANNs/PCA on the output of potentiometric (pH, CO_2 , and Cl^- ISEs), voltammetric (6 working electrodes), and conductivity sensors; in this case ANNs were able to separate class types while PCA was not [75]. Finally, indirect temperature compensation

(i.e., via the ANN processing rather than an applied calibration) for ISEs has been demonstrated after inclusion of a temperature sensor as an additional input to neural networks [159, 160, 161]. These cases all represent significant contributions to in-situ deployment of electronic tongues and are discussed in more detail in the following section.

Environmental in-situ applications

While not many, there have already been a number of ‘electronic tongue’ systems described in the last ten years that have been designed for and tested under in-situ conditions. These systems have each pushed the envelope for their respective fields, and they evidence a trend toward the possibility of in-situ measurement of an even larger number of analytes at lower concentrations. The salient features, and most significant challenges, with each of these systems are compiled here.

The earliest in-situ ready systems described in the literature (ca. 2001) were purely voltammetric. Such a method using four metal working electrodes was used for water quality monitoring at a drinking water plant [83] with outputs clustered using PCA. (Ion concentration quantification was not targeted in this application.) While results were promising, significant issues were encountered with electrode drift such that results on consecutive days could not reasonably be compared. The researchers concluded from this that hybrid systems, using more than one type of measurement technique, would be a superior design, and in fact, they subsequently implemented the hybrid system described above [75]. This same group also installed a voltammetric system for in-situ monitoring of milk at a dairy at several stages of the processing line [130]. Even within the tight constraints of electrode materials allowed due to sanitation concerns (they were forced to omit the reference electrode from the system design), it was possible to correctly separate milk samples at different stages and to separate these from cleaning and sanitation periods.

Subsequently, two ISE-based electronic tongue systems have been described by del Valle et al. [159, 160, 161] for monitoring of environmental anions and cations and also integrating temperature sensors to compensate for diurnal temperature variations. The first is a flow-injection analysis (FIA) system (samples are injected into a known carrier fluid) adapted for installation in a weather-protected greenhouse to monitor a fertigation (irrigation/fertilization on artificial soils) system, namely to measure NH_4^+ , K^+ , NO_3^- , Na^+ , Cl^- , and optionally phosphate [159, 160]. Target concentrations of all constituents were in the 2-20 mM range. In addition to the 8-12 ISEs (some specific, some generic in response), a temperature probe was included in the sensor array and as input to the ANN used to process the data. Training samples were subsequently run in three distinct temperature regimes (10°C, 24°C, 34°C). In addition, the base solution for training samples was a mixture (50%/50%) of waters taken from the two relevant samples streams in the greenhouse to minimize background interference from other sample constituents. Finally, the day number of the sample (counting from initial calibration) was included as an input to compensate algorithmically for any drift in the electrode signals [159] and measurements were also taken of a reference solution daily [160]. In practice, the base solution for the training sample was not actually representative of typical solution composition (due to the gross variations between the two mixed streams), and thus the training of the ANN did not fully meet expectation [159], although subsequent training with different mixes (e.g., 50% distilled water) produced significantly improved results (RMSE < 8mM). Final relative errors achieved (calculated on the logarithm of the data) were (11%, 15%, 15%, 15%, 24%, 32%) for (Cl^- , K^+ , Na^+ , NO_3^- , NH_4^+ , PO_4^{2+}). Significantly, this methodology also successfully compen-

sated for diurnal temperature fluctuations in the greenhouse, producing accurate, steady readings for the analytes of interest over a period of weeks, demonstrating the utility of including non-analyte data streams as a component of the neural network input.

Subsequent projects in this same lab (ca. 2008) targeted measurement of similar ion sets in two different situations, with the additional goal of linking up wireless monitoring via Bluetooth and radio links to these data stations [161]. The first application was monitoring of (Cl^- , K^+ , Na^+ , NO_3^- , NH_4^+) in a Continuous Stirred-Tank Reactor (CSTR) set up to simulate natural biodegradation conditions and ‘shocked’ with a concentrated influx of fertilizer with a simulated surface water background. The second application was in-situ monitoring of (K^+ , Na^+ , NH_4^+) at a eutrophic reservoir in Mexico. ISE distributions similar to that described above were used, along with a temperature sensor to compensate for temperature changes. Training samples included mixes of lab standards as well as representative water samples from the application sites. Concentrations were in the range of 0.5-15mM in all cases excepting NH_4^+ in the second application which was trained down to $\sim 8 \mu\text{M}$. The reported relative error for the more challenging second application, as computed on the logarithms of the concentrations, was (4.3%, 3.2%, 8.4%) for (NH_4^+ , K^+ , Na^+) – approximately 30–40% error in concentration space for the published concentration ranges – however errors were primarily one-sided showing a systematic bias due to the influence of the total salinity of these waters. The primary recommendation of the researchers was thus the need for inclusion of a more complete set of ISEs to simultaneously quantify the other components of the waters.

Finally, publications describing in-situ use of two commercial instruments must be acknowledged, though less information is available about the design or functionality of these systems. First, in-situ monitoring of groundwater in Tunisia, including Cl^- , K^+ , Na^+ , NO_3^- , NH_4^+ , Ca^{+2} , Cd^{+2} , and F^- , is described in [162]. ISEs selective for these analytes were connected to an ELIT 9808 IonAnalyser which directly displays concentration data for the user on a laptop via proprietary software. Information is not available regarding the algorithms used by this software, however the manual specifies that only individual calibration of the electrodes is required, making it unclear whether electrodes signals are corrected for expected cross-interferences. In addition [163] describes measurement of samples from areas of agricultural runoff using the Hydrion-10 ISE multiprobe and compares these results to traditional lab analysis (ICP-MS, CE). This probe uses a proprietary version of NL-PLS in addition to some chemical calculations (e.g., the carbonate system concentrations are determined based on pH and measurement of dissolved CO_2). Researchers found the best agreement for NO_3^- and good agreement for most analytes, though they cite systematic error in the Na^+ and Cl^- measurements.

2.3.3 Summary of signal processing algorithms, relative utility

As discussed above, PCA, PLS, and ANNs are the primary methodologies currently applied to sensor arrays in the development of electronic tongues. This section provides a brief summary of the relative utility of each and of the uses *in chemical applications* explored in the current literature and detailed above.

Generally, PCA has been mentioned as the preferred algorithm for classification problems, however it has been shown that this method will not resolve distinct classes in all cases. In these instances, PLS [80] and ANN [75, 126] have been shown to produce better results. Uniquely, PLS and ANN techniques have even been combined into a two-stage classification engine (PLS pre-processing data for input to the ANN) which produced better

results than either algorithm alone [152].

Quantification systems have conversely relied upon PLS and ANNs, with limited exploration into two-stage PLS+ANN hybrid methodologies. Despite the inconclusive results in side-by-side comparison studies, ANNs are generally considered the de facto method of choice for these applications today and a number of studies have been done on the characterization and optimization of this method. Most frequently, feed-forward network architectures with Bayesian Regularization (back propagation) methods for training are used. The number of hidden layers is typically one or two. The effects of the number of inputs and outputs have been investigated in many cases, and in some instances it has been shown that several single-output ANNs can provide better predictions than a single multiple-output ANN [81, 133, 137]. Two-stage methods using PCA to pre-process (compress) data before use in a neural network have been investigated but have demonstrably worsened the results relative to a neural network alone [140] in some cases, unless all principal components are included [164, 152, 140, 165]. Non-linearities in the data, of importance to the identification problem, may often be shunted to the lowest-ranked principal components, thus removing information otherwise useful to the ANN algorithm. In one case, however, a combined PCA/ANN method was shown to have comparable results to a PLS regression for the same data [166]. Importantly, comparative studies have shown ANN-processed sensor arrays produce significantly improved output signals relative to their component sensors used individually [110, 131, 137]. Continued experimentation with ANNs for these applications has produced substantial improvements in the past decade and provides a solid starting point for future work on ANNs applied to sensor arrays.

2.3.4 Shortcomings of current chemical quantification systems

While significant progress has been obtained with the systems detailed above, there remain challenges to be addressed. To date, of the electric tongues described above, only the Hydrion-10 has been commercialized, with mixed results, leaving a substantial residual demand for systems that can perform multi-analyte identification quickly, accurately, and reproducibly. Development of a system using commercially-available sensors as array elements should address both of these difficulties while increasing the quality of sampling possible with these currently available products.

Additional constraints of current electronic tongue technology include (1) limited use of multiple sensor technologies (“hybridization”), (2) limited use of non-analyte sensors (e.g. pH, temperature, conductivity, ORP), (3) use of buffer-based standard samples for training data sets (as opposed to samples with representative background concentrations), and (4) limited exploration into alternative machine learning algorithms or ANN architectures. Investigation into methods for overcoming these challenges and evaluating the utility of these possibilities is a major driver in the development of this thesis project.

Chapter 3

Hardware Setup

Abstract

Commercially available sensor hardware is often poorly adapted to use in the field, used for measurement of only a single analyte (or a few analytes), and generally coupled with proprietary hardware or software post-processing on the primary signals. To overcome these limitations in creation of hardware for simultaneous measurement of the charge balance of natural waters, a suite of commercial sensor hardware is combined with custom electronics to create an in-situ ready instrument for measurement of the charge balance of fresh waters. Sensors are evaluated on selectivity, cross-interferences (can provide additional information), and detection limit, and the resulting sensor suite (11 ion-selective electrodes, 1 temperature probe, and 1 electrical conductivity meter) is described. Custom circuits overcoming obstacles such as high impedance of ISEs (errors in measurements due to current loading, risk of both high and low-frequency electrical noise pickup in ISE circuits), possible instability of the reference electrode, referencing of 11 ISEs to a single reference electrode, and need for digitization of data. Circuit diagrams, including custom PCB and LabView software, are provided.

3.1 Sensor selection

The architecture proposed in this thesis involves use of a suite of electrodes for quantification of major (and some minor) ions in fresh waters. Selection of the appropriate sensor hardware was based on the following:

1. Commercial availability
2. Low power requirements (< 1 W)
3. Detection limit low enough for quantification at environmental levels
4. Relative strength of response to primary analyte (vs. interferences)
5. Minimization of interferences
6. Availability of independent information in response to interfering ions, if any
7. Response time (e.g., 5-10 min. sampling time)

The following tables contain an overview (representative subset) of specific commercially-available hardware (Table 3.1) and the details of the sensor hardware actually selected for this project (Table 3.2). Of particular note in Table 3.1 are the YSI and Hach ISE probes for their respective multi-sonde systems (YSI 6820, Hach Hydrolab). Both systems were

released in only the last few years (~2010-11), and at this time, information regarding their functionality is available primarily through the manufacturer data sheets. Both systems report similar uncertainty values (maximum of $\pm 10\%$ or $2 \text{ mg-N}\cdot\text{L}^{-1}$ for NO_3^- and NH_4^+) which translate into detection limits of $\sim 143 \mu\text{M}$ for the nitrogen species. Keeping in mind that the mean concentrations for natural surface waters (fresh) is in the range of $0.5\text{-}1.5 \text{ mg-N/L}$, it is clear that these probes are not yet able to measure nitrogen species *at natural levels*. Additional challenges mentioned in the manufacturer literature includes temperature correction (slope of the response curve is expected to be a strong function of temperature - both companies attempt to compensate for this in software) and correction for ionic strength effects (not mentioned in the YSI literature, accomplished by Hach by converting measurement of electrical conductivity to ionic strength via assumption of a ‘typical’ river water ion distribution). Neither product contains correction for interfering ions, and recommendations are made to verify values through lab analysis of grab samples for any locations where the apparent concentrations are higher than expected. Essentially, this means that the user cannot discriminate in the field between a high nitrate concentration and interference on the electrode by another ion species, i.e., adaptive sampling to follow gradients will not be reliably possible with such instrumentation.

Because of the shortcomings in commercially-assembled electrode arrays - or due to the inability of accessing hardware without use of proprietary software - the ISE array used in this thesis was custom assembled. All hardware selected for this application was available in half-cell configuration (no built-in reference electrode) with BNC-type connectors for optimal noise protection. ELIT electrodes, generally available as solid-state ISEs, were found to be reliable and easy to use (short conditioning requirements, relatively fast response time, low drift) and available with a compact 6-electrode (plus single external reference electrode) mounting head. A mix of solid state and glass ISEs was chosen, as interferences were expected to be at least partially uncorrelated (i.e., strength of ion interferences for the glass Na^+ and solid state Na^+ ISEs was expected to vary). No solid state pH electrodes with BNC-type (direct voltage measurement) connectors were commercially available at the time of hardware procurement, so a single glass electrode was used for this analyte. No commercial electrodes were available for magnesium or sulfate, although their presence is expected to be detected as an interference by other ISEs in the sensor suite. In addition, a hardness (divalent cation) electrode is included to measure both calcium and magnesium, and an ISE marketed for both lead and sulfate (membrane contains a proprietary derivative of sulfate) is investigated. While a carbonate electrode was purchased and tested, preliminary results showed that it did not produce stable results in the concentration ranges of the given samples; as such, signals from this channel were omitted from all analyses.

Table 3.1: Overview of commercially-available sensors of interest for this application (accurate as of 2009; manufacturers such as WPI and YSI released some additional ISE-based instrumentation in 2010-11). Analytes listed are not comprehensive and are intended to be representative of quantities of interest for this application.

Manufacturer	Sensor	Method	Analytes Measured
<i>WPI</i>	ISE	ISE (PVC)	NH_4^+ , Ca^{2+} , HCO_3^- , Cl^- , NO_3^- , NO_2^- , K^+ , Na^+
<i>NexSens</i>	WQsensors	ISE	Ca^{2+} , Cl^- , NH_4^+ , NO_3^-
<i>ELIT</i>	ISE	ISE (solid state, PVC)	NH_4^+ , Ca^{2+} , Cl^- , NO_3^- , NO_2^- , K^+ , Na^+ , S^{2-}
<i>YSI</i>	YSI 9600 YSI 6820 (multiprobe)	spectrophotometry ISE	NO_3^- , (H_2PO_4^-) NH_4^+ , Cl^- , NO_3^-
<i>Hach</i>	Hydrolab (multiprobe)	ISE	NH_4^+ , Cl^- , NO_3^-
EnviroTech	NAS-3X / EcoLAB	colorimetric (wet chemistry) + spectrophotometry	NO_3^- , H_2PO_4^- , NH_4^+
Systea	NPA / DPA	colorimetric (wet chemistry) + spectrophotometry	NO_2^- , NO_3^- , NO_2^- , H_2PO_4^- , NH_4^+
Satlantic	ISUS (SUNA)	UV-absorption	NO_3^-
Merck	ELITE	ISE	NO_3^-
Orion Industrial		ISE	Cl^- , Na^+ , NH_4^+ , Ca^{2+}
Radiometer-Tacussel			Cl^-
Topac	GAT 4000	polarographic, voltammetric	NO_3^- , NO_2^- , Cl^-
WTW	R 503	ISE	Ca^{2+} , Cl^- , K^+ , Na^+ , NO_3^- , S^{2-} , NH_4^+
Hydrion bv	Hydrion-10	ISE array using NL-PLS	Temperature, EC, pH, K^+ , Na^+ , Ca^{2+} , NH_4^+ , Cl^- , NO_3^- , CO_2 , HCO_3^-

Table 3.2: Sensor hardware incorporated into the sensor suite.

Analyte	Membrane	Manufacturer / Model	Published LOD
Ca ²⁺	Solid-state PVC polymer matrix	ELIT 8041	0.50 μ M (0.02 ppm)
K ⁺	Solid-state PVC polymer matrix	ELIT 8031	10 μ M (0.4 ppm)
Na ⁺	Solid-state PVC polymer matrix	ELIT 8230	2.0 μ M (0.05 ppm)
NH ₄ ⁺	Solid-state PVC polymer matrix	ELIT 8051	2.0 μ M (0.03 ppm)
Cl ⁻	Solid-state poly-crystalline	ELIT 8261	30 μ M (1 ppm)
NO ₃ ⁻	Solid-state PVC polymer matrix	ELIT 8021	5 μ M (0.3 ppm)
CO ₃ ²⁻	PVC 'Dry Contact'	Thomas Brand 4230A37	0.13 μ M (0.008 ppm)
Pb ²⁺ (SO ₄ ²⁻)	Solid-state (unspecified)	HI 4012	unspecified
Cl ⁻	Solid-state (unspecified)	Hanna Instruments 4007	50 μ M (1.8 ppm)
Hardness (divalent cation)	Plastic (unspecified)	Thermo Sci. 9332BNWP	6.0 μ M
pH	Glass	Thermo Sci. 9101BN	pH 0 - 14
Na ⁺	Glass	Ross 8411BN	1 μ M (0.02 ppm)
Reference	double junction CH ₃ COOLi	ELIT 003N	

Figure 3-1 shows the physical setup of the sampling system; all components are shown in the lower left (from right to left, ISE hardware, BNC interface wall, isolation/low-pass filter hardware, data acquisition, PC for recording and display via LabView - all components will be described subsequently in this chapter). The ISE hardware is highlighted in orange, while the custom circuitry is highlighted in yellow.

A wiring diagram for the BNC interface 'wall' is given in Fig. 3-2. Note that this is the layout view when facing the 'wall' from the sensor side. Tables 3.3 and 3.4 contain wiring information for, respectively, (1) inputs from ISEs to the isolation hardware (letters refer to locations in Fig. 3-2) and (2) outputs from the isolation hardware to the data acquisition board (numbers listed are the I/O pin number for connection on the data acquisition board).

3.2 ISE-to-PC isolation and filtering circuitry

Because of the system architecture (11 ISEs measured relative to a single reference electrode) and due to the lack of transparency in most commercial software, signal conditioning circuitry was custom designed for this application and coupled with an off-the-shelf data acquisition unit programmable using LabView software. Importantly, this allows direct knowledge of how signals are pre-processed (or not) such that the subsequent signal pro-



Figure 3-1: Photographs of lab setup for ISE sampling, including ISE hardware (orange), custom circuitry (yellow), data acquisition (teal), and PC for LabView interface (far left).

cessing can be optimally coupled with the available data. While the Hydrion-10 hardware has been designed with many of the same considerations, the proprietary software, short lifetime of the reference electrode (\sim months due to a constant outward flux of filling solution), and 8-minute pre-programmed measurement window (for further understanding of why this may not be optimal, see Chapter 4) made it a less flexible or optimal option than use of a custom system.

Because of the high output impedance of both glass (10s of $M\Omega$ s) and solid state (100s of $k\Omega$ s to a few $M\Omega$ s) ion-selective electrodes, measurement of accurate and noise-free signals requires appropriate physical shielding and matched input circuitry. Specifically, such sensors can be sensitive to capacitive loading and coupling of AC signals in addition to being easily loaded by small leakage currents running between the ion selective and reference electrodes. They may also suffer in cases where ISE-to-ISE (or other sensor) pathways promote even small, potentially unexpected, leakage currents.

Several efforts have been integrated in the design of this hardware setup to address these issues, including the following:

1. Choice of a sensor suite requiring only a single reference electrode.
2. Use of a secondary glassy carbon ‘ground’ electrode.
3. Installation of a grounded copper Faraday cage surrounding electrodes and electrode-

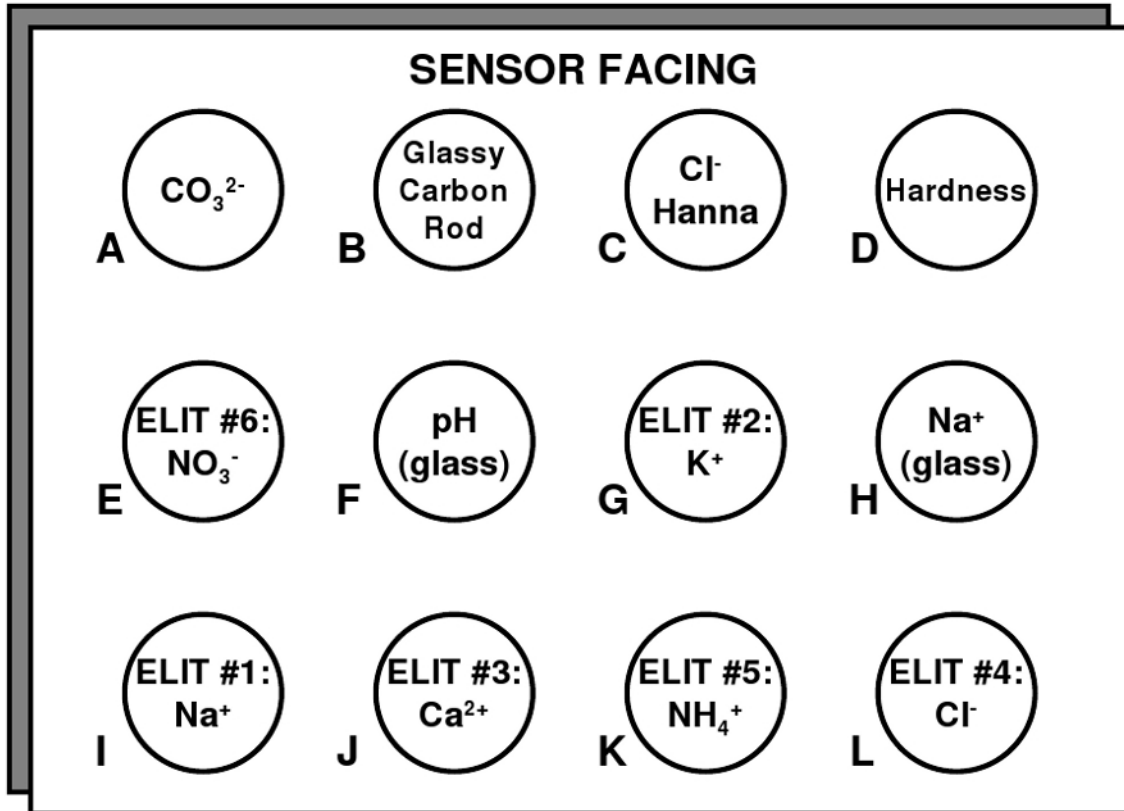


Figure 3-2: Wiring diagram for BNC interfaces from ISEs to isolation hardware.

to-coax-wire connections.

4. Asynchronous measurement of electrical conductivity to avoid interference in ISEs due to the imposed currents.
5. Creation of a custom isolated input / low-pass active filter stage inserted between sensor and data acquisition hardware.

Use of a single reference is important to minimize stray currents conducted via the sample solution. Installation of the ‘ground’ electrode (connected directly to the ground terminal of the DC power supply) also serves to minimize current through reference and ion-selective electrodes; this architecture allows measurement of the voltage at the reference electrode identically to measurement of the voltages at the ion-selective electrodes. The last item is of particular importance because it achieves appropriate impedance matching be-

Table 3.3: Physical layout of inputs to LPF Stage from Coax Plug Wall.

<i>Level</i>	Input Port				GND
	4	3	2	1	
<i>4</i>	A	LiAc ref	C	D	Glassy Carbon
<i>3</i>	E	F	H	G	
<i>2</i>	I	J	K	L	
<i>1</i>					SO ₄ ²⁻

Table 3.4: Physical layout of outputs from LPF Stage to Data Acquisition I/O Pins. Note: * value connected to all ‘-’ inputs for used analog input ports.

<i>Level</i>	Output Port				
	GND	4	3	2	1
<i>4</i>	AGND	58	32*	47	49
<i>3</i>		31	29	26	24
<i>2</i>		21	19	17	15
<i>1</i>					56

tween each of the adjacent stages. The high output impedance of the ISEs is matched using an ‘Ultra Ultra-Low Input Current’ operational amplifier (*op amp*) input stage (LMC6001: input current approx. 25 fA, input resistance >1 Tera Ω) such that the circuitry imposes virtually no current load on the ISEs. The low output impedance LMC6001 is connected directly to the Gohm input impedance of the analog inputs on the National Instruments data acquisition system (detailed further below); this architecture allows any load imposed by the data acquisition system to be supplied by the power supply via the op-amp circuit rather than via the ISEs. While design of this system was undertaken independently, many of the design decisions have been recently validated by comparison to the architecture used in [160].

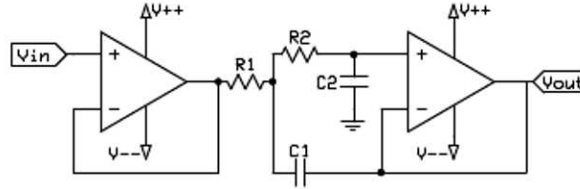


Figure 3-3: Two-pole Butterworth filter.

In addition to providing impedance matching and in response to ill-defined behavior identified in initial experiments, the custom circuitry also implements a low-pass filter (cutoff 10 Hz) to remove any unwanted signals coupled into the sensor hardware by, e.g., 60 Hz power running to other instruments in the lab. An active two-pole Butterworth filter (Fig. 3-3) was used; this type of filter provides for maximal flatness in the passband when the following conditions are met:

$$R_1 = R_2 = R \quad (3.1)$$

$$C_1 = 2 * C_2 \quad (3.2)$$

In addition, response is governed by the following:

$$\omega_{3dB} = 1/\sqrt{2 * R * C_2} \quad (3.3)$$

In this case, the requirement of $\omega_{3dB} = 10Hz$ leads to the constraint:

$$R * C_2 = 7.071 * 10^{-2} \quad (3.4)$$

Values for components were chosen as follows: $R_1 = R_2 = 100k\Omega$, $C_1 = 1.5\mu F$, $C_2 =$

0.68 μ F. The circuit diagram for high input impedance (isolation) and low-pass filter stages is provided as Fig. 3-4; the corresponding custom printed circuit board (PCB) schematic is given in Fig. 3-5. Note that each of these corresponds to one ‘level’ in the above diagrams, i.e., processes a maximum of four ISE inputs; the complete system requires four identical copies of this circuit.

Power requirements for all ISEs and signal conditioning electronics are < 10 mW, i.e., can be powered by four 9V batteries for approximately 30-35 hours of continuous operation. Subsequent sections describe the data acquisition hardware and software, currently run on a desktop PC in the lab; these have not yet been optimized for minimal power consumption, however doing so is still expected to result in a field-ready device consuming < 100 mW.

3.3 Data acquisition hardware and software

Signals output from the low-pass filter stage were fed into a National Instruments NI USB-6218 data acquisition module (32 differential analog inputs, 16-bit analog-to-digital conversion, maximum sampling rate of 250 kS/s). Signals were input, displayed, and logged using a custom LabView program.

The program developed in LabView manages the following tasks:

1. Sequential sampling of each of the sensor channels (12 in total).
2. Analog-to-digital conversion of ten samples per second per channel; these ten samples are averaged to produce a single sensor reading per second.
3. Addition of a timestamp and writing of all sensor data to a log file for later analysis.
4. Display of current sensor values (in mV), along with display of averages for last 50 and last 100 samples on each channel. Agreement in the current, ‘last 50’, and ‘last 100’ values is used as visual confirmation that sensors are approaching equilibrium.
5. Display of a plot of each sensor’s sampling history.

The LabView interface is presented in Figs. 3-6 and 3-7. Figure 3-7 shows the custom LabView program used to manage data acquisition, sampling frequency, signal averaging, saving data to file, and displaying data and graphs in real time for the user. Note that blocks with ‘film’ type edging correspond to ‘case statements’ that contain information on how to process data uniquely based on the input index (i.e., have code to display information from the eleven ISEs differently) but can only display one case to the user at a time. Visible code (case 10 for the top block and case 11 for the bottom block) are representative of the code that is not displayed excepting that name labels are assigned uniquely for different ISEs. The user interface is shown in Fig. 3-6 during a representative sample run chosen to show different signal types expected during measurements. Some signals are extremely steady (e.g., pH, which is being calibrated here), while others may drift systematically or seemingly randomly if not immersed in a solution containing ions to which they respond. (In this case, several sensors were not immersed in the fluid because pH calibration solutions are known to cause sluggish response in some ISEs after immersion.) The top portion of the screen shows the most recent reading along with readings averaging the past 50 and past 100 (approximately 1 minute and 2 minutes) measurements, useful for identifying how steady readings are at any given time. The counter at the right indicates the total amount of samples taken since sampling started (approximate correlation with seconds).

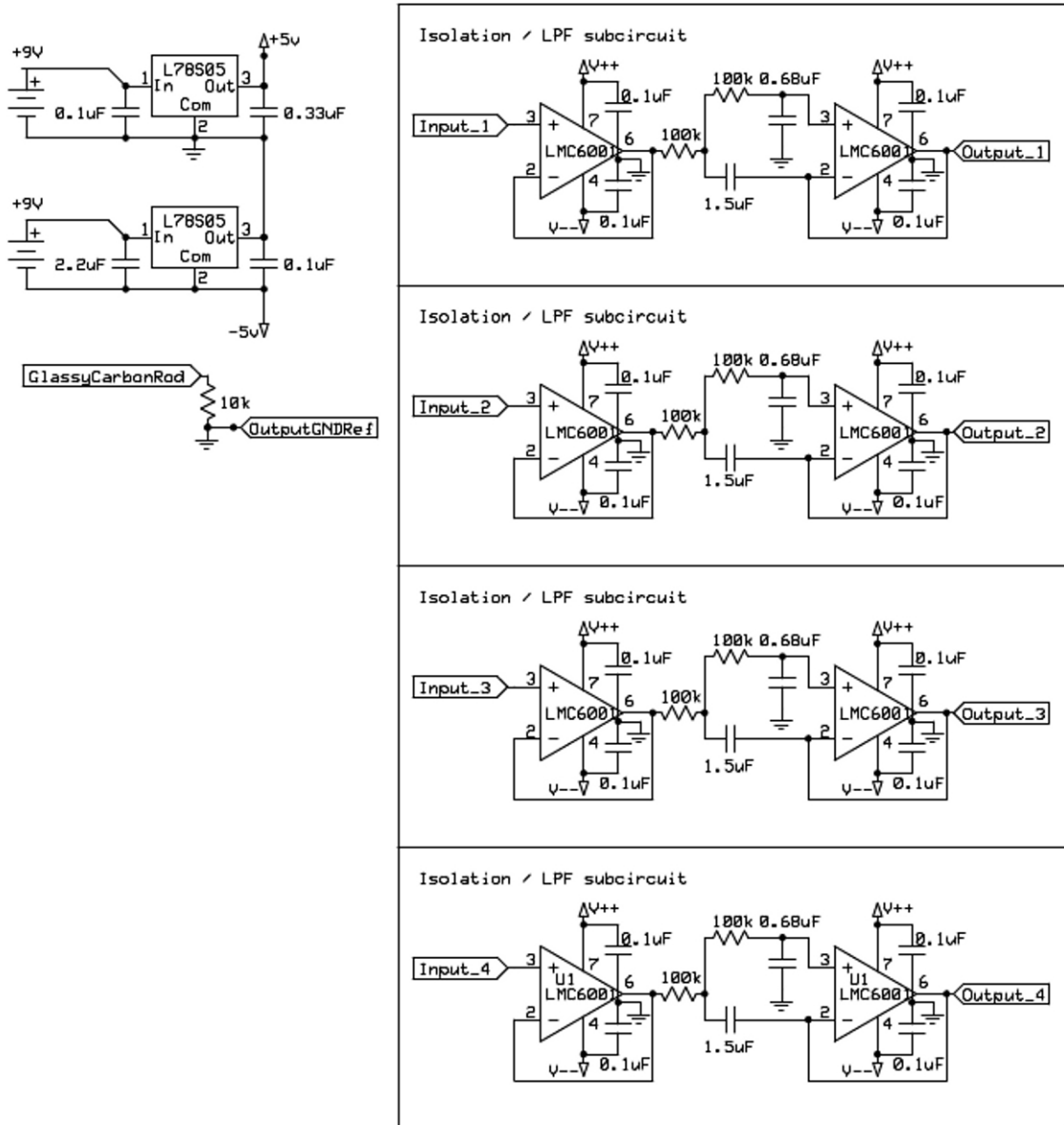


Figure 3-4: Isolation input / low-pass filter circuit schematic.

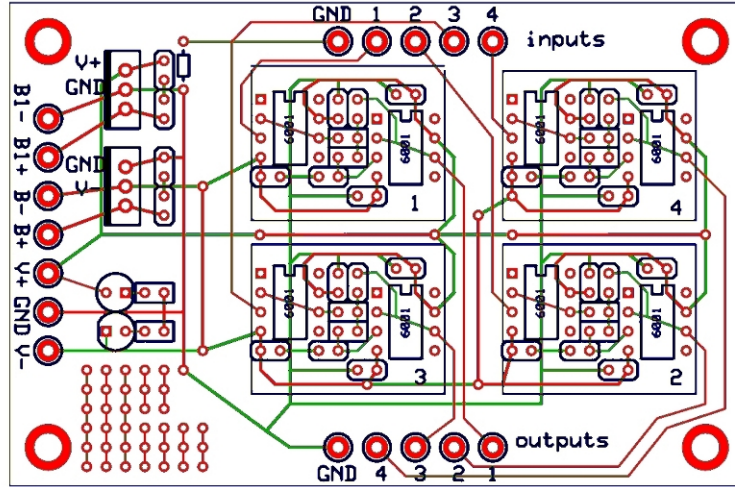


Figure 3-5: Isolation input / low-pass filter PCB. Red traces are on the top layer of the PCB while green traces are on the bottom layer. Black encodes component outlines and text printed on the PCB screenprint layer.

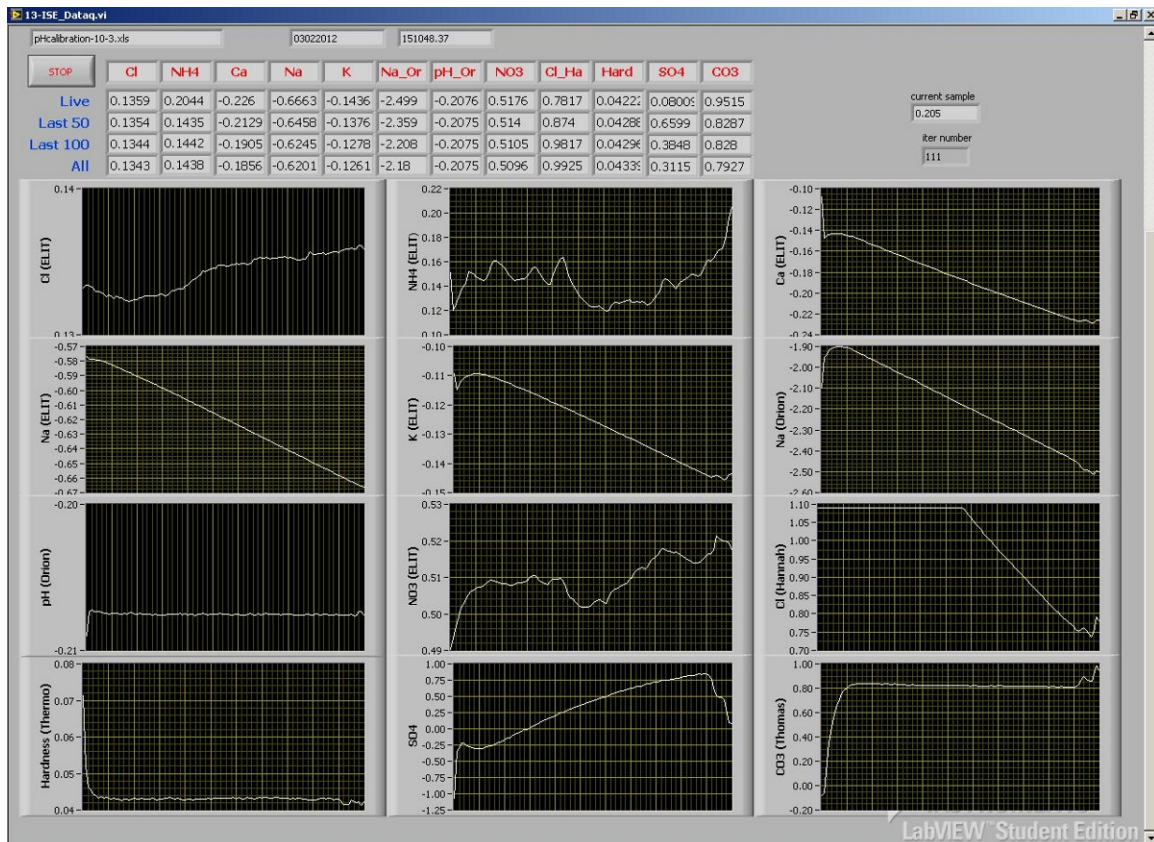


Figure 3-6: LabView user interface.

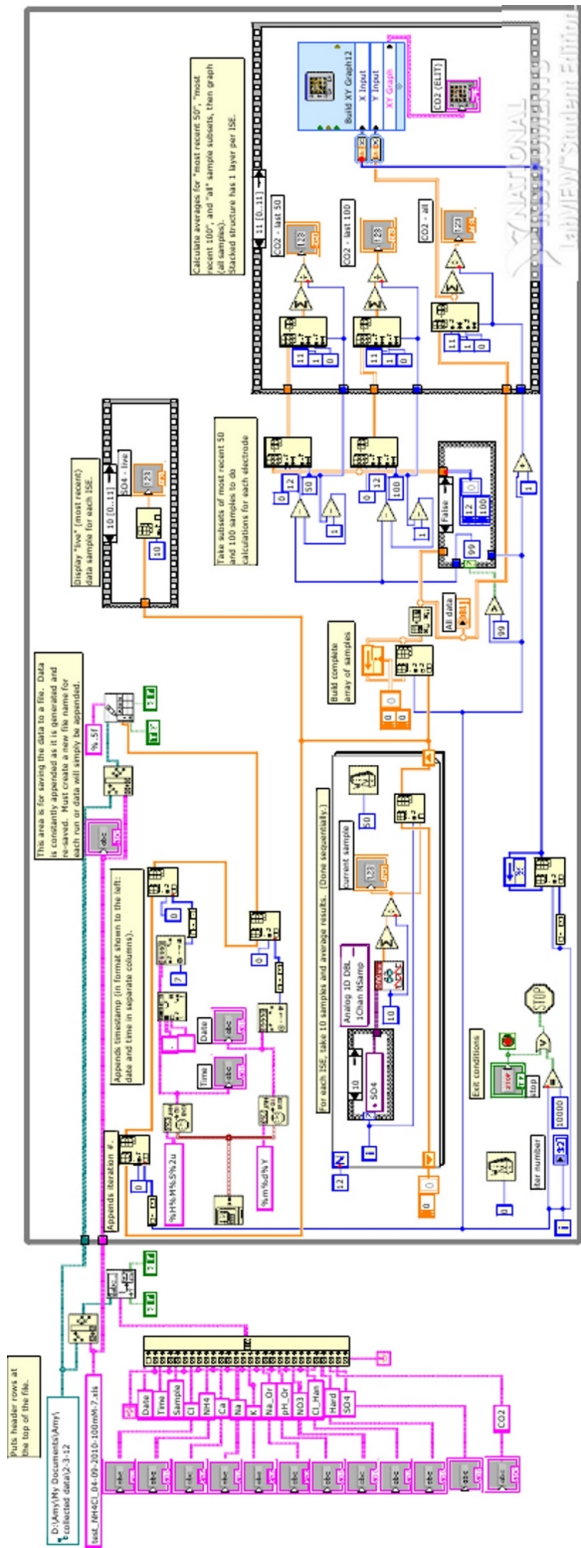


Figure 3-7: LabView schematic for data acquisition.

Chapter 4

Determination of Equilibrium Potential for Ion-Selective Electrodes

The text here is adapted from [167]: Mueller and Hemond, *Towards an Automated, Standardized Protocol for Determination of Equilibrium Potential of Ion-Selective Electrodes*.

Abstract

An automated real-time method for determination of ISE steady state value and response time is developed, following most recent IUPAC recommendations. Specifically, detection of the ‘steady state’ is related to (1) the time derivative of the emf as it reaches a limiting value ($\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}$, e.g., $0.1 - 1.0\text{mV} \cdot \text{min}^{-1}$) and (2) the duration of time for which the absolute value of the time derivative remains less than this limiting value (stability window, denoted \mathbf{win}_{st}).

A suite of representative ISEs, including glass, solid state, and polymer-based electrodes, is examined to determine sensitivity of results to parameterization choice. Measurements taken over a wide range of concentration values and in un-processed samples (i.e., without use of ionic strength adjustment) provide insight into behavior of ISEs in applications where analyte concentrations span a wide range and/or sample pre-processing may not be an option, e.g., use of sensors for in-situ environmental sampling. Results show that declared steady state emf is strongly sensitive to variations in $\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}$ but relatively unaffected by changes in the stability window when $\mathbf{win}_{\text{st}} \geq 30\text{sec}$. Linearity of calibration curves produced, quantified by root mean squared error (RMSE) against a linear fit, improves as $\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}$ decreases, however the percentage of measurements which reach a declared steady state within the prescribed sample window (~ 6.5 min) falls with corresponding decreases in the $\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}$ parameter. Response time, defined as the time required to reach declared steady emf, is also a strong function of parameterization. Dependence of response times on sample composition and/or ISE membrane composition and type are also discussed; results for ISEs in samples comprised exclusively of interfering ions are included. In general, limiting emf derivatives of $\{0.25\text{--}0.4\text{ mV} \cdot \text{min}^{-1}\}$ and stability windows of $\{30\text{--}40\text{ sec}\}$ achieve both good analytical accuracy and compliance with potentially short sampling window requirements. Methodology based on use of these parameters can improve sampling speed and accuracy as well as promote inter-comparison of data and ISE characterizations

among research teams.

Keywords: Ion selective electrodes, equilibration detection, automation

4.1 Introduction

Ion selective electrodes (ISEs) now exist for measurement of a great number of both cations and anions. Their simplicity, small size, and low power requirements make them particularly attractive for a wide range of applications where real-time and/or in-situ analysis is beneficial. The electromotive force (emf), or electrical potential, produced by an ISE varies with the logarithm of ionic activity over a substantial portion of the ISE’s usable range and is typically described by the Nernst equation equilibrium model. However, because several tens of seconds, or even minutes, may elapse before this emf reaches a sufficiently steady value, any measurement by ISE necessarily involves determining the time at which this sufficiently steady (sometimes designated “equilibrium”) electrode emf is declared. This can be problematic, given that ISE response characteristics vary by electrode and membrane type, manufacturer, history of usage and storage, and aqueous solution composition and ionic strength. In addition, solution properties and accuracy requirements vary widely between applications in medicine, industry, and environmental chemistry. As a result, ISE measurement technique is often customized by field, practitioner, or application.

The steady or “equilibrium” emf value is often subjectively judged by the analyst, values being detected by eye as signals “level off” or “look flat” on a graph. While such a protocol can be entirely adequate under the eye of an experienced analyst in applications within a constrained range of concentrations and at nearly constant ionic strengths, other applications within industrial and environmental chemistry may face wide ranges of analyte concentration or highly varied (and sometimes very weak) ionic strengths and/or cases where sample pre-processing may not be an option, e.g., use of sensors for in-situ sampling. In these cases, electrode emf can approach stable values quite slowly and/or have a non-monotonic approach to steady state [149, 4], making subjective or ad hoc means of declaring ISE “equilibrium” particularly problematic. Further, use of automation, which can be highly beneficial or even mandatory in the case of measurement from an autonomous mobile platform, requires an explicit algorithm for “equilibrium” emf detection. More broadly, lack of a uniform protocol likely **undermines reproducibility and weakens comparability of data across differing dates, times, sample solutions, or laboratories.**

A corollary to the lack of a uniform protocol for determination of “equilibrium” emf is the lack of a standardized criterion for determining electrode response time, a key metric used to characterize ISEs, which limits the extent to which the published response times of competing electrode technologies is meaningful. Various guidelines for determination of ISE response time have been proposed and supported by the IUPAC in the past. Early recommendations included use of t_{90} or t_{95} (time at which emf reaches 90% or 95% of final stable emf) and t^* (time at which emf reaches final stable value within ± 1 mV) [168]. Three drawbacks of these methods have been previously identified [4]:

1. Both methods require knowledge of the ‘final stable emf,’ which by definition is not known during live data readings, making automation difficult to impossible;
2. The log-linear relation of concentration and voltage dictates that a 1 mV change in emf corresponds to an approximately 4% change in concentration for singly-charged ions (more for doubly-charged ions), which may not be acceptable in all applications;

3. Somewhat counter-intuitively, these definitions necessitate that the response time is not the time at which a stable emf is reached/recorded.

In 1994 the IUPAC updated the recommended method for determination of ISE response time [25], proposing a method based on the slope of emf/time ($\Delta E/\Delta t$), specifically the “time which elapses between immersion in a sample and the first instant at which the emf slope reaches a pre-determined limiting value.” In addition to providing a more functional methodology (in the sense that it can be automated), this has the benefit of creating a methodology within which the response time is by definition the time at which the steady state emf is reached. While this steady state emf is not necessarily identical to the value for a true chemical equilibrium in the Nernstian sense, $\Delta E/\Delta t$ would in practice be chosen such that the emf would be sufficiently stable and repeatable for most analytical purposes. Use of such a method is also likely to avoid very long equilibration times, which are both inconvenient in a practical sense and may increase the exposure of the methodology to low frequency noise or signal drift.

Adoption of this method has not been widespread, however, as noted by other researchers in the field [169]. This may be due to (1) interest in presenting statistics which are easily compared with historical publications and work, or (2) ambiguity in defining an appropriate, standardized $\Delta E/\Delta t$ value, as the IUPAC recommendations simply indicate selection “on the basis of the experimental conditions and/or requirements concerning the accuracy (e.g., $0.6 \text{ mV} \cdot \text{min}^{-1}$.)” [25].

Further, this $\Delta E/\Delta t$ method may not be robust or determinate when faced with real-world signals that inherently contain some amount of noise. Ion selective electrodes in particular may be vulnerable to noise due to (1) their high output impedance and consequent sensitivity to electromagnetic noise, (2) potential drift in output signals, e.g., due to leaching, and (3) their sensitivity to temperature changes. Because even small amounts of noise may lead to non-idealized (i.e., non-monotonic) $\Delta E/\Delta t$ time series, this issue must be considered in the development of an automated implementation of these IUPAC recommendations.

We address these problems by systematically examining the approach to steady state emf of a representative suite of ISEs over a large range of solution concentrations and compositions, with the goal of characterizing a reliable, reproducible, and fully automatable detection methodology. The specific choice of ISEs, solution compositions, and ionic strength range in this work is guided by the objective of making in-situ, real-time measurement of the major ion composition of natural waters; however the approach to assessing ISE protocols for their means of declaring steady emf is independent of application. Our method is consistent with the 1994 IUPAC recommendations but also provides for automated detection of a steady state emf and response time from a dynamic ISE time series *in real time*. Optimization with respect to response time (minimization), steady state emf error (minimization), and method robustness (e.g., sensitivity of results to a small change in parameterization) is detailed. Cases where signals do not follow the traditionally-expected exponential-like approach to “equilibrium,” i.e., are not monotonic, are considered, along with restrictions that may prevent steady state conditions from being detected within an acceptable sampling window or at all. Typical response times recorded when using this method, along with the parameters that most strongly affect the response time are presented. Finally calibration curves for electrodes considered in this work are shown, along with discussion of the sensitivity of RMSE error to parameterization choice, demonstrating that this method leads to robust calibration curves, with near-Nernstian slope, high R^2

value, and linear response down to μM levels.

4.2 Materials and Methods

4.2.1 Theory

Development of a practical algorithm for determining the emf at which an ISE could be considered sufficiently steady was based on the following assumptions:

1. The time at which steady state is reached can be related to a specific value of $\Delta E/\Delta t$ (here termed $\Delta E/\Delta t_{\text{limit}}$), following IUPAC recommendations;
2. Once meeting the criterion of (1), recording the emf value at the soonest possible time point is the most accurate (as longer immersion may lead to drift or signal change due, e.g., to leaching or slow surface processes), also following IUPAC recommendations;
3. Automatically-detected steady state emf should be consistent with values declared by an experienced analyst inspecting the same data.

Additional challenges posed by real (non-idealized) data which are not addressed by IUPAC recommendations include the following:

4. Noise in data often leads to non-monotonic $\Delta E/\Delta t$ signals;
5. ISE response itself may be in the form of a non-monotonic time-series signal (see Fig. 4-1), e.g., for measurements in low-concentration standards where slow surface reactions (order minutes) can cause the standard electrode potential and selectivity coefficients to change over time [4], where surface processes changing the chemical composition and/or morphology of the electrode membrane surface can change the surface potential over time [4], or before the liquid junction potential has stabilized [68, 66].

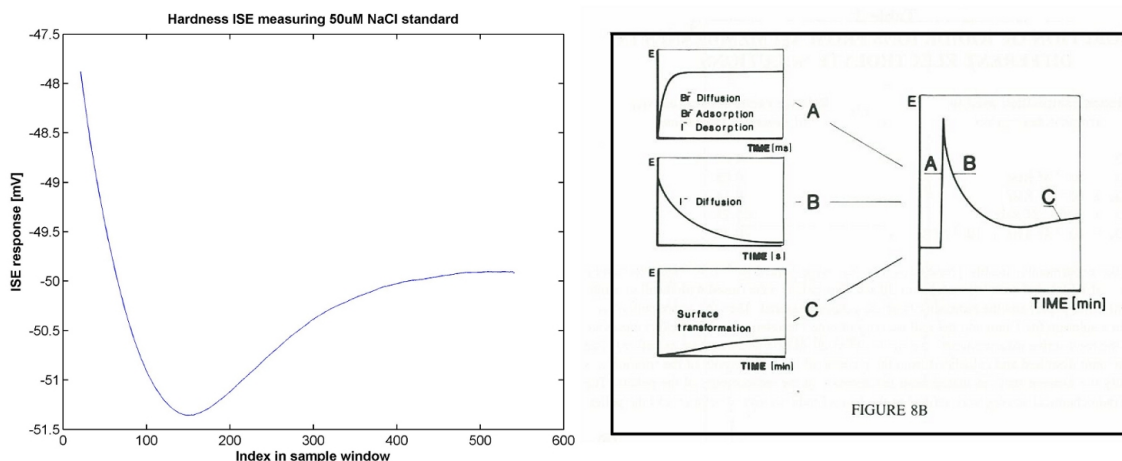


Figure 4-1: Typical non-monotonic ISE response signal as seen in this study (left) and as elucidated by Lindner et al. [4] (right).

These latter issues are worthy of consideration as data with non-monotonic characteristics can lead to erroneous identification of steady state conditions by automated systems,

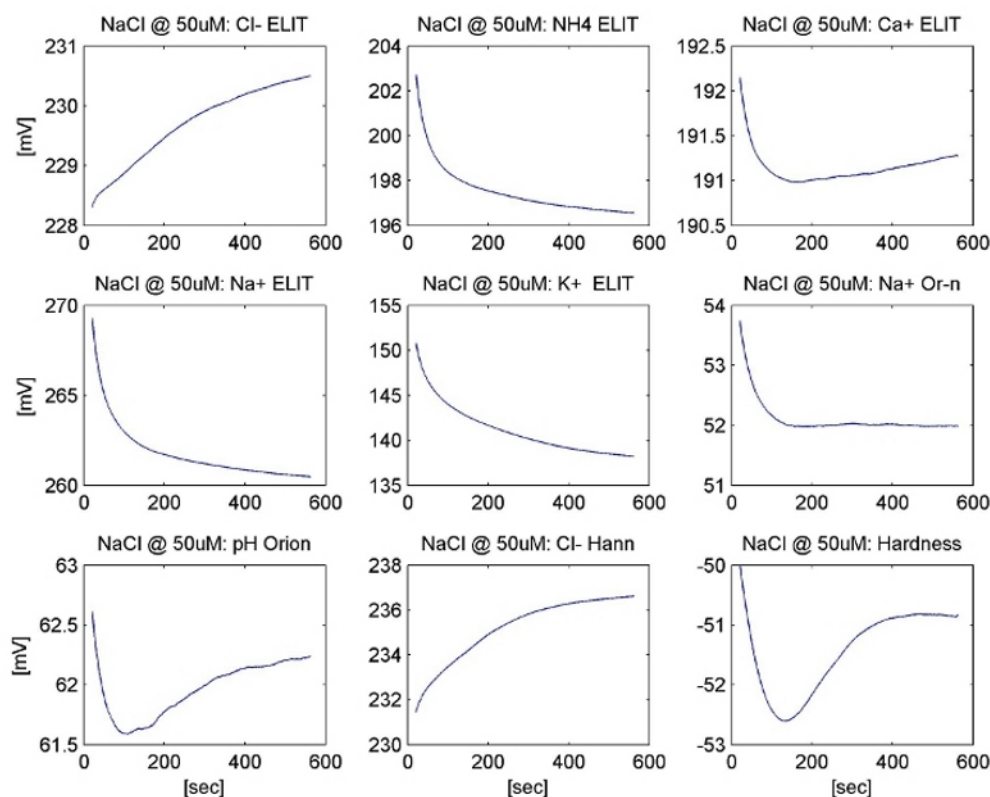


Figure 4-2: Range of time-series responses of ion selective electrodes to a single aqueous sample.

primarily due to transient minima in $\Delta E/\Delta t$ signals or irregularities in overall $\Delta E/\Delta t$ signal shape. Signal smoothing (e.g., running average or boxcar averaging) typically mitigates small transients caused by relatively high-frequency noise, however, not all signals will produce a monotonic derivative signal even after smoothing. For instance, conditions such as those shown in Fig. 4-1 require identification of a steady state emf at a non-limiting point in the curve (the dip between “B” and “C” as shown on the right) to avoid bias caused by slow surface processes [4]. There exist many such resulting signal shapes; Fig. 4-2 presents the an example range of time-series responses recorded during this sampling campaign.

To address these issues, our automated method incorporates a $\Delta E/\Delta t$ *stability condition* to qualify the IUPAC recommendation of defining “equilibrium” on the basis of the first instance at which a limiting $\Delta E/\Delta t$ is reached. Specifically this stability conditions is defined by a **stability window**, being met at the first window of time of duration \mathbf{win}_{st} during which the absolute value of the emf slope remains less than the designated limiting value, $\Delta E/\Delta t_{limit}$. Declared steady emf is subsequently determined as the average emf value within the designated stability window.

4.2.2 Sensitivity Analysis

The sensitivity of resulting *steady state emf* and *response time* values to the choice of parameterization was examined for values of $\Delta E/\Delta t_{limit}$ and \mathbf{win}_{st} consistent with those

seen in published research [4, 169] and with the level of signal stability expected from ISEs. Specifically, all combinations of parameters within the following ranges were considered:

$$\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}} = \{0.1, 0.2, 0.4, 0.8, 1.0\} \text{ [mV} \cdot \text{min}^{-1}\text{]}$$

$$\mathbf{win}_{\text{st}} = \{10, 20, 30, 40, 50\} \text{ [sec]}$$

The mid-range parameter set $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 30 \text{ sec}\}$ was chosen as a convenient baseline for viewing trends over parameterization variation, presented below. For all cases, recall that the declared steady emf is determined as the average emf value within the designated stability window.

Resulting steady state values were subsequently used to produce linear calibration curves over as much of the concentration range as possible, i.e., with the linear range determined by a maximization of R^2 for the linear fit. Slope, slope margin of error, intercept, intercept margin of error, R^2 , and RMSE for the linear fits were compared for the range of parameterizations. RMSE values were judged to be the most informative, producing the largest spread among varying parameter sets. RMSE is thus used below to quantitatively differentiate calibration curves produced using different parameterizations.

Uncertainty caused by variation within the stability window must also be considered. Assuming the maximum allowed rate of change in emf ($\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}$) continues over the width of the stability window, Eq. 4.1 shows the relationship of the maximum emf change over the stability window (dV_{max}) to parameterization choice.

$$dV_{\text{max}}[\text{mV}] = \text{win}_{\text{st}}[\text{sec}] * \Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}}[\text{mV}/\text{min}] * \frac{1}{60} \quad (4.1)$$

Conditions where less than 1% uncertainty is introduced by this variation correspond to the:

$$dV_{\text{max}} < 0.25\text{mV} \text{ (singly-charged ions)} \quad (4.2)$$

$$dV_{\text{max}} < 0.13\text{mV} \text{ (doubly-charged ions)} \quad (4.3)$$

where the baseline case $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 30 \text{ sec}\}$ has $dV_{\text{max}} < 0.2 \text{ mV}$. Note that these limits are surpassed in some parameterizations considered, e.g., those where $\Delta\mathbf{E}/\Delta\mathbf{t}_{\text{limit}} > 0.5 \text{ mV} \cdot \text{min}^{-1}$.

Fig. 4-3 shows the results of the tighter sensitivity analysis $\{0.3\text{--}0.5 \text{ mV} \cdot \text{min}^{-1}, 20\text{--}40\text{sec}\}$ omitted from the primary text. The maximum difference in average declared concentration values for this analysis was less than 0.7%. In addition, one sees a trend in mean changes symmetrically around the baseline case, indicating that this baseline is likely to be an acceptable representation of parameterizations in this range.

4.2.3 Experimental Setup

Materials

Salt solutions used for electrode characterization are listed in Table 4.1, along with the concentration levels of the 13 standards which each contained the indicated concentration of a *single salt only*, producing a total of 52 different salt standards. Concentrations from $0.1 \mu\text{M}$ to 0.1 M were chosen to simulate the widest possible range of analytes expected in environmental fresh water applications while salt solutions were chosen to correspond to primary specificity and to expected or known interferences of the electrodes. Consequently, for many salt/electrode combinations the electrode response was determined solely by *interfering* ions to which the response was substantially sub-Nernstian (or, perhaps, not well defined, as in the case of divalent ISE responses to univalent ions).

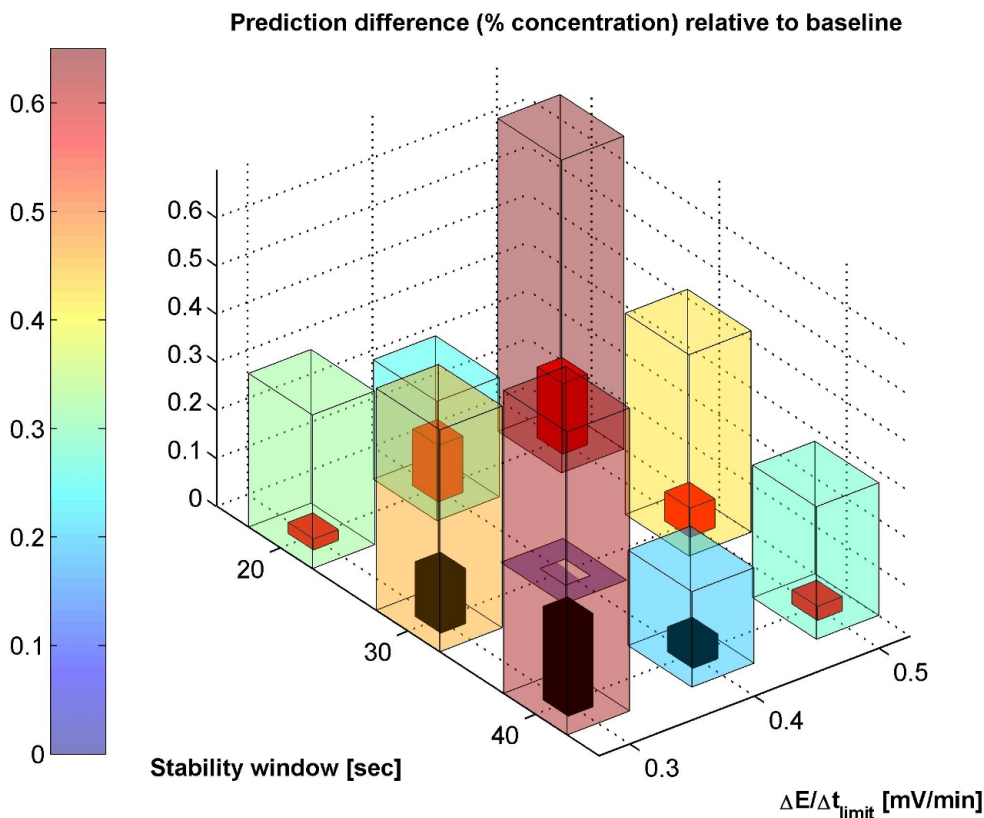


Figure 4-3: Mean percent change in determined steady state concentration for the tighter range of parameterizations relative to the baseline case of $\{ 0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec} \}$. Interior solid bars show the mean change (black: > 0 ; red: < 0) while exterior transparent bars show the mean absolute value of the change. Note that parameterization difference within this range can result in no more than a 0.6% change in declared concentration.

Standards were made using Millipore Milli-Q water ($18.2 \text{ M}\Omega\text{-cm}$) and Reagent A.C.S. grade salts (NaCl and NH_4Cl , Fisher Scientific; KCl , MCB Reagents; $\text{CaCl}_2 \cdot 2\text{H}_2\text{O}$, Mallinckrodt). Salts were dried overnight at 55°C before being weighed out using an Ohaus Precision Standard TS4KD balance. Standards were made via serial dilution from 100 mM (for $1.0 \mu\text{M}$ to 10 mM) or $100 \mu\text{M}$ (for 0.1 to $0.5 \mu\text{M}$) in Class A Pyrex volumetric flasks. Glass and plasticware used in this process were first acid washed for at least 24-hours in $10\% \text{ HNO}_3$ and rinsed 7-10 times in Milli-Q water.

Instrumentation

Nine ion selective electrodes (6 solid-state, 1 plastic, 2 glass membrane) were characterized in the single-salt solutions described above. Electrodes were selected both to maximize relevance to environmental applications, measuring ions typically found in surface and ground waters, and to provide some general insight into behavior of different ISE technologies. All electrodes were configured as half-cells and simultaneously referenced to a single double-junction reference electrode. A list of ISEs used is given in Table 4.2.

	Concentration
	0.10 μM
	0.25 μM
	0.50 μM
Salt	1.0 μM
<i>NaCl</i>	2.5 μM
<i>KCl</i>	5.0 μM
<i>NH₄Cl</i>	10 μM
<i>CaCl₂</i>	25 μM
	50 μM
	100 μM
	1 mM
	10 mM
	100 mM

Table 4.1: Single-salt standards used for electrode characterization, producing a total of 52 standard salt solutions.

Procedures

Time course data were collected simultaneously from all 9 electrodes for each standard via a custom LabVIEW interface (LabVIEW 2009), receiving input from a National Instruments USB-6218 Data Acquisition device. A custom-built low-pass filter stage (active 2-pole Butterworth filter, cutoff frequency of 10Hz) was installed between electrode outputs and data acquisition inputs to reduce high-frequency noise coupled into the system via high-impedance electrodes. Note that results should not be dependent on the specific filter or analog-to-digital hardware used as long as it is properly implemented, i.e., where ISE leads are connected via a high input impedance interface, internal circuitry, filters and amplifiers have low leakage currents, etc.

During data collection, standards were measured from low to high concentration, with seven replicates of each standard measured sequentially for each concentration; the electrode circuit was broken by removal from the solution into air between measurements. Samples were measured *unstirred* to minimize electrical signal interference, to record complex membrane responses as the boundary layer develops, and to simulate conditions expected during in-situ sampling. Because samples were not expected to be heterogeneous (and thus requiring stirring to achieve an accurate equilibrium measurement), measurement in still (unstirred) samples also avoided measurement noise due to small spurious fluid and/or electrical currents caused by fluid (and dissolved ion) movement in the sample dish.

A complete set of data for each single salt solution (13 standards \times 7 replicates) was recorded in a single day to minimize potential environmental or electrode drift effects. Each sample time-course was recorded for approximately 6.5 minutes at 1.3Hz, yielding time sequences of 500–600 samples. This produced a total of 3276 (4 salts \times 13 standards \times 7 replicates \times 9 ISEs) full time course measurements (each of 500–600 instantaneous samples) recording ISE approach to steady state.

Measurements of approximately 75mL of sample were recorded under unstirred batch conditions in a temperature controlled setting (less than 2°C range) to minimize electromagnetic noise and to reproduce expected environmental in-situ sampling conditions. Note that while the magnitude of response times under unstirred conditions is likely to be greater than

Analyte	Membrane	Manufacturer	Published LOD
Ca^{2+}	Solid-state polymer matrix	PVC ELIT 8041	$0.50 \mu M$ (0.02 ppm)
K^+	Solid-state polymer matrix	PVC ELIT 8031	$10 \mu M$ (0.4 ppm)
Na^+	Solid-state polymer matrix	PVC ELIT 8230	$2.0 \mu M$ (0.05 ppm)
NH_4^+	Solid-state polymer matrix	PVC ELIT 8051	$2.0 \mu M$ (0.03 ppm)
Cl^-	Solid-state crystalline	poly- ELIT 8261	$30 \mu M$ (1 ppm)
Cl^-	Solid-state (unspeci- fied)	Hanna Instruments 4007	$50 \mu M$ (1.8 ppm)
Hardness (di- valent cation)	Plastic (unspecified)	Thermo 9332BNWP	Sci. $6.0 \mu M$
pH	Glass	Thermo Sci. 9101BN	pH 0 - 14
Na^+	Glass	Ross 8411BN	$1 \mu M$ (0.02 ppm)
Reference	double junction CH_3COOLi	ELIT 003N	

Table 4.2: Ion selective electrode hardware; information on membrane composition and published detection limit (LOD) as given by manufacturers where available.

under stirred conditions [4], it is expected that the proposed methodology and demonstrated trends will pertain equally to other experimental conditions.

4.3 Results and Discussion

4.3.1 Steady state determination

Figure 4-4 shows the mean absolute difference (mV) in declared steady state emf for all parameterizations relative to the baseline case ($0.4 \text{ mV} \cdot \text{min}^{-1}$, 30 sec). These data are averages over time series for 9 electrodes, 4 salts, 13 standards, and 7 replicates ($N > 3200$ samples). For reference, a second set of axis values is shown on the colorbar to indicate the corresponding percent change in concentration values for a monovalent Nernstian relationship. While not all electrode responses are expected to be monovalent or Nernstian, e.g., responses to interfering analytes, this provides an indication of the *minimum corresponding concentration change* expected for a given voltage change. These data clearly demonstrate that **within this range, sensitivity to parameterization choice itself may introduce greater than 2% difference in declared concentration for this set of electrodes.**

Manual review of the results indicated that the highest and the lowest $\Delta E / \Delta t_{\text{limit}}$ parameterizations (0.1 , 0.8 , and $1.0 \text{ mV} \cdot \text{min}^{-1}$) frequently selected a steady state emf that differed significantly from that chosen visually by an experienced analyst; the same was true for the shortest stability windows (e.g., 10 sec). Consequently, a second, more tightly clustered, sensitivity analysis ($0.3\text{--}0.5 \text{ mV} \cdot \text{min}^{-1}$, 20–40 sec) was undertaken to investigate sensitivity to parameterization in the range where manual interpretation of the

plots suggested that the results would likely be consistent with an analyst’s judgment. Maximum difference in average declared concentration values for these parameterizations was less than 0.7%., indicating that sensitivity to parameterization choice in this range is minimal (data not shown).

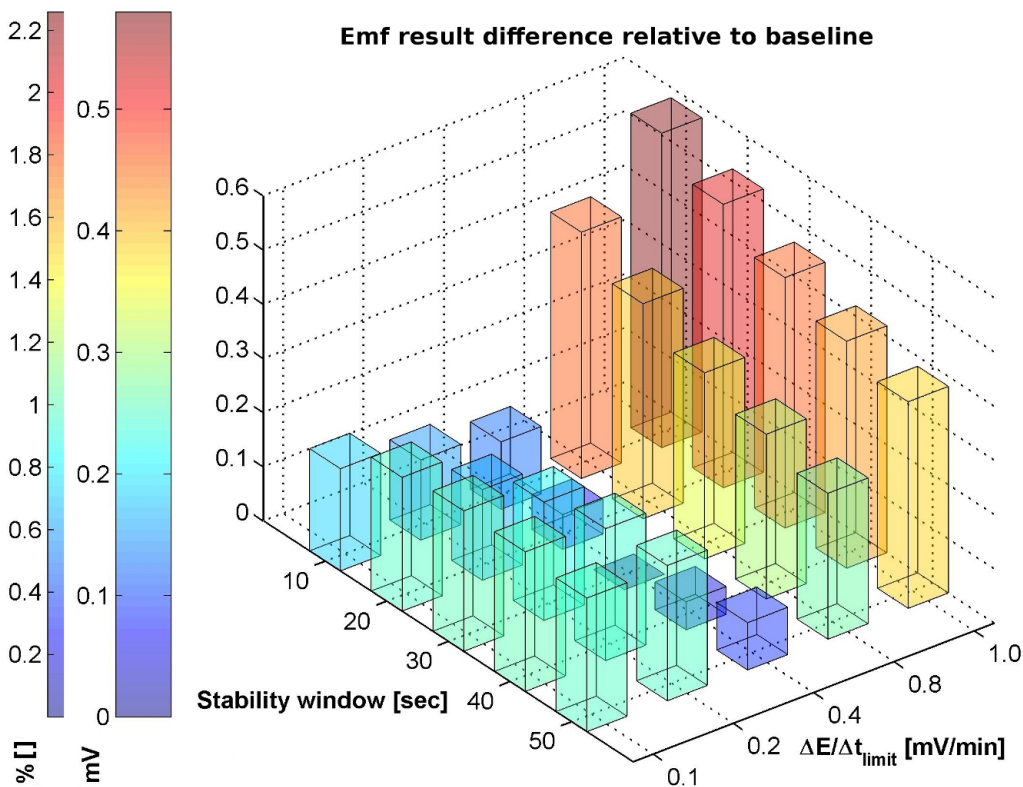


Figure 4-4: Mean absolute change in determined steady state emf [mV] for a range of parameterizations relative to the baseline case of $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 30 \text{ sec}\}$. Difference in bar heights indicates that emf declared for a specific time series may vary significantly with parameter choice.

4.3.2 Linearity of electrode response

Choice of appropriate parameterization must also result in acceptable calibration curves demonstrating reproducibility and ideally a high degree of linearity over a usable range of concentrations. The response of most modern commercial ISEs to primary ions is expected to be approximately Nernstian over a substantial range of ionic concentration. Shown in Figure 4-5 are calibration curves for four solid state electrodes (ELIT Na^+ , K^+ , NH_4^+ , Ca^{+2}) each in their corresponding salt solution (e.g., Na^+ ISE measuring NaCl solution); note that measurements were taken at approximately 19 °C, at which the expected Nernstian slope is 57.9. While the exact membrane composition of the sodium ISE is proprietary, similarly sub-Nernstian responses have been recorded with electropolymerized ionophores and phthalocyanine-based electrodes in the past [170] and remain poorly explained in the literature. When using parameter values of $0.4 \text{ mV} \cdot \text{min}^{-1}$, 30 sec, all four electrodes

produce linear responses ($R^2 > 0.99$) with near Nernstian slopes (Na^+ : 54.8 ± 0.6 ; K^+ : 58.0 ± 0.4 ; Ca^{+2} : 29.3 ± 0.7 ; NH_4^+ : 58.1 ± 0.6) down to at least the $1 \mu\text{M}$ level (shown with 3σ error bars). Activity corrections do not change these values for singly-charged ion solutions; for calcium-containing solutions, however, inclusion of the activity correction (Debye-Huckel) increases the slope to 31.4 ± 0.3 .

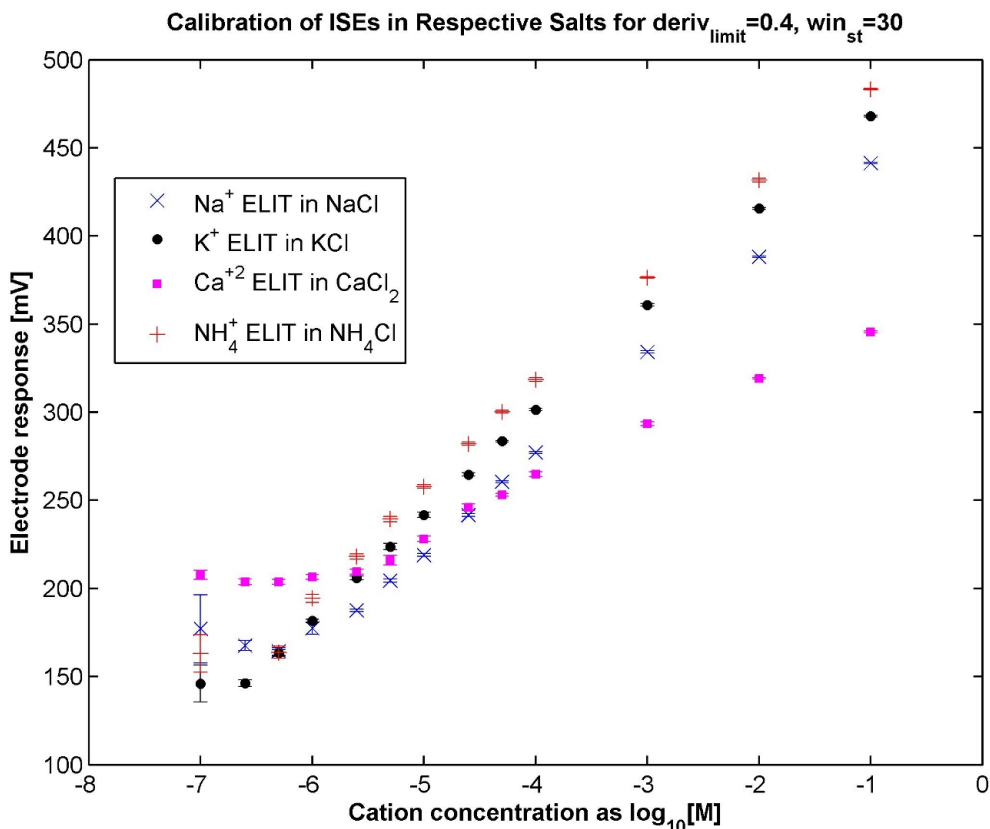


Figure 4-5: Calibration curves for four ELIT ISEs in their respective salts (3σ error bars). Linear fits with near-Nernstian slopes (Nernstian slope at measurement temperature of 19°C is 57.9) and $R^2 > 0.99$ are found for concentrations down to $1\mu\text{M}$ in all cases (down to $0.25\mu\text{M}$ for K^+).

Many other metrics of goodness, including slope, slope margin of error, intercept, intercept margin of error, R^2 , and RMSE were also inspected to compare calibration curves resulting from differing parameterizations. Of these, RMSE was identified as the most differentiating indicator for quality of fit of ISE data to the Nernstian ideal. Importantly, however, this introduces a necessary tradeoff between (1) producing the most accurate data (which typically requires using the most stringent limitations) and (2) producing usable results in a practical time frame (typically requires relaxing requirements to maximize number of results). The nature of this trade-off and resulting findings are discussed further below.

4.3.3 Rate of failure to declare equilibrium as a function of parameterization

In practice it may be necessary to choose a parameterization that will lead to a declared steady state emf within a prescribed maximum time. For example, in the case of data gathered by a moving underwater vehicle, sample time directly translates into spatial resolution of the chemical mapping, and longer sample periods result in lower spatial resolution. Maximum sampling time for many applications could thus be less than 10, or even 5, minutes; for the purpose of the present study, a sample period of approximately 6.5 minutes was used. Results were then analyzed to determine the number of samples which did not reach a declarable steady state in this sample period, expressed as the **equilibrium failure rate**, λ_{failure} . This rate was then examined as a function of parameterization; more ‘stringent’ values were expected to increase the failure rate.

Figure 4-6 shows the observed λ_{failure} (as a fraction of ~ 3200 samples) as a function of solution composition and probe type. From this figure, it is clear that the fraction of samples which do not reach steady state is a strong function of $\Delta E/\Delta t_{\text{limit}}$, with as few as 50% of the samples reaching a declared steady state for parameterizations where $\Delta E/\Delta t_{\text{limit}} \leq 0.2 \text{ mV} \cdot \text{min}^{-1}$. This favors adoption of the maximum $\Delta E/\Delta t_{\text{limit}}$ value that is consistent with acceptable analytical accuracy for applications where sample time is a constraint.

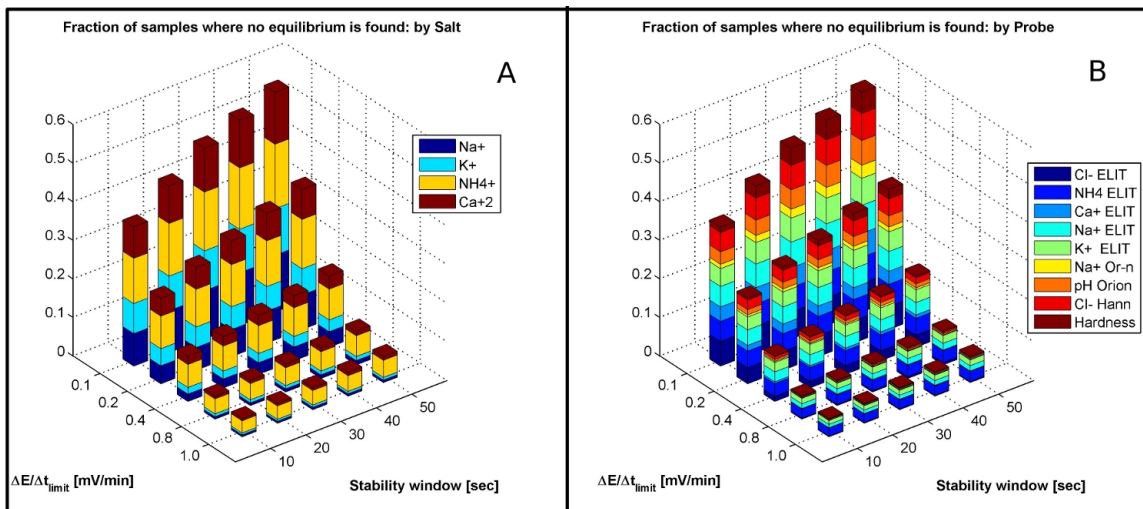


Figure 4-6: Effect of parameterization on equilibrium failure rate, λ_{failure} , for a sample period of ~ 6.5 minutes. Bars are subdivided by solution content (A) and probe (B) to demonstrate the range of characteristics affecting the response time of electrodes.

4.3.4 Optimization against system constraints

In order to determine an optimal choice of parameterization, it is necessary to consider both analytical accuracy and λ_{failure} minimization. To do so, RMSE (against the optimal linear fit) was plotted against the *equilibrium success rate* ($\lambda_{\text{success}} = 1 - \lambda_{\text{failure}}$) for each of 45 parameterizations, including those listed in Section 4.2.1 and several additional values for $\Delta E/\Delta t_{\text{limit}}$ in the ‘most promising’ mid-range identified above. The resulting 36 plots (one for each combination of salt solution and ISE probe) were examined for visible trends or trade-offs in accuracy vs. equilibration success rate. Although overall plot shape showed

significant variability, three features were generally identifiable:

1. Existence of an RMSE baseline for parameterizations: Below some cutoff, further lowering of $\Delta E/\Delta t_{\text{limit}}$ results in a drop in λ_{success} without a corresponding drop in RMSE;
2. An approximately linear trade-off region: Increasing $\Delta E/\Delta t_{\text{limit}}$ produces an increase in λ_{success} that is coupled with an increase in RMSE;
3. An equilibrium success rate saturation region: Further increase of $\Delta E/\Delta t_{\text{limit}}$ produces an increase in RMSE without a corresponding increase in the λ_{success} .

In many cases, $\Delta E/\Delta t_{\text{limit}} < 0.2 \text{ mV} \cdot \text{min}^{-1}$ corresponded to condition 1 while $\Delta E/\Delta t_{\text{limit}} > 0.4 \text{ mV} \cdot \text{min}^{-1}$ corresponded to condition 3. Exact curve shape was unique for each salt solution / probe combination, however, with some demonstrating clear parametric optima and others showing parameterizations which would not be desirable even under conditions where an increase in RMSE could be tolerated. Generally, these trends reinforced the conclusions stated above regarding unacceptable parameterization values.

Results for $\Delta E/\Delta t_{\text{limit}} = \{0.15\text{--}0.4\} \text{ mV} \cdot \text{min}^{-1}$ are qualitatively summarized and shown in Figure 4-7. Note that (*) indicates *acceptable results* (not necessarily optimal results) *for all salt/probe combinations in the present study* while hashing indicates poor results. Results were further evaluated to identify acceptable parameterizations separately for selectivity matched (M) salt/probe pairs (e.g., NaCl as measured by the Na⁺ ELIT ISE) and un-matched (U) pairs (e.g., CaCl₂ as measured by the Na⁺ ELIT ISE).

		win_{st} [sec]				
		10	20	30	40	50
$\Delta E/\Delta t_{\text{limit}}$ [mV/min]	0.15					
	0.2				U	
	0.25		U	*	U	U
	0.3		U	*	*	M
	0.35			U	*	M
	0.4			U	*	M

Figure 4-7: Summary of parameterization ‘goodness’ as judged by simultaneous minimization of RMSE and maximization of equilibrium success rate. Cross-hatching = poor results; * = good results for all probes; M/U = good results for the subset of selectivity matched (M) or un-matched (U) salt/ISE probe pairs.

Overall, parameterizations of $\Delta E/\Delta t_{\text{limit}} = \{0.25\text{--}0.4 \text{ mV} \cdot \text{min}^{-1}\}$ and $\text{win}_{\text{st}} = \{30\text{--}40 \text{ sec}\}$ produced the best results over the large range of solution compositions, concentrations, and ISE technologies that were studied. When jointly optimizing for analytical results and equilibration success rate, the parameter set of $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 40 \text{ sec}\}$ is thus determined to be optimal for the electrode set tested.

4.3.5 Quantification of response time

The effect of parameterization was also investigated for its effect on the response times assigned to each electrode under the conditions of this study. Figure 4-8 (left) shows the relative difference in determined response time with respect to the $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 30 \text{ sec}\}$ parameterization baseline; additional sensitivity analysis in a tighter parameterization range are shown in Figure 4-8 (right). Note that cool colors indicate parameterizations producing shorter response times than baseline (generally $\Delta E/\Delta t_{\text{limit}} > 0.4 \text{ mV} \cdot \text{min}^{-1}$), while hot colors indicate longer response times (generally $\Delta E/\Delta t_{\text{limit}} < 0.4 \text{ mV} \cdot \text{min}^{-1}$). Dot size is proportional to the magnitude of this difference in all cases.

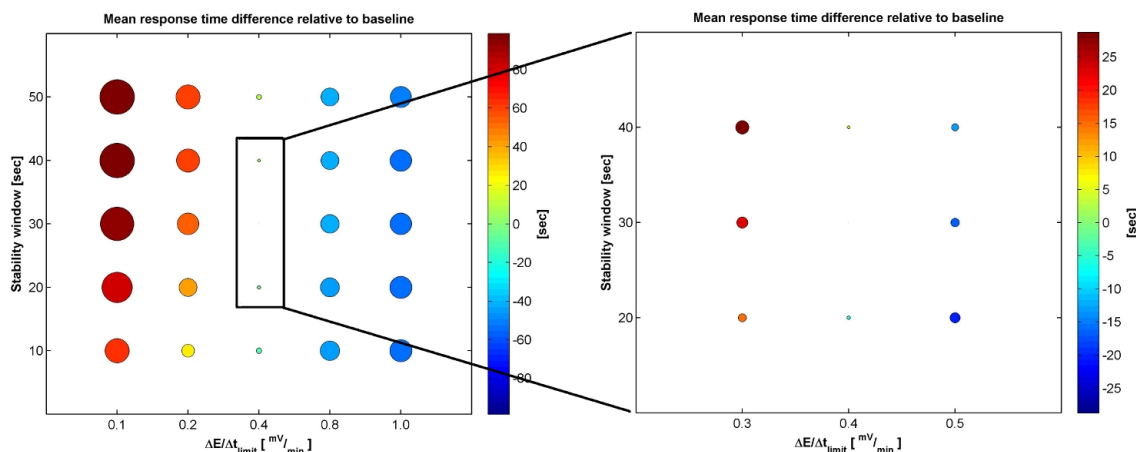


Figure 4-8: Mean difference in determined response time [sec] relative to the $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 30 \text{ sec}\}$ baseline over a range of parameterizations; plot on the right shows results for a more constrained parameterization set (referenced to plot on left). Blue tones indicate that parameterization produces shorter response times than baseline while red tones indicate longer response times (note colorbar scale change from left to right). Note total difference of almost 3 minutes across parameterizations shown in plot on left as compared to a difference of less than 1 minute on the right.

In general, response time determinations were more sensitive to parameterization, and to $\Delta E/\Delta t_{\text{limit}}$ in particular, than were ionic concentrations. Response time varied by nearly 3 min. over the extended parameter set, suggesting that use of a standardized value for $\Delta E/\Delta t_{\text{limit}}$ in comparisons of ISE time responses may be important.

Systematic response time differences among membrane types and identity of ions were also seen (Figure 4-9). Bars outlined in blue in the figure indicate electrodes that are specifically marketed for sensitivity to the given salt cation. Response times for most electrodes to their indicated analyte were in the range of 80–100 sec; response times to other analytes (i.e., interfering ions or those to which the electrode was not strongly responsive) were often more than double that value. Response time also varied as a function of salt solution concentration, changing by a factor of three or more over the concentration range from $0.1 \mu\text{M}$ to 0.1 M (data not shown). In general, the data indicate that a number of characteristics, including specificity of the ISE to the ions in solution, the total concentration of ions in the solution, and possibly the membrane type, strongly contribute to the response time of the electrode, as has been noted previously by other researchers, e.g., [4].

Fig. 4-10 presents an alternate interpretation of average response time for each of the

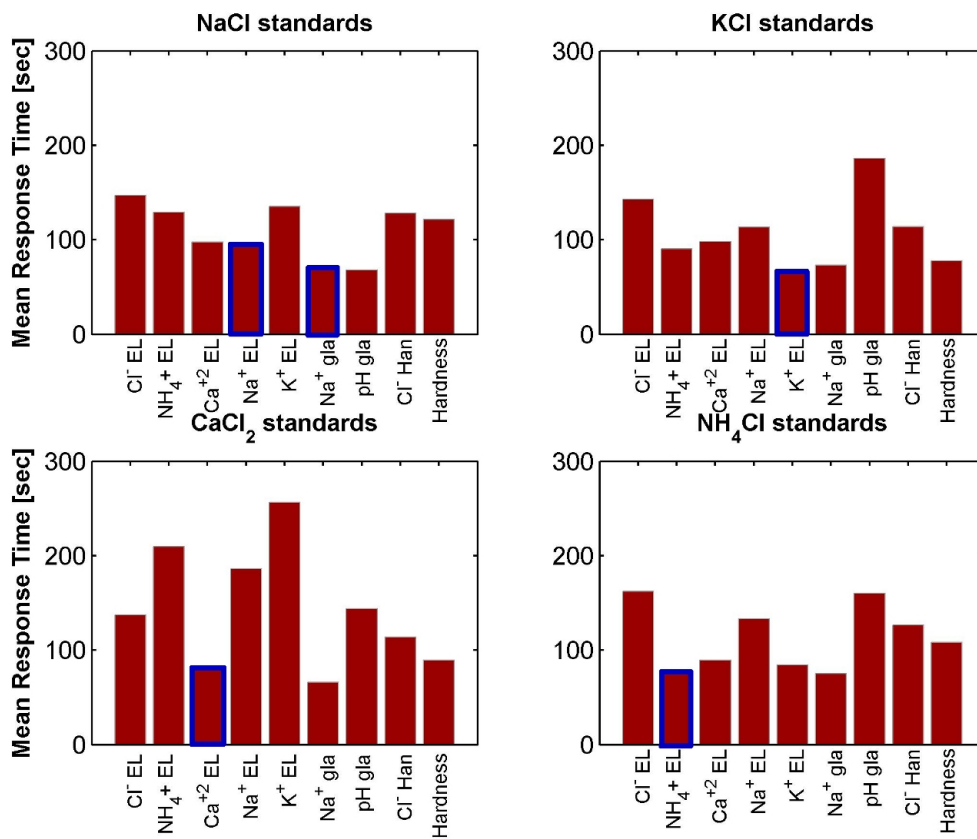


Figure 4-9: Effect of electrode sensitivity and membrane type on mean response time for baseline parameter values.

electrodes relative to different concentration standards. Inspection of the response time as a function of salt solution concentration reveals that response time can change by a factor of three or more over the concentration ranges considered. Lower response times reported for the lowest concentrations may indicate a higher failure rate on channels insensitive to the analyte being measured rather than a truly lower response time. The increase in response time for the highest concentrations of CaCl₂ is not well understood but may have to do with the introduction of substantial interference on the monovalent cation electrodes as the ionic strength of the solution increased.

Finally, Figs. 4-11–4-12 present this data with the effects of probe and concentration separated. It is clear that response time is controlled independently by these two variables, and thus only very high level conclusions can be made. Certain electrodes have predictable behavior (Cl⁻ sensors have slow response at low chloride concentrations, the K⁺ electrode has response orders of magnitude slower when in solutions with no analytes to which it is sensitive), however it seems evident that many of the patterns in electrode response time are controlled by a complex combination of these variables or at least partially by another variable not presented in these figures. This justifies ever more strongly the need for a standardized, signal-based methodology for determining arrival at equilibrium potential: response of a given electrode can vary by orders of magnitude depending on the contents of a give sample, and different electrodes in the sensor suite will arrive at equilibrium potential

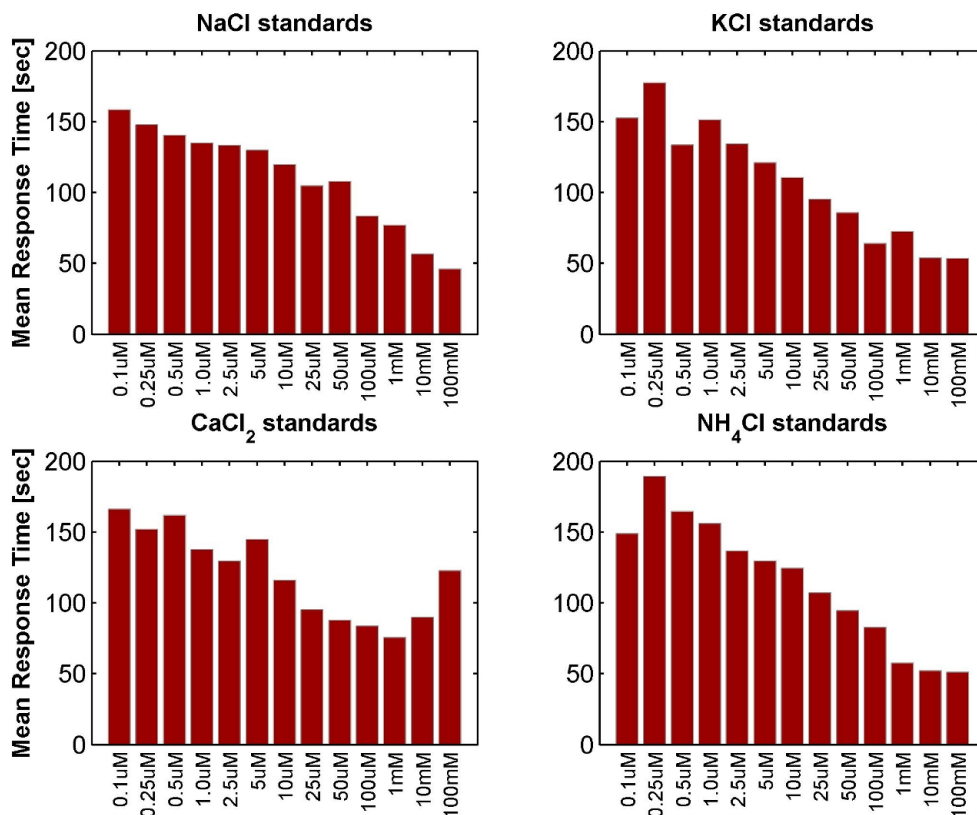


Figure 4-10: Response time averaged over 9 electrodes for different salt solutions at a range of concentrations. Data are taken for parameter values $\{0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec}\}$.

at different times.

These data point to the need for a uniform testing protocol to support intercomparability of published response times for different ISEs. They also have important implications for ISE sampling methodologies where, for example, electrodes are simply allowed to equilibrate for a pre-specified number of minutes before taking a reading; depending on the sample composition and specific ISE membrane type, surface reactions may cause the electrode emf to diverge from an accurate reading before the reading, or the emf may not yet have reached adequate stability.

4.4 Conclusions

Time series emf measurements from a variety of ISEs measuring a wide range of sample compositions and concentrations demonstrate the value of determining steady state using a standardized method employing both $\Delta E/\Delta t_{\text{limit}}$ and \mathbf{win}_{st} criteria. Quantitative trade-offs were also found between (1) obtaining analytically optimal results and (2) obtaining results within a constrained sampling period. In this study, analytical accuracy was preserved for values of $\Delta E/\Delta t_{\text{limit}} = \{0.25\text{-}0.4 \text{ mV}\cdot\text{min}^{-1}\}$ and $\mathbf{win}_{\text{st}} = \{30\text{-}40 \text{ sec}\}$, while at least 90% of samples reach a declared steady state within the sampling time of 6.5 minutes for values of $\Delta E/\Delta t_{\text{limit}} > 0.3 \text{ mV}\cdot\text{min}^{-1}$. Variability in declared emf as a function

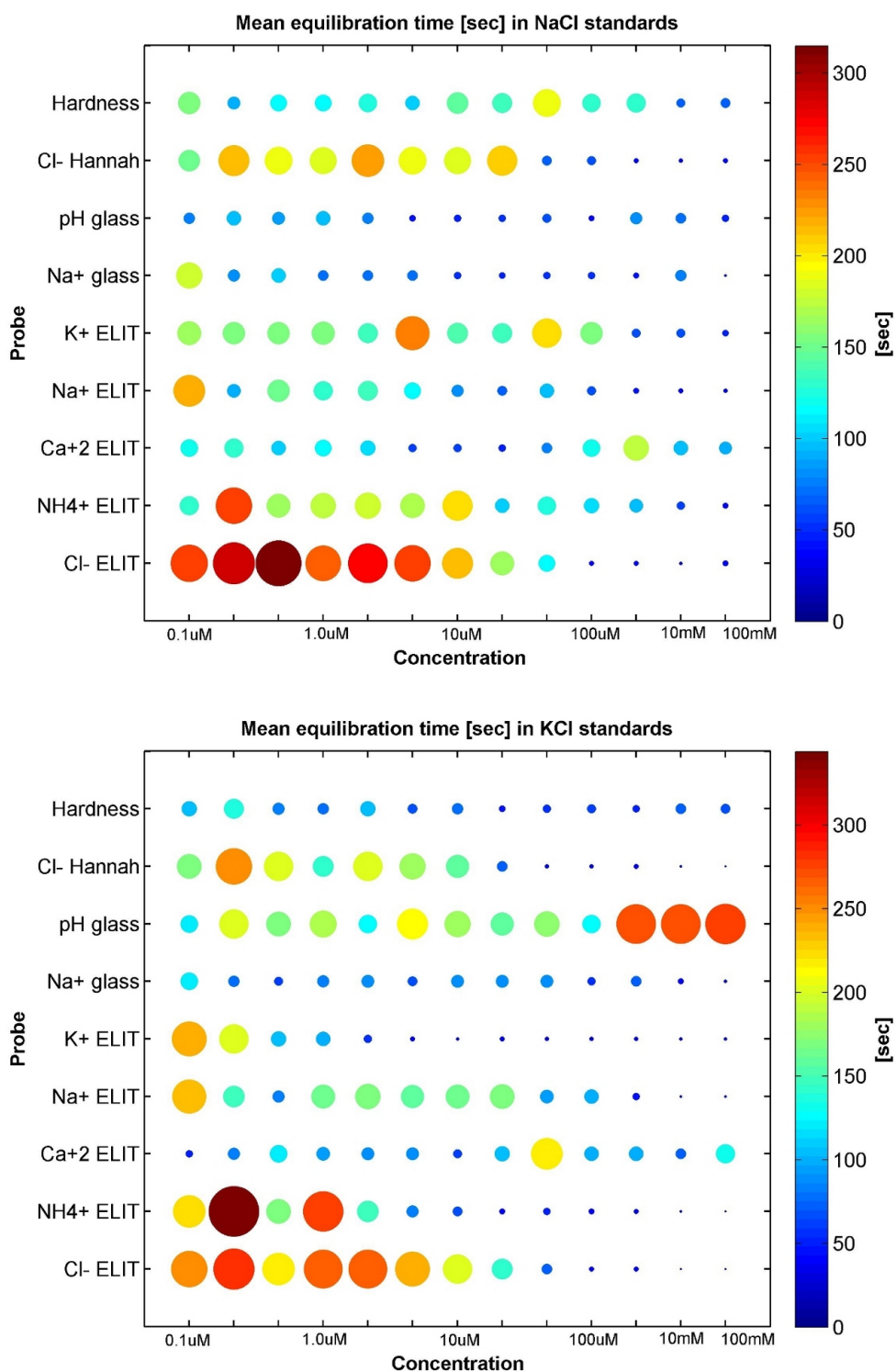


Figure 4-11: Response time of independent electrode channels as a function of NaCl or KCl concentration. Data are taken for parameter values $\{0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec}\}$.

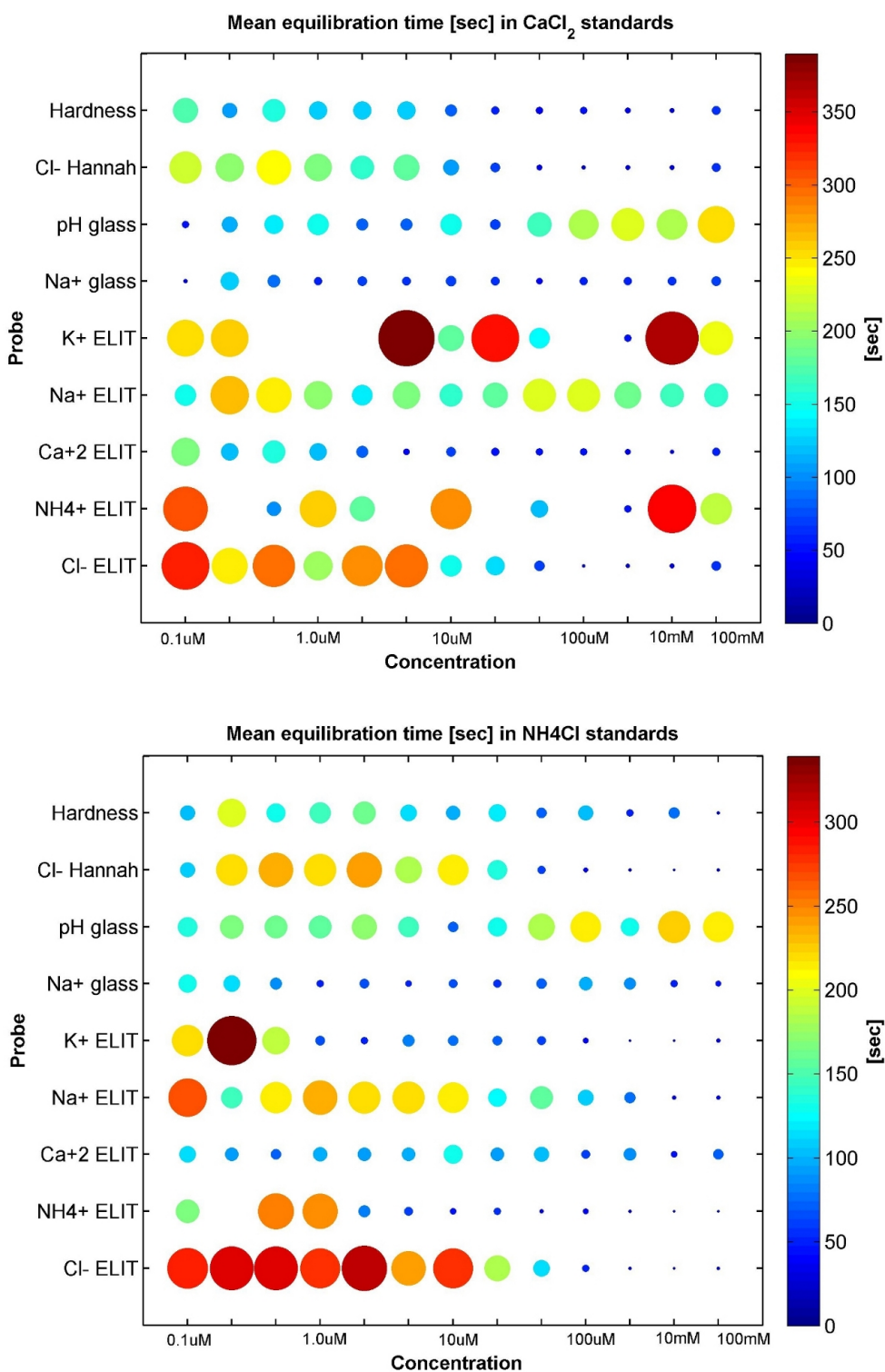


Figure 4-12: Response time of independent electrode channels as a function of CaCl_2 or NH_4Cl concentration. Data are taken for parameter values $\{0.4 \text{ mV}\cdot\text{min}^{-1}, 30 \text{ sec}\}$.

of parameterization choice was not strongly dependent on the choice of $\Delta E/\Delta t_{\text{limit}}$ within this range, while highly linear Nernstian calibration curves are attained, demonstrating the utility of this method for use in analytical applications.

For a given value of $\Delta E/\Delta t_{\text{limit}}$, steady state emf was not strongly dependent on the width of the stability window, \mathbf{win}_{st} , with individual acceptable results being observed for windows as short as 20 sec. Because stability window width strongly influences the calculated response time for the ISEs (longer windows corresponding to longer response times), however, choice of \mathbf{win}_{st} may influence utility of the ISE for some time-critical applications. Analytical results are not shown to greatly improve for $\mathbf{win}_{\text{st}} > 40$ sec but are shown to degrade for $\mathbf{win}_{\text{st}} < \sim 20\text{--}30$ sec; values of $\mathbf{win}_{\text{st}} = \{30\text{--}40 \text{ sec}\}$ are thus generally recommended for the ISEs used in this study. For applications where minimization of response time is essential, however, a parameter set of $\{0.4 \text{ mV} \cdot \text{min}^{-1}, 40 \text{ sec}\}$ is recommended.

Finally, determined **response times** were more sensitive to the choice of method parameters than were steady state emfs, being generally proportional to the value chosen for $\Delta E/\Delta t_{\text{limit}}$. Response time was also strongly dependent on ion/ISE sensitivity and membrane type. This further supports use of a standardized method for ISE characterization to promote reliability and inter-comparability of results, such as the real-time, standardized method described here.

4.5 Acknowledgments

Funding for this work was provided through a GSA Graduate Student Research Grant, an MIT-Xerox Fellowship, a Martin Foundation Sustainability Fellowship, an NSF Graduate Research Fellowship, NSF EAR 0330272, the Singapore-MIT Alliance for Research and Technology, and Sea Grant 6919676.

Chapter 5

ANN Design

Abstract

A primer is provided to introduce standard artificial neural network (ANN) techniques, including terminology and architecture. Application of ANNs, an unconstrained non-linear function estimator, to estimation of chemical concentrations from raw sensor readings is suggested as an alternative to parameterization of traditional models, for which parameters are strongly dependent on both analyte concentration and temperature. Subsequently, a novel architecture is presented, optimized for use with environmental chemical data via incorporation of *a priori* chemical knowledge. Methods are presented for implementation of both charge balance and conductivity constraints, for the base case where the dot product model is used and for extension to log-normalized data where the model can no longer be represented in this manner. While parameterization for these constraints is analyte-dependent, the architecture is not, allowing extension of this method to other environmental problems for which mathematical constraints can be stated. Finally, while training of the neural networks requires substantial time and processor power (~weeks of standard desktop computing), a trained network can process input data extremely rapidly, i.e., $\ll 1$ second, to support real-time sample measurements.

5.1 Introduction

Artificial neural networks are conceptually modeled on a simple understanding of human neural structure, constructed of a topology of interconnected neurons whose “firing” triggers (or fails to trigger) the “firing” of subsequent neurons based on the relative strength of the interconnection. The number of inputs need not match the number of outputs, and because ANNs are typically used to solve over-constrained systems, the number of outputs will usually be smaller than the number of inputs. Initial neuron interconnections are specified by the system designer, however weights of interconnections are adapted during training, and training methodologies do exist which can even “prune” unnecessary interconnections to yield the most efficient neural structure. The number of hidden layers, number of nodes per layer (need not be constant between layers), transfer function between layers (often different for input, hidden, and output layers), learning rate, momentum, and training algorithm must also be specified and will vary between applications; the roles and settings for all of these parameters are discussed in more detail in this chapter. Significant variation among neural networks produced with different tuning parameter values has been shown,

and it is important to note that ANN problems have no closed form solution. Sensitivity to choice of parameter values has been analyzed relative to chemical applications [142, 165, 143, 110], and it is possible to identify parameter values favored in the relevant literature as a starting point for analysis of any new problem [29, 110]. It is, however, typically necessary to explore the space of possible ANNs through trial-and-error to find an optimal system on an application-by-application basis as exhaustive tests have shown resulting weights and biases can be determined more strongly by training parameters than by training data [171]. This search may be automated [172, 173], e.g., using quality of predictions for the test set data as the goodness heuristic, though it is unclear that the quality of an ANN necessarily varies monotonically with each parameter. Significantly, in cases where genetic algorithms have been used to search the space of possible ANNs for chemical applications [165, 174], the overall search time was shortened but the resulting neural networks performed at approximately the same level as those created by human trial-and-error. As such, a human-directed optimization search is followed in this work.

5.2 ANN parameters: function and values

A number of neural network subcomponents and parameters are referenced throughout this thesis. These parameters define both the neural network architecture and the way in which it is trained. Functionality of these parameters is discussed in more detail here, and Table 5.1 provides a reference for nearly all of the parameters used in this document.

A prototypical neural network neuron is shown in Fig. 5-1, with inputs p_i , weights w_i , bias b , transfer function f , and output a . The output a as a function of the inputs p_i is shown for the case where inputs are combined with the **dot product** function. This is the default function for combining inputs, however as with most neural network parameters, this can be changed according to the application (e.g., as described in Section 5.3.3). The number of inputs is controlled mainly by the architecture of the system but can also be affected by the training methodology. Neuron inputs can come from the network inputs or the outputs of any other neuron (assuming correct delay lines are included for feedback loops), and the default assumption is that each neuron has an output connection to each of the neurons *in the subsequent layer only*, i.e., feedback connections or connections to layers further ‘downstream’ (in terms of signal travel) must be explicitly specified if desired. Certain training algorithms also support the ability to ‘prune’ connections between neurons, creating a network that is not fully interconnected, however this methodology was not used in this thesis.

Network inputs and outputs are typically normalized before training, to equalize the effect of each channel on the output RMSE calculation, e.g., to values between ± 1 . The broad methodology followed in neural network training is shown in Fig. 5-2. The data used as inputs and targets during training are termed the *training set*, while independent pairs of input/target data are typically reserved for *validating* and/or *testing* the network ‘goodness’ (usually measured with MAE or MSE) in addition to preventing overtaining (the case in which network prediction is tuned well for

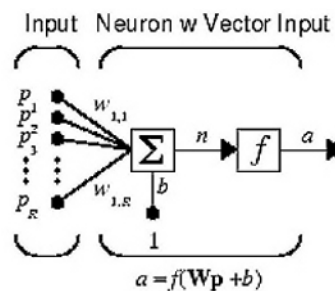


Figure 5-1: Prototypical neuron component of a neural network. *Figure courtesy of the MathWorks.*

the training set but produces poor results on the independent validation or training data). Adjustment of weights is accomplished using a specific training methodology – typically back propagation – and may progress rapidly or slowly depending upon the parameterization of the training method, how long training has been ongoing, etc.

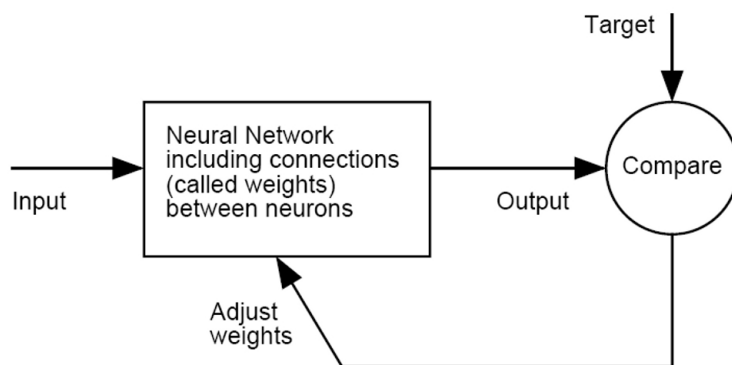


Figure 5-2: Overview of neural network training. Figure from [5].

5.3 ANN architecture with chemical constraints

In order to incorporate chemical constraints into the neural network architecture, a non-traditional format must be explored. The possibility of using feedback, i.e., analogous to use of an op-amp feedback circuit, was explored but eventually rejected due to the lack of already-developed software to support zero-delay feedback loops. Instead, an architecture is proposed in which **certain weights are prevented from being altered during training**. This allows an explicit calculation of weighted sum functions of analyte concentrations, which can then be trained to **as an additional output**.

To explore this further, let us consider a neural network with n concentration outputs $\{C_1, \dots, C_n\}$. Let us assume also that there is measurable quantity A whose relationship to individual concentrations is governed by a weighted sum (dot product) relationship as follows:

$$A = \sum_{i=1}^n a_i \cdot C_i \quad (5.1)$$

In this case a single neuron, using the dot product function to combine its inputs and with weights appropriately set (to the a_i coefficient values), can be used to represent this relationship *as long as the weights are not altered during training*. It is significant to note that both conductivity and charge neutrality take this equation form, implying that both chemical constraints can, in concept, be easily integrated into a relatively simple neural network structure. This formulation, as well as discussion of the practical details associated with input/output scaling and using $\log_{10}(C_i)$ values as targets, are explored in more detail below.

Table 5.1: Neural network characteristics and parameters.

Parameter	Use	Value(s)
Input layer	Contains 1 neuron per input signal.	Any integer # of neurons, varies by application
Hidden layers	Any layers that are not input or output layers. May have any number of neurons per layer, but generally layers progressively decrease in size.	Any integer # of neurons, number of layers typically <5
Output layer	Contains 1 neuron per output signal.	Any integer # of neurons, varies by application
Transfer function	Controls calculation of individual neuron output signal based on input signals (after weighting).	Many options (hardlims, logsig, poslin, satlins, tansig, tribas); tansig is default.
Training goal	Training stops when RMSE meets this goal.	10^{-4} - 10^{-6}
Max epochs	Training stops if full data set is used in training more than this number of times.	100 - 10,000
Learning function	Updates weights after each epoch.	Many options; 'learngdm' is default: $dW = mc * dW_{prev} + (1 - mc) * lr * gW$
Learning rate (lr)	Used by learning function.	Default is 0.01
Momentum constant (mc)	Used by learning function.	Default is 0.9
Weight function	Function for combining weighted inputs.	Many options; 'dotprod' is default.
Training methodology	Method used to update weights based on output errors.	Many options (see [5]); back-propagation methods are the typical default.
μ	Adaptive parameter controlling the magnitude of change to weights due to output errors. (Parameter of training methodology.)	Default is 0.001
μ_{inc}	Increase factor for μ .	Default is 10
μ_{dec}	Decrease factor for μ .	Default is 0.1
Error weights	Can be used to selectively weight error on different output channels in RMSE calculation.	Default is vector of 1s

5.3.1 Network architecture

Implementation of these constraints is visualized here for an example network with the following characteristics:

- 13 input signals.
- Two hidden layers with 6 and 12 neurons with, respectively, tansig and purelin transfer functions.
- 12 target outputs.

Figure 5-3 shows the Matlab-formatted representation for this network, along with such a network that has one or two additional constraint layers appended as specified above. One will notice that the basic network structure is unchanged and that the weights and bias values for the constraint layers are all marked as ‘untrained.’ Yellow nodes mark points at which inputs are normalized / outputs are un-normalized, such that training takes place using ‘fairly’ weighted data on every channel (as explained above).

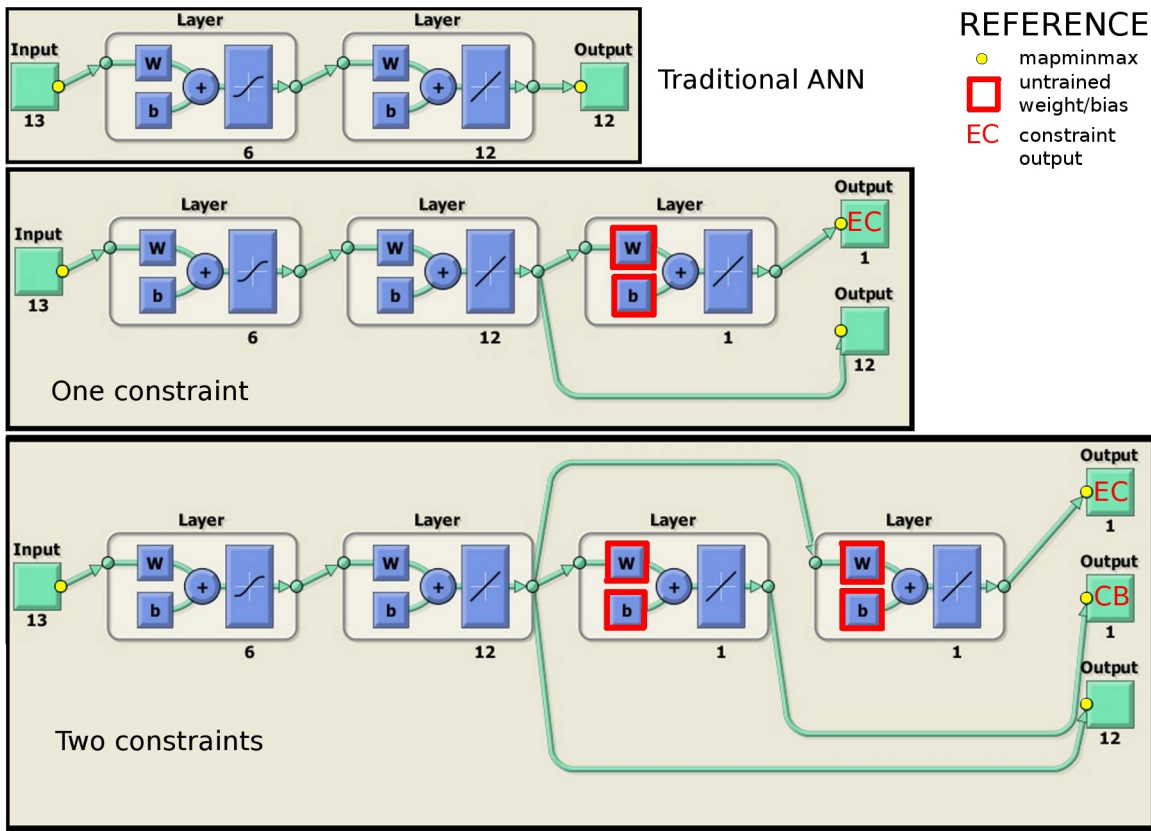


Figure 5-3: Matlab-formatted representation of three neural network architectures: a traditional structure (top), one with a single constraint layer (middle), and one with two constraint layers (bottom). (Note that the middle case is labeled EC for the Electrical Conductivity case but could represent either of the chemical constraints discussed here.) Weights and biases omitted from training in the non-traditional architectures are boxed in red, while nodes where Matlab applies the mapminmax (or inverse) function are highlighted in yellow.

5.3.2 Assigning weights to implement chemical constraints

Because constraint vectors take their input from a point in the network *where signals are still normalized*, assignment of weight and bias values to these ‘untrained’ layers must take into account both the natural chemical relationship being modeled and the normalization of the data incorporated into the neural network training algorithm. This section derives appropriate such values for the electrical conductivity relationship; implementation of a charge balance constraint follows an identical process.

Recall that electrical conductivity (EC) can be related to the ions in solution via the following relationship [175], where the γ_i coefficients represent the molar conductance in $\frac{\mu S/cm}{mmol/L}$ of species C_i (see Table 5.2).

$$EC = \sum_{i=1}^n \gamma_i \cdot [C_i] \quad (5.2)$$

For a network without input/output normalization, the weight values would be identical to the γ_i coefficients and the corresponding bias signals would be zero. Because of the difference in magnitude of output signals, however, such a network would be unlikely to accurately estimate small-magnitude signals. As such, inputs and outputs are normalized (using Matlab’s built-in *mapminmax* function) before training as follows. For input vector \mathbf{C}_i (e.g., $[Ca^{2+}]$ for the entire suite of training samples):

$$\tilde{\mathbf{C}}_i = \frac{(y_{max} - y_{min})}{(C_{i,max} - C_{i,min})} \cdot (\mathbf{C}_i - C_{i,min}) + y_{min} \quad (5.3)$$

where y_{min} and y_{max} are chosen by the user (typically ± 1 , and which can be set independently for each signal), $C_{i,min}$ is the minimum value in the \mathbf{C}_i vector, and $C_{i,max}$ is the corresponding maximum value. By defining C_i and y ranges, this can be rewritten as

$$\tilde{\mathbf{C}}_i = \frac{y_{range}}{C_{i,range}} \cdot (\mathbf{C}_i - C_{i,min}) + y_{min} \quad (5.4)$$

which can be inverted to yield the original concentration value as

$$\begin{aligned} \mathbf{C}_i &= (\tilde{\mathbf{C}}_i - y_{min}) \cdot \frac{C_{i,range}}{y_{range}} + C_{i,min} \\ &= \alpha_i \cdot \tilde{\mathbf{C}}_i + \beta_i \end{aligned} \quad (5.5)$$

where α_i and β_i are defined by the known constants y_{min} , y_{range} , and $C_{i,range}$. The calculation for electrical conductivity (for all samples) thus becomes

$$\mathbf{EC} = \sum_{i=1}^n \gamma_i \cdot \alpha_i \cdot \tilde{\mathbf{C}}_i + \sum_{i=1}^n \gamma_i \cdot \beta_i \quad (5.6)$$

At this juncture it is necessary to account for the normalization on the EC channel itself, i.e., because the calculation performed by the neuron must actually output $\tilde{\mathbf{EC}}$, the normalized EC signal which will be subsequently restored automatically by Matlab when the inverse *mapminmax* function is applied.

$$\begin{aligned}
\tilde{\mathbf{E}}\mathbf{C} &= \frac{(y_{max} - y_{min})}{(EC_{max} - EC_{min})} \cdot (\mathbf{E}\mathbf{C} - EC_{min}) + y_{min} \\
&= \frac{y_{range}}{EC_{range}} \cdot (\mathbf{E}\mathbf{C} - EC_{min}) + y_{min} \\
&= \alpha_{EC} \cdot \mathbf{E}\mathbf{C} + \beta_{EC}
\end{aligned} \tag{5.7}$$

Combining this with Eqn. 5.6 and distributing coefficients to separate terms yields

$$\begin{aligned}
\tilde{\mathbf{E}}\mathbf{C} &= \sum_{i=1}^n \alpha_{EC} \cdot \gamma_i \cdot \alpha_i \cdot \tilde{\mathbf{C}}_i + (\beta_{EC} + \sum_{i=1}^n \alpha_{EC} \cdot \gamma_i \cdot \beta_i) \\
&= \sum_{i=1}^n w_i \cdot \tilde{\mathbf{C}}_i + B
\end{aligned} \tag{5.8}$$

From this it can be seen that the individual weights for connections from each signal $\tilde{\mathbf{C}}_i$ to the EC layer should be set to the product $\alpha_{EC} \cdot \gamma_i \cdot \alpha_i$ and the bias connection to the EC layer should be set to $(\beta_{EC} + \sum_{i=1}^n \alpha_{EC} \cdot \gamma_i \cdot \beta_i)$.

The corresponding calculation for a charge balance layer progresses identically, however it is important to determine exactly what relationship is being encoded with this constraint. It is possible to calculate a full charge balance, i.e.,

$$\sum_i |z_i| \cdot [C_i^{z_i^+}] - \sum_i |z_i| \cdot [C_i^{z_i^-}] \approx 0 \tag{5.9}$$

however, this may not be the most appropriate formulation for the constraint due to the manner in which neural nets are implemented. Use of the equation in this form requires that the entire target vector would contain zeros, and consequently the normalization stage would result in undefined values. It is instead better to move a few terms (e.g., those corresponding to the carbonate system or pH) to the right-hand side of the equation. For example:

$$\begin{aligned}
&([Na^+] + [K^+] + [NH_4^+] + [2 \cdot Ca^{2+}] + [2 \cdot Mg^{2+}]) - \\
&([Cl^-] + [NO_3^-] + [2 \cdot SO_4^{2-}] + [HCO_3^-] + [2 \cdot CO_3^{2-}]) \\
&= [OH^-] - [H^+]
\end{aligned} \tag{5.10}$$

This particular formulation has been chosen for the work described in this thesis since, in theory, the pH of the system is relatively well known and should thus be a good signal with which to constrain the rest of the system. The coefficient values required to calculate the weight and bias values for the charge balance constraint layer are correspondingly given in Table 5.2.

5.3.3 Weight constraints with logarithmic targets

Normalization of input and output data has been described above as a method for ‘fairly’ weighting contributions of all signals to MSE calculations (and thus to ANN training algo-

Table 5.2: Charge balance and conductivity constraint multipliers used for calculation of non-trained neuron weights. Conductance values adapted from [6, 7]. *Note that the charge balance constraint has been formulated such that the balance of all other ions is trained to the net contribution from H^+ and OH^- as explained further in the text.

Analyte	γ_{CB} []	γ_{EC} [$\frac{\mu\text{S}/\text{cm}}{\text{mmol}/\text{L}}$]
Na^+	+1	50.10
K^+	+1	73.5
Ca^{2+}	+2	119
NH_4^+	+1	73.5
Mg^{2+}	+2	106
SO_4^{2-}	-2	160
Cl^-	-1	76.46
NO_3^-	-1	71.46
H^+	0*	349.6
OH^-	0*	199.1
HCO_3^-	-1	44.5
CO_3^{2-}	-2	138.6
HSO_4^-	-1	52.0

rithms), however it should be noted that the linear transformation performed by *mapminmax* may not result in optimally ‘fair’ weighting in cases where signals vary over several orders of magnitude *on a single input or output channel*. In this case it may be advisable to perform a log-transformation on the data before passing it to the *mapminmax* function, but transformation of the data in this manner is not directly compatible with the constraint methodology mapped out above. It is necessary in this case to both recalculate appropriate weight values *and to create a new method for calculating weighted inputs* as the dot product will no longer correctly represent the known chemical relationship. The mathematics of these alterations are described in detail in this section.

Now for an input vector \mathbf{C}_i with all elements >0 , we perform two stages of normalization:

$$\mathbf{L}_i = \log_{10} \mathbf{C}_i \quad (5.11)$$

$$\tilde{\mathbf{L}}_i = \frac{(y_{max} - y_{min})}{(L_{i,max} - L_{i,min})} \cdot (\mathbf{L}_i - L_{i,min}) + y_{min} \quad (5.12)$$

where the signals available internal to the neural network (previously $\tilde{\mathbf{C}}_i$) correspond in this case to $\tilde{\mathbf{L}}_i$. The EC calculation, however, must still be made using \mathbf{C}_i , so it is necessary to manipulate these equations such that \mathbf{C}_i is given as a function of other system parameters. **Note: the mathematics derived here apply to a network for which concentration values are both log- and mapminmax-transformed while the constraint (EC or CB) values are only processed using mapminmax.**

$$\begin{aligned} \tilde{\mathbf{L}}_i &= \frac{y_{range}}{L_{i,range}} \cdot (\mathbf{L}_i - L_{i,min}) + y_{min} \\ &= \alpha_i \cdot \mathbf{L}_i + \beta_i \end{aligned}$$

$$= \alpha_i \cdot \log_{10} \mathbf{C}_i + \beta_i \quad (5.13)$$

Inverting this equation gives \mathbf{C}_i as a function of other system parameters:

$$\begin{aligned} \mathbf{C}_i &= 10^{\frac{\tilde{\mathbf{L}}_i - \beta_i}{\alpha_i}} \\ &= 10^{\frac{\tilde{\mathbf{L}}_i}{\alpha_i}} \cdot 10^{-\frac{\beta_i}{\alpha_i}} \\ &= \delta_i \cdot 10^{\frac{\tilde{\mathbf{L}}_i}{\alpha_i}} \end{aligned} \quad (5.14)$$

Substituting this into the known EC relationship (Eqn. 5.2 and correcting for the mapmin-max normalization on the EC data (Eqn. 5.7) yields

$$\mathbf{EC} = \sum_{i=1}^n \gamma_i \cdot \delta_i \cdot 10^{\frac{\tilde{\mathbf{L}}_i}{\alpha_i}} \quad (5.15)$$

$$\begin{aligned} \tilde{\mathbf{EC}} &= \sum_{i=1}^n \alpha_{EC} \cdot \gamma_i \cdot \delta_i \cdot 10^{\frac{\tilde{\mathbf{L}}_i}{\alpha_i}} + \beta_{EC} \\ &= \sum_{i=1}^n w_{1,i} \cdot 10^{w_{2,i}} \cdot \tilde{\mathbf{L}}_i + B \end{aligned} \quad (5.16)$$

As is clear from this formulation, **two** weights are not required for each input signal to calculate EC as a function of the normalized $\tilde{\mathbf{L}}_i$, and the calculation is no longer in dot product form. While no weighting function of this form is available from the Matlab Neural Network Toolbox (built in functions stored in `\MATLAB\toolbox\nnet\nnet\nnweight`), it is possible to program a custom weighting function which can then be assigned as the `layer.WeightFcn` within the standard neural network framework. This custom function must include the following:

- size of weight vector w required for input vector (to this layer) p of size $r \times s$
- output z as a function of inputs p and weights w
- $\frac{dz}{dp}$ as a function of w, p
- $\frac{dz}{dw}$ as a function of w, p

A function of this sort, named `cbsum.m`, was created in Matlab and installed in the NNET toolbox (see Appendix B). Note that, while the last two items (partial derivatives) are required for the function to work, they will not affect neural network training in cases where training is disabled on the corresponding weights, i.e., in all cases where this function is currently being used.

5.4 Conclusions

The Matlab Neural Network toolbox provides a powerful, flexible framework for the creation of a wide range of neural network architectures. When combined with custom weighting functions and the ability to omit certain weights from alteration during training, network architectures implementing known chemical constraints can be designed using standard Matlab functions. Assignment of weights and bias values based on known constraint equations requires attention to data normalization as well as inherent chemical relationships;

this chapter has laid out the fundamental requirements for implementing both conductivity and charge balance constraints in a neural network outputting concentrations (or log concentrations) of the majority of ions found in fresh waters.

Extension of such constraint-incorporation techniques has the potential to add ANNs to the growing toolbox of methods available for analysis of environmental datasets. ANNs have an inherent advantage in cases where (1) the relationship between the environmental factor and measurement are poorly understood or poorly constrained, (2) the relationship is expected to be non-linear, and (3) collection of sample data is less costly/time-consuming than the experiments required to constrain relationship parameters. In cases where multiple factors are of interest, e.g., chemical speciation within a closed system, and additional knowledge is available at the system level, e.g., mass balance for specific atoms, water flows, etc., this novel ANN formulation may provide a straightforward way to incorporate *a priori* knowledge with available sample data to further improve estimation capabilities.

Chapter 6

Proof-of-concept: 1-Anion Subsystem

Abstract

Commercially-available ion selective electrodes (ISEs) have the potential to support rapid in-situ measurement of ion concentrations in natural waters if cross-ion interferences and nonlinearities can be overcome; we investigated use of artificial neural networks (ANNs) in this capacity. A semi-synthetic data set representative of ISE responses to a range of natural waters (from Massachusetts (MA) and Texas (TX)) was produced using actual water analyses and multi-ion calibrations for 9 ISEs. The resulting signals were processed using an ANN optimized to predict concentration values for each ion. Results showed an improvement of up to 3 orders of magnitude relative to use of ISEs as stand-alone sensors, with useful results down to $\sim 10 \mu\text{M}$ even for non-dominant analytes. Accuracy (removal of bias) is accomplished at all concentration levels, while precision (decrease in scatter about the target) is achieved relatively better at higher concentrations. Networks are trained with MA-only and MA+TX data sets to examine (1) degradation in prediction quality when the ANN is used to process data outside the training range and (2) subsequent improvement with a more representative training set. Non-traditional ANN structures that incorporate constraints based on conductivity and charge neutrality are shown to further improve results despite a dependence on quantification of the carbonate system in the absence of a carbonate-specific electrode, although improvements are not evenly distributed across channels. The proposed hardware/software system holds promise for deployment in field settings without requiring development of new sensor hardware.

6.1 Introduction

The concentrations of major ions in natural waters profoundly influence ecosystem structure and health, driving changes in natural floral and faunal species and affecting suitability of waters for human use. Unfortunately, due to the high cost of grab-sampling campaigns and laboratory analysis, current ionic data sets are often incomplete and have low spatiotemporal resolution and thus may not provide the information necessary to diagnose or remediate threats to ecosystems, e.g., the identification of the particular species causing environmental degradation. One historical example concerns the causes of stream acidification: initially many researchers accepted a paradigm in which sulfate deposition was the driving force,

whereas full ionic analyses of actual waters demonstrated similar contributions from nitrate (and/or organic N species) on many watersheds. In contrast, data covering the full spectrum of dissolved ionic species, especially when available in real-time, would create the ability to lead high-resolution, adaptive studies, resulting in more precise identification of problems and better-targeted follow-on sampling if additional lab analyses are required.

Commercially available portable low-power sensors such as ion-selective electrodes (ISEs) are available for measurement of a large number of ions and could have promise in this capacity if cross-ion interferences could be addressed without sample pre-processing. We have thus undertaken development of an ISE-based in-situ instrument for real-time identification of dissolved ion species, targeting both the dominant ionic species and those frequently involved in eutrophication or acidification (specifically: Na^+ , K^+ , Ca^{2+} , Mg^{2+} , NH_4^+ , Cl^- , NO_3^- , SO_4^{2-} , and the carbonate and pH systems) by coupling a suite of ISEs with novel signal processing that allows mixed responses to be successfully decoupled. Relevant environmental parameters measured simultaneously provide additional information that improves accuracy of the results.

The liquid-media multi-sensor array was pioneered by Otto and Thomas [114], who employed an array of five electrodes (selective and cross-selective) with partial least-squares regression (PLS) for the simultaneous quantification of Ca^{2+} , Mg^{2+} , Na^+ , and K^+ in biological liquids (where measurement background is fairly constant). Quantification by ISE-array was later introduced for environmentally-relevant applications such as for heavy metals [131, 132, 133], inorganic pollutants in modeled groundwater (Mn(II) , Fe(III) , Ca^{2+} , Mg^{2+} , Na^+ , Cl^- , and SO_4^{2-}) [133, 137], and small sets of inorganic ions [110, 140, 141], although these applications to polluted systems did not extend analysis to natural concentration levels. Typically, such systems use artificial neural networks (ANNs) to process data from the matrix of sensor signals, although PLS systems continue to be developed for side-by-side comparison [116, 118, 131, 137]. No definitive trend is visible in such comparisons: PLS produced similar results in [116, 118] and worse results in [131, 137]. In recent years, artificial neural networks have become the de facto algorithm of choice for such applications due to the ease with which non-linear functions can be approximated using this format.

In the present work we use a novel ANN architecture to process data from an ISE sensor array **while incorporating known chemical constraints by disabling training on a subset of neurons**. The ANN architecture is particularly useful here as the network structure essentially approximates the non-linear transfer function describing the relationship of single-ion concentrations to the suite of sensor measurements *without requiring the user to specify the form of this non-linear function*. Such capability is particularly necessary when using ISE hardware in situ, as recent research has demonstrated that the coefficients used in traditional models, e.g., Nernst, Nikolski-Eisenman, are often dependent on temperature as well as the concentrations of the primary and interfering analytes and ISE response is often non-linear in the sub- $10\mu\text{M}$ range where natural concentrations may lie.

An initial test of this combined hardware/software technique was undertaken by using a data set derived from ion concentration data, collected using traditional sampling techniques and published by the US Geological Survey, to approximate the expected response of the sensor suite to real environmental samples. This electrode response data is then processed using an optimized ANN structures, *with chemical constraints incorporated*, to produce concentration data for the analytes of interest, with results evaluated over a range of ionic concentrations.

6.2 Experimental (materials and methods)

The electrode response data used in this work was created in the following stages, detailed below:

1. Electrode characterization
2. Collection and pre-processing of water quality data published by the USGS for water samples in which all major ions were measured
3. Simulation of electrode response, including the addition of measurement noise.

A subset of the simulated electrode response signal / actual USGS data pairs were used to train a range of ANN structures, and the resulting ANNs were characterized with respect to their success in predicting accurate concentrations for two disparate data sets.

6.2.1 Electrode characterization

Description and characterization of the electrodes used in this study is described in [167]. Briefly, both glass and solid-state ISEs were used, with a single electrode marketed as sensitive to each of Ca^{2+} , K^+ , NH_4^+ , pH and hardness, and two electrodes each for Cl^- and Na^+ . In addition, probes measuring temperature and conductivity were included in the hardware suite. For the purposes of this manuscript, the following nomenclature will be used: ‘primary’/‘named’ refers to the ion for which the ISE is marketed, while ‘interfering’ refers to all other ions to which the ISE responds to some degree. In cases where response is strong to a number of ions, interfering ions may further be referred to as ‘secondary’, ‘tertiary’, etc., ordered by the magnitude of their response.

The response of each ISE was measured for all ions considered in this experiment. Typical response curves are shown in Fig. 6-1, where each curve describes the response of a single ISE in a single salt solution (i.e., one cation/anion pair). This figure demonstrates that (a) in some cases ISE responses can approximate Nernstian even for interfering ions (left), and (b) nearly all electrodes respond, with different magnitude and detection limit, to ions of similar charge to their primary analyte (right).

Response slope and single-salt Limit of Detection (LOD) figures were quantified using these response curves; LOD levels are in the μM range, making these sensors theoretically usable at environmental levels (see Table 6.1). Because ISE response is typically not log-linear at the low end of the response range, the LOD is listed as the concentration below which response cannot be differentiated from the noise baseline (1σ level).

6.2.2 Water quality data: selection, filtering, pre-processing

Data published by the USGS [8] for two distinct geologic regions of the United States (Massachusetts and Texas) were adapted to train our ANN algorithm, test the response of the trained ANN algorithms to the ionic makeup of realistic natural waters, and test the ANN response to data dissimilar to those used in training. Data from freshwater river, stream, and lake sites measured most frequently by the USGS over the past 50 years were screened for samples for which the full ionic composition of the water was measured simultaneously. Approximately 600-700 such water samples were found for each of the two regions. (This number is extremely small relative to the total number of measurements taken by the USGS over this time period, underscoring the need for the proposed in-situ instrumentation.) A detailed list of the provenance and number of samples is given in Tables

Table 6.1: Experimental Characterization of ISE limits of detection (LOD). Primary refers to the ‘named’ ion of selectivity (e.g., Ca^{2+} for the ELIT Ca^{2+} ISE), while secondary, tertiary, and quaternary (analyte indicated in parenthesis after the LOD value) are ordered by response magnitude ($\text{mV}\cdot\text{M}^{-1}$) and not the LOD value. Note that Cl^- was the only cation in this study (excepting OH^- at low, fairly constant concentration) and thus does not have data for response to non-primary ions.

ISE	Primary	Secondary	Tertiary	Quaternary
Ca^{2+} (ELIT)	$0.30 \mu\text{M}$	N/A (NH_4^+)	N/A (Na^+)	N/A (K^+)
K^+ (ELIT)	$0.086 \mu\text{M}$	$0.25 \mu\text{M}$ (NH_4^+)	$10.6 \mu\text{M}$ (Na^+)	N/A (Ca^{2+})
Na^+ (ELIT)	$0.17 \mu\text{M}$	$0.5 \mu\text{M}$ (K^+)	$1.35 \mu\text{M}$ (NH_4^+)	$0.25 \mu\text{M}$ (Ca^{2+})
NH_4^+ (ELIT)	$0.22 \mu\text{M}$	$0.25 \mu\text{M}$ (K^+)	$8.6 \mu\text{M}$ (Na^+)	N/A (Ca^{2+})
Cl^- (ELIT)	$2.1 \mu\text{M}$			
Cl^- (Hanna)	$2.2 \mu\text{M}$			
Hardness (Thermo)	$9.5 \mu\text{M}$	$1000 \mu\text{M}$ (NH_4^+)	$9200 \mu\text{M}$ (Na^+)	N/A (K^+)
Na^+ (Ross – glass)	$41.1 \mu\text{M}$	$147 \mu\text{M}$ (NH_4^+)	N/A (K^+)	N/A (Ca^{2+})

6.2 and 6.3, while a partial comparison of the MA and TX data is shown in Figs. 6-2 and 6-3.

Table 6.2: Data from USGS sites in Massachusetts [8].

Site Number	Site Name	Number of points	Date range
01096550	Merrimack River	86	28-Oct-1980 to 17-Jan-1995
01102500	Aberjona River	93	23-Oct-1998 to 07-Sep-2001
01103500	Charles River	58	28-Oct-1980 to 23-Nov-1994
01109000	Wading River	34	20-Oct-1998 to 10-Sep-2001
01111230	Blackstone River	77	18-Nov-1980 to 12-Jun-2002
01111500	Branch River	23	08-Mar-1993 to 12-Jun-2002
01116500	Pawtuxet River	23	11-Mar-1993 to 11-Jun-2002
01118500	Pawcatuck River	98	20-Nov-1980 to 01-Aug-2002
01170100	Green River	44	25-Mar-1993 to 07-Sep-2001
01198000	Green River 2	1	22-Aug-1994
01198125	Housatonic River	129	11-Feb-1993 to 09-Jan-2009
	TOTAL	666	28-Oct-1980 to 09-Jan-2009

To create a set of test data that represented as closely as possible these natural waters examined by the USGS while being compatible with the suite of ISEs being used, these data were altered to contain only a single major anion (Cl^-) (in addition to OH^- and bicarbonate) plus the ISE-represented cations (Ca^{2+} , K^+ , Na^+ , NH_4^+ , H^+). Chloride concentration was adjusted to compensate for the omitted anions by invoking charge neutrality against the above-listed cations and assuming equilibrium with atmospheric CO_2 . Conductivity was then calculated using the relationship and data given in [175, 176]. Hardness was adjusted for the omission of Mg^{2+} and is thus based solely on the measured Ca^{2+} concentrations. Mean changes made to these parameters are shown in Table 6.4.

Table 6.3: Data from USGS sites in Texas [8].

Site Number	Site Name	Number of points	Date range	
07227500	Canadian River near Amarillo	85	13-Jan-1993	to
			04-Feb-2009	
08062500	Trinity River near Rosser	94	19-Jan-1993	to
			25-Nov-2008	
08066500	Trinity River at Romayor	107	15-Oct-1980	to
			28-Sep-1995	
08114000	Brazos River at Richmond	119	28-Oct-1980	to
			25-Sep-2002	
08188500	San Antonio River at Goliad	101	14-Oct-1980	to
			19-Dec-2006	
315203097222601	Whitney Lake (Site AC) near Whitney	36	17-Feb-1999	to
			03-Aug-2004	
324932098575101	Hubbard Creek Lake (Site P1) near Breckenridge	66	07-Jan-1993	to
			21-Aug-2008	
	TOTAL	608	14-Oct-1980	to
			04-Feb-2009	

Table 6.4: Mean change in parameters, given as (simulated value - recorded value) for chloride, conductivity, and hardness (percentage change relative to the mean of measurements is given in parentheses), resulting from the creation of the semi-synthetic data set from actual ionic data measured by the USGS.

Location	Δ chloride (%)	[mM]	Δ conductivity [uSiemens/cm] (%)	Δ hardness [mM CaCO ₃] (%)
MA	0.64 (56.3)		-13.2 (-5.2)	-0.16 (-28.9)
TX	3.3 (56.5)		49.9 (4.3)	-0.72 (-29.0)

6.2.3 Simulated electrode responses

The processed USGS data were then used to simulate responses approximating those expected from the studied ISE suite had they been immersed in the corresponding actual water samples; this process required (1) calculation of electrode response to primary and interfering ions and (2) addition of random noise. The mV response of each electrode to each ion was calculated using single-salt response curves (e.g., Fig. 6-1), with the total response being a summation of the individual excursions beyond baseline (following the Nikolski-Eisenman equation). Magnitude of additive Gaussian random noise was estimated using the standard errors around the linear fit for each of the single-salt calibration curves. The resulting semi-synthetic data set was thus constrained by actual measured cation concentrations but was also chemically self-consistent within the framework of this proof-of-concept experiment.

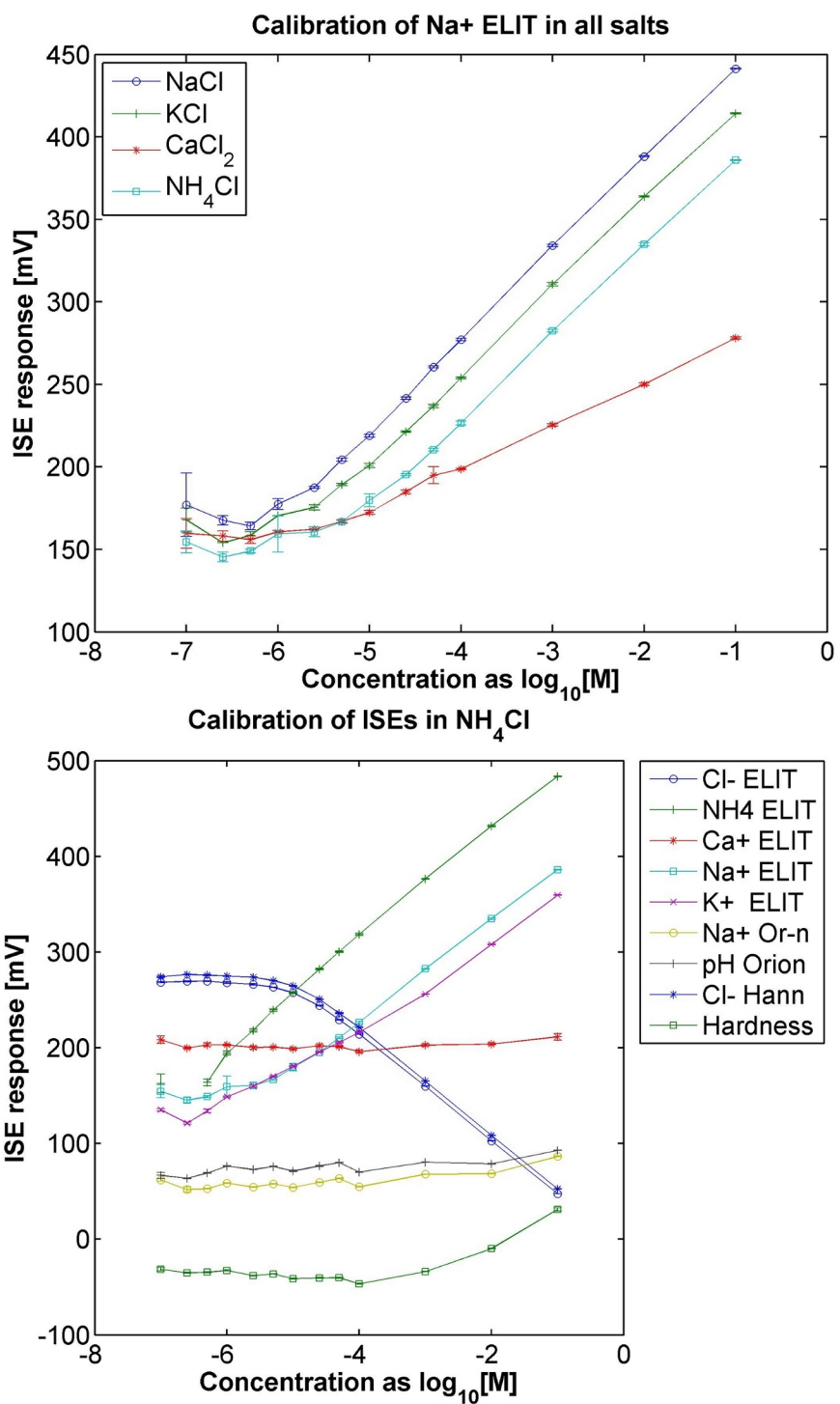


Figure 6-1: Electrode response to a range of salts. (Top) Response of the ELIT Na⁺ ISE to four cations. (Bottom) Response of 9 electrodes to different concentrations of NH₄Cl solution. Note that log-linear responses exist for most ISE/ion pairs, indicating that use of these electrodes in mixed-salt solutions will produce responses due partially to each of the ionic constituents.

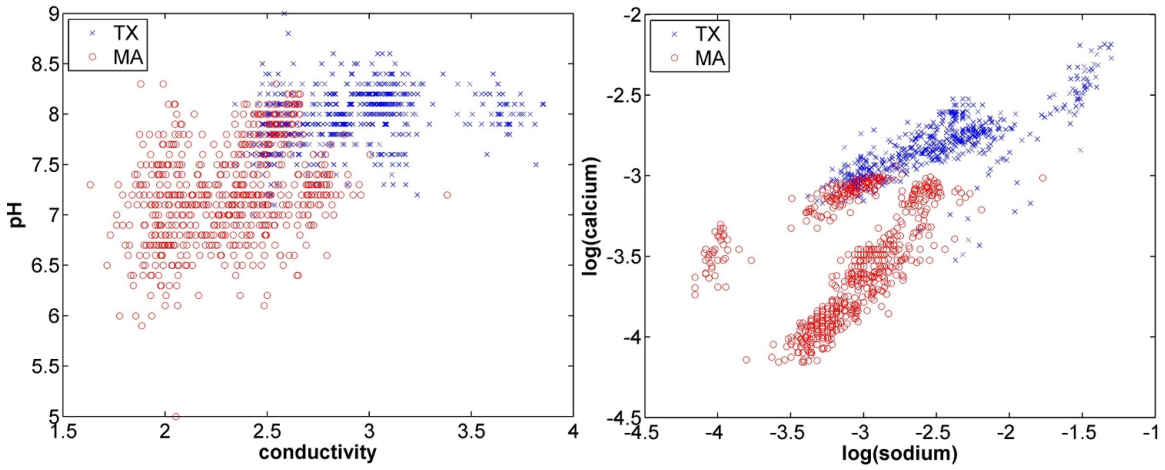
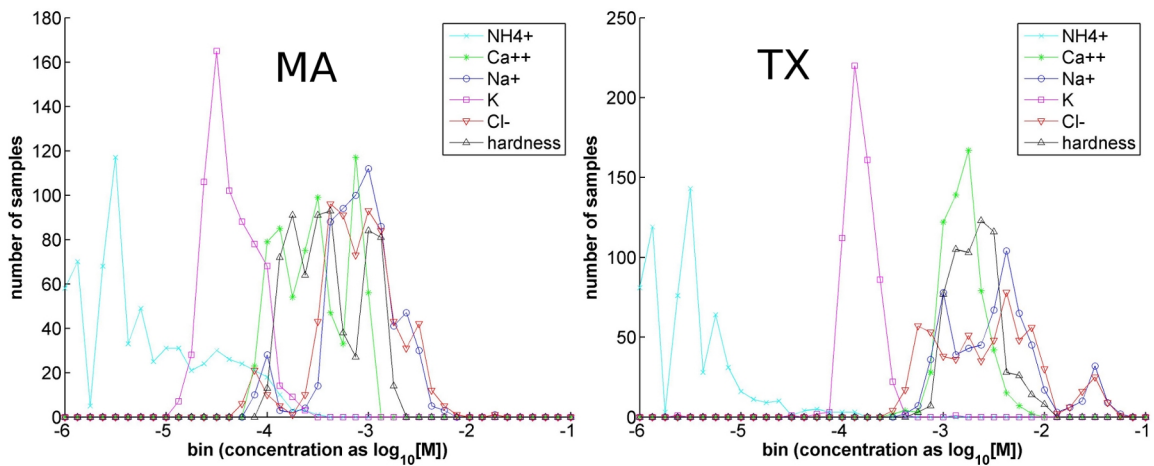


Figure 6-2: Direct comparison of MA and TX data 'fingerprints'.

Figure 6-3: Ionic characteristics of Massachusetts (left) and Texas (right) data selected for simulated data set.



6.2.4 ANN training and use

Ion composition data and the corresponding simulated ISE response data were used to train a range of feed-forward ANN structures using the Matlab R2010b Neural Network Toolbox v7, where the ISE responses were taken as the ANN inputs and the USGS data as modified were taken as the targets. The general structure of these feedforward ANNs is given in Fig. 6-4.

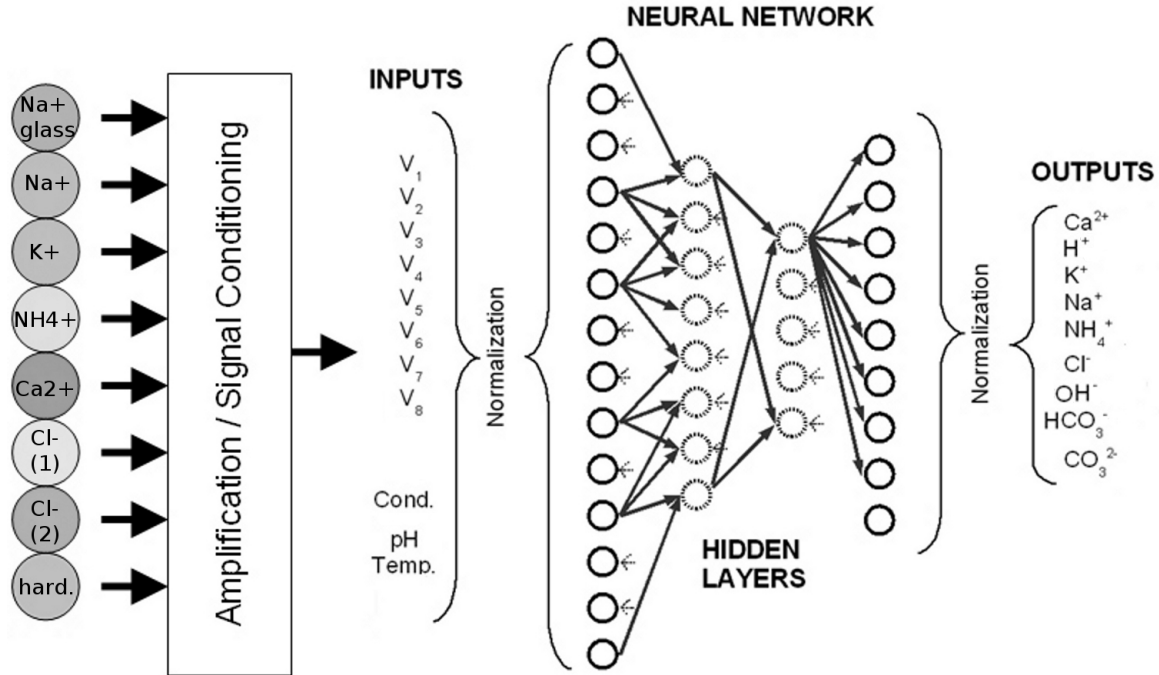


Figure 6-4: Basic feedforward ANN structure used as the starting point for training.

Input and target data were normalized using `mapminmax` such that the range of each input and target data vector was $[-1,1]$, and a linear transfer function was used between the last hidden layer and the output layer of the network. Parameterization of the rest of the ANN architecture was varied over the following ranges:

- Number of hidden layers: 1–3
- Nodes per layer: 3–20
- Transfer function: `tansig`, `poslin`, `satlin`
- μ : [0.1 0.001]
- μ_{dec} : [0.1 0.5 0.9]
- μ_{inc} : [1.5 10 50]

The maximum number of epochs allowed to complete training was set to 10^4 , and the training goal was MSE of 10^{-6} . Total Mean Square Error (MSE) was calculated as the sum of MSE for each channel, on the normalized data. (For cases where constraints were implemented, the MSE for each constraint channel was included in the formula for total MSE.) Default learning rate and momentum constant were used for the `learnsgdm` function, and the dot product was used as the base weight function. Hidden layers and nodes per layer can be understood in reference to Fig. 6-4; for more information about the role of other parameters and their use in Matlab, please see [107, 5].

Constraints based on conductivity and charge balance were built into the ANN structure by explicitly setting a subset of connection weights which are then omitted from training to preserve the appropriate mathematical relationship. Fig. 6-5 shows the configuration of output layers used to implement the charge balance constraint; construction of the conductivity constraint was completed in an analogous manner (further discussion provided in Appendix 5). During training, these layers could also be omitted, resulting in a simpler (traditional fully-trained) feedforward ANN structure.

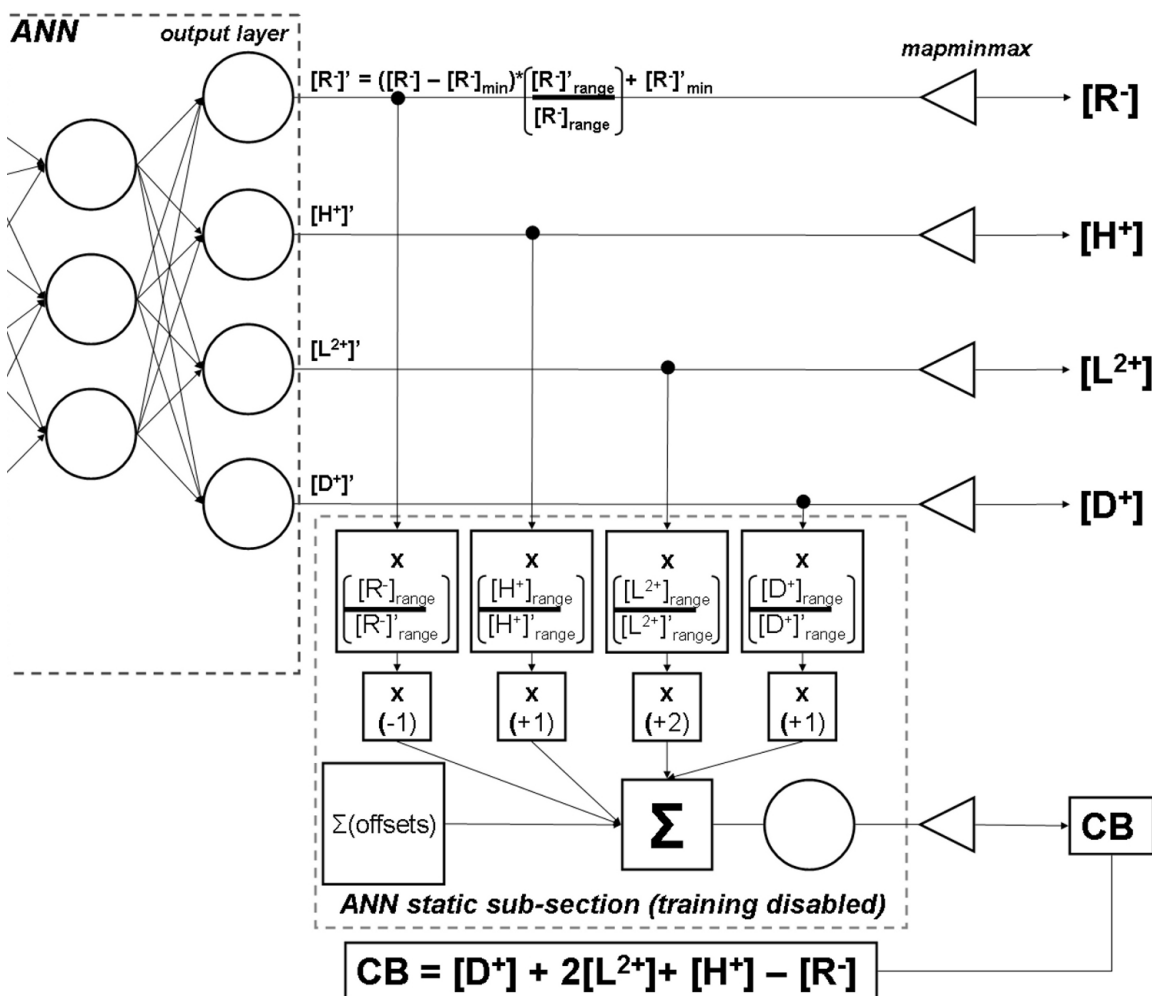


Figure 6-5: Inclusion of charge balance constraint via addition of a non-trained output layer in the ANN. Hidden and output layers shown on the left refer to those included in the generic feedforward ANN structure shown in Fig. 6-4.

The data were subsequently split into three subsets: training data (50% of data) used to optimize the network weights, validation data (20% of data) used to determine the optimal stopping point for training (to prevent overtraining), and test data used to assess the accuracy of the trained network on an independent set of data. To maximize the representativeness of the training data set, its constituents were randomly chosen from the initial data set, excepting that the maximum and minimum values for each analyte were guaranteed to be included in the training set. Optimal ANNs and parameterizations

were identified using the normalized root mean square error (NRMSE) for concentration parameters. Because of the geochemical importance and the generally lower concentration of NH_4^+ compared to other major cations, errors in NH_4^+ were weighted at 10-times errors on other channels. To examine the generalization of the resulting ANNs, training was conducted with (1) MA data only and (2) both MA and TX data. These optimized ANNs were used to process the two regional data sets, and NRMSE is reported and compared for each data set.

6.3 Results

Optimized ANNs were identified for each of the training cases identified above (MA data only, combined MA and TX data) for networks in which both conductivity and charge balance constraints were included. The tansig transfer function was always found to produce the best results, along with the smallest training goal (i.e., goal of lowest mean square error) and with $\mu = 0.1$. Generally, approximately 30 nodes are required to adequately represent this data, spread over two or three hidden layers, with optimal ANNs having the following hidden layers: MA=(18), MA+TX = (18-12). MA-only training required more extreme changes in μ throughout training ($\mu_{inc} = 50$, $\mu_{dec} = 0.1$) while MA+TX training was optimized with slower changes in μ ($\mu_{inc} = 1.5$, $\mu_{dec} = 0.5$).

Results using the MA-trained ANN to analyze MA data (Fig. 6-6) show a substantial improvement in accuracy (removal of bias offset due to cross-interferences of secondary/tertiary ions) as well as an improvement in precision, although some scatter is still visible at the relatively lower concentrations. Concentration estimates are improved down to the 1–10 μM level even for non-dominant analytes (e.g., NH_4^+).

The change in normalized RMSE (NRMSE) due to the post-processing of the ISE signals using the optimized ANN is shown in Table 6.5 (MA-trained ANN, MA data only). While not exactly equal to the relative percent error, these numbers can be roughly interpreted in an analogous manner; an NRMSE of 10^2 indicates that, on average, results are approximately two orders of magnitude higher than the target values. Generally, NRMSE values for ISEs alone are seen to be high due to the additive interferences from other constituents in solution (i.e., all concentrations are overestimated). Use of the ANN improves estimates relative to use of ISEs as stand-alone sensors by up to three orders of magnitude. Inclusion of constraints based on conductivity and charge balance improve the overall NRMSE, with the most significant improvements seen for ammonium concentration estimates and slightly worse estimates on some other channels as a trade-off.

To determine the extent to which a MA-trained ANN would successfully predict analyte concentrations for TX data, which has a substantially different ionic ‘fingerprint’, the optimal ANN described above was used to predict analyte concentrations for a combined MA and TX data set (Table 6.6). In all cases, the ANN NRMSE for TX data is higher than for the corresponding MA data set, though the predictions are still better than for ISEs alone. Because the TX data had generally higher ion concentrations, often outside the range of the MA data set, this degradation in performance was expected; further implications are discussed below.

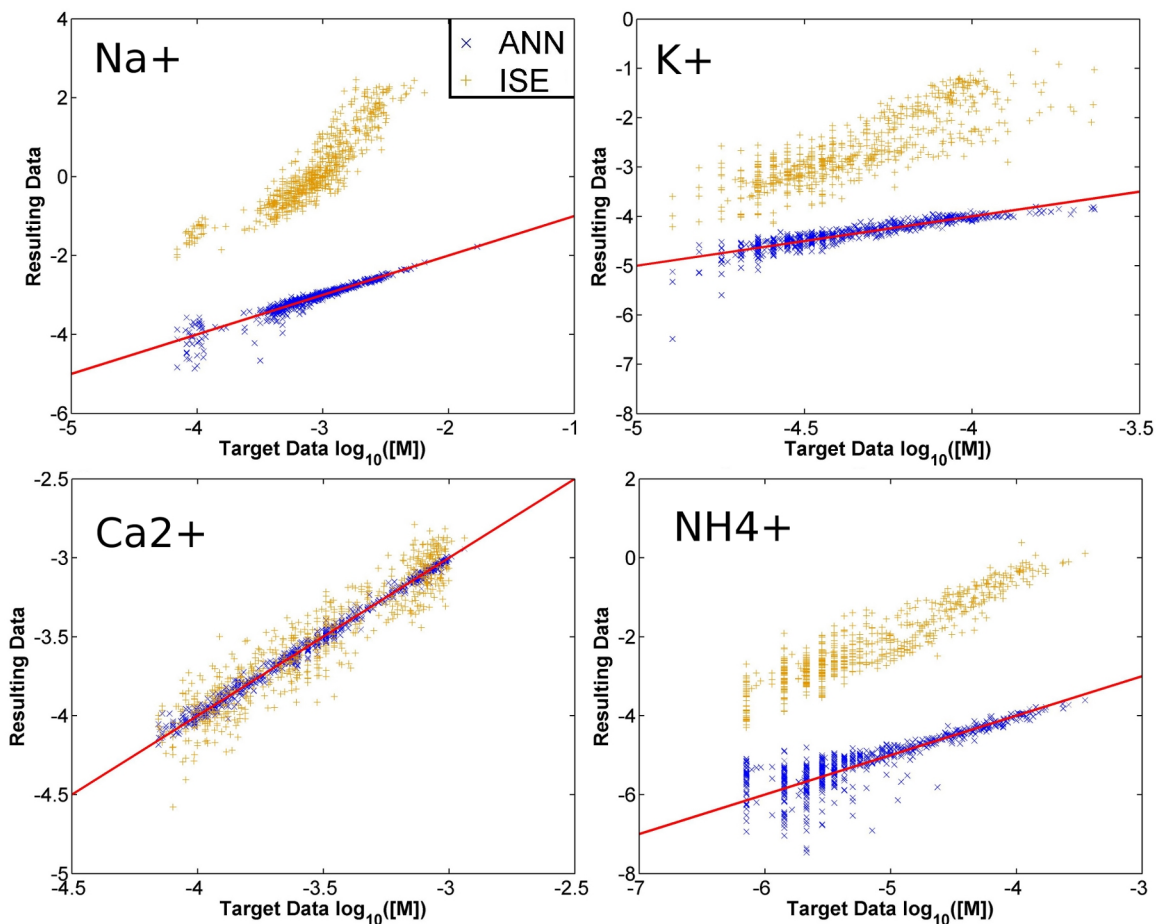


Figure 6-6: Improvement in prediction of MA concentrations using a MA-trained ANN with both conductivity and charge balance constraints. Results are shown relative to use of ISEs as stand-alone (single analyte) sensors.

Table 6.7: NRMSE comparison of MA+TX-trained ANN applied to (a) MA data and (b) TX data.

Target Analyte	MA data	TX data
Na ⁺	0.44	0.097
K ⁺	0.37	0.24
NH ₄ ⁺	0.14	0.094
Ca ²⁺	0.28	0.069
Cl ⁻	0.14	0.095

Subsequently, an ANN was trained using both MA and TX data, the resulting ANN representing the global optimum for the combined data set. Resulting predictions of ion concentrations for both data sets are shown in Fig. 6-7 and Table 6.7.

In this case, the results for TX data are significantly improved for all ions, although results for MA data are degraded by up to a factor of two. The degraded results for the MA data can likely be attributed to their smaller absolute contribution to total MSE, due to lower ionic concentrations in the MA samples (recalling that the TX data had generally

Table 6.5: NRMSE of analyte concentration predictions for MA data made using ISEs only (as stand-alone single-analyte sensors) and by using ISEs processed with the optimal ANN (minimization of NRMSE for these five analytes). Best predictions for each analyte are identified with bold font.

Target Analyte	ISE alone	ANN (base)	ANN (+ CB)	ANN (+CB, +cond)
Na ⁺	6.0·10 ⁴	0.069	0.067	0.12
K ⁺	3.6·10 ²	0.12	0.11	0.24
NH ₄ ⁺	9.2·10 ³	0.26	0.31	0.059
Ca ²⁺	0.36	0.15	0.18	0.073
Cl ⁻	0.15	0.075	0.067	0.059

Table 6.6: NRMSE comparison of MA-trained ANN results when applied to (a) MA data and (b) TX data. Results are compared to (c) NRMSE for ISEs used as single-analyte sensors on TX data, and degradation in estimation performance is represented by (d) the ratio of NRMSE for TX data to NRMSE for MA data.

Target Analyte	ANN, MA data	ANN, TX data	ISEs, TX data	$\frac{NRMSE_{TX}}{NRMSE_{MA}}$
Na ⁺	0.12	0.96	3.4·10 ⁴	8.2
K ⁺	0.24	0.49	1.9·10 ²	2.0
NH ₄ ⁺	0.059	0.86	3.3·10 ⁴	14.6
Ca ²⁺	0.073	0.70	0.43	9.6
Cl ⁻	0.059	0.76	0.14	12.8

higher ionic concentrations).

6.4 Discussion

Results demonstrate that coupled use of a sensor suite and ANN methods can remove bias associated with ISE interferences, improve concentration estimates by up to three orders of magnitude, and predict concentrations of ionic analytes at environmental levels, provided the data are adequately represented by the ANN training data set. Inclusion of TX data, having higher ionic concentrations, allowed the ANN to successfully learn to predict concentrations over a wider range, however this also led to degraded performance at the lowest concentrations. While weighting factors in the current experiments emphasized estimation of ammonium, a critical but relatively less abundant analyte, future work will focus on extending this technique to improve predictions for all ions in low-concentration in any sample.

These results encourage extension of ISE/ANN techniques to encompass the full major ion suite (including magnesium, nitrate, sulfate, and bicarbonate), with the corresponding ANN trained using actual lab and field samples. Such an instrument could serve to estimate the major ion balance of natural waters, thus promoting improved understanding of natural water systems (identification of acidification sources, nutrient flow, or surface water-groundwater exchange) as well as expediting remediation (delineation of affected zones, contamination source ‘fingerprinting’).

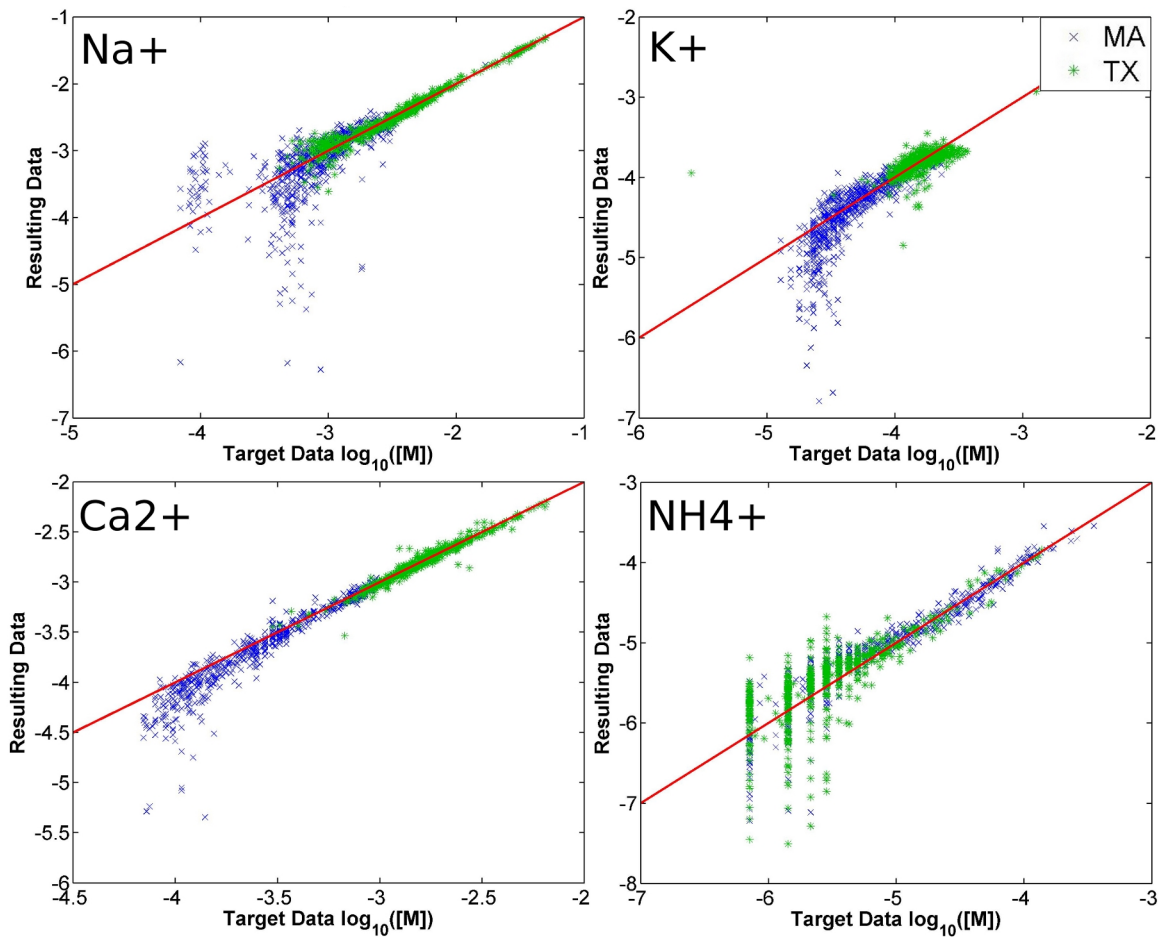


Figure 6-7: Concentration prediction results using a MA+TX trained ANN to process both MA and TX data.

Acknowledgments

This research was supported by: Sea Grant NA10OAR4170086, the National Research Foundation of Singapore (MIT SMART, Agt. dated 9/18/07), a GSA Graduate Student Research Grant, an MIT-Xerox Fellowship, a Martin Foundation Fellowship for Sustainability, an NSF Graduate Research Fellowship, and NSF EAR 0330272.

Chapter 7

Full ionic set for environmental sampling

Abstract

A method is proposed for the training and use of a novel ANN to process raw outputs of a suite of ion selective electrodes for the prediction of ion concentrations. A sample set representative of environmental concentrations and mixes but limited in composition to the analytes of interest (Na^+ , K^+ , NH_4^+ , Ca^{2+} , Mg^{2+} , Cl^- , NO_3^- , SO_4^{2-} in addition to the pH and carbonate systems) is created based on a statistical model constructed using available USGS data for the Northeastern USA. A suite of 14 sensors (11 ion selective measurements, pH, temperature, and electrical conductivity) is used to characterize 75 ion mixtures and 65 single-salt standards; resulting signals were verified and used to train a range of ANN architectures for prediction of analyte concentrations. Inclusion of mathematical constraints based on *a priori* chemical knowledge in a novel extended ANN further improve estimation capabilities. Results are unbiased on all channels (i.e., offset due to interference from other analytes has been removed) and concentrations can be predicted within approximately 20% even at concentrations $<10 \mu\text{M}$ for nitrate species. Useful un-biased estimates are provided even for analytes for which the hardware suite does not contain a specific sensor (SO_4^{2-} , Mg^{2+}), demonstrating that information can be harvested solely from interferences on other electrodes.

7.1 Introduction

The hardware and software developed in previous chapters for use in an ion multi-probe are combined here to provide concentration estimates for the full charge balance of samples representing natural fresh waters. This requires the development of a statistical model for the related concentrations of the analytes of interest in New England waters, random selection of a representative sample suite, characterization of these samples using the given hardware, and training of a wide range of ANN architectures using this collected data. The results of this experiment are validated by measuring the quantified accuracy with which ion concentrations can be predicted in these model environmental samples. While the resulting multiprobe has not yet been tested in-situ, where there are expected to be interferences from e.g., DOC, this work targets accurate prediction of even minor constituents as well as analytes for which no commercial sensors exist (e.g., magnesium, sulfate, bicarbonate). Sup-

plementary information for this chapter, as referenced in the text, is available in Appendix A.

7.2 Sample set creation

Creation of the representative environmental samples was based on historical data for New England waters, downloaded from the USGS database for Water Quality Samples (<http://nwis.waterdata.usgs.gov/usa/nwis/qwdata>). Between 25,000 and 65,000 data points (measurement of a single analyte at a given site and time) were downloaded for each of five states (Massachusetts, Connecticut, Vermont, New Hampshire, and Maine). Data were first sorted and plotted by analyte and subsequently filtered to identify date/time combinations where all analytes of interest were sampled simultaneously. The approximately 200,000 data points yielded 3218 simultaneous measurements of the ion set of interest $\{\text{Na}^+, \text{K}^+, \text{NH}_4^+, \text{Ca}^{2+}, \text{Mg}^{2+}, \text{Cl}^-, \text{NO}_3^-, \text{SO}_4^{2-}, \text{pH}\}$; these simultaneous-set data form the basis of the statistical model discussed in the following section.

7.2.1 Statistical representation of environmental samples

Historical sampling records are available that detail the concentration of many ions at different locations and times. It is thus possible to collect available data for a particular analyte of interest and to create a probability distribution function (PDF) from which one can make inferences about the fraction of waters falling into certain concentration ranges, etc. Examples of this type of plot are shown in Fig. 7-1 for six environmental ions, with separate PDFs shown for each of the five New England states for which data was downloaded. (Note that while the true environmental PDF is a continuous function of concentration, the estimated PDF represents a discrete approximation of this function created by binning available data, and the accuracy of the estimated PDF is thus dependent on sample size.) It is of interest to note that the waters in these five states range from hard to soft (representing the relative contributions of granite, limestone, and other bedrocks in these geologies), and it is thus expected that the PDF will be somewhat bimodal.

This method cannot be used to infer the statistical likelihood of having two different analyte concentrations simultaneously in certain sub-ranges, however, as environmental ion concentrations are not statistically independent, i.e.:

$$p_{x_1, x_2}(x_1, x_2) \neq p_{x_1}(x_1) \cdot p_{x_2}(x_2) \quad (7.1)$$

and by extension for the entire analyte suite:

$$p_{x_1..x_n}(x_1..x_n) \neq \prod_i^n p_{x_i}(x_i) \quad (7.2)$$

Instead to accurately represent the interdependencies of the target ions in natural water samples, it is important to sample directly from the n -dimensional joint PDF $p_{x_1..x_n}(x_1..x_n)$ rather than independently sampling from PDFs for each constituent. A discrete approximation of this joint PDF can be created using the data plotted above in the following manner.

By extension from the 1-D PDFs shown, the joint PDF can be conceptually represented as an n -dimensional matrix of combinations of concentration ranges (bins) for each analyte.

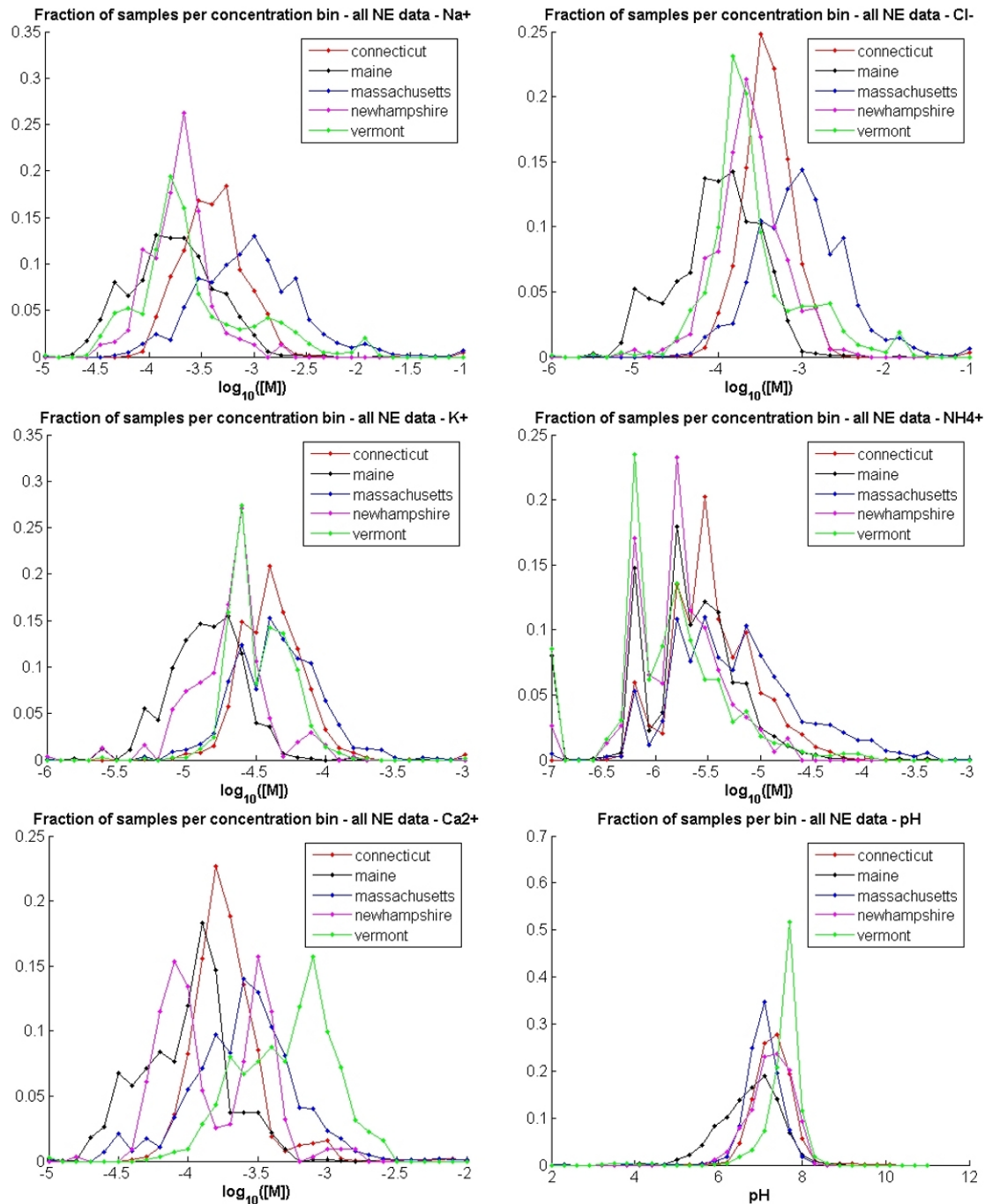


Figure 7-1: One-dimensional probability distribution functions for representative environmental ions, created using archived USGS data for the five states listed. Density values are plotted at bin mid-points.

Each available simultaneous set of data is indexed (the n -D index corresponding to bin number on each axis), the total number of samples indexed to each location is counted, and the final distribution is divided by the total number of counts to produce an n -D surface which encloses a hypervolume of 1. This process was followed to create an 8-D joint PDF for

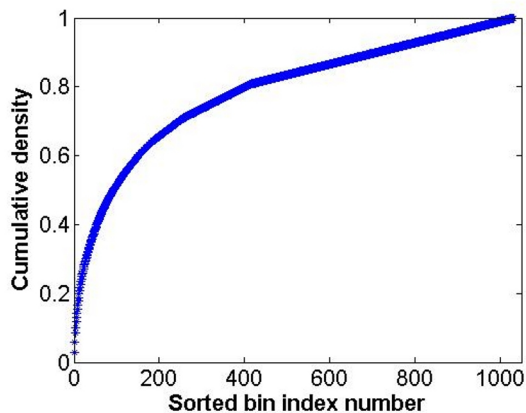


Figure 7-2: Cumulative distribution function for 8-D ion Joint PDF. Independent axis is a sorted bin index, with bins sorted by descending density contribution.

(Na^+ , K^+ , Ca^{2+} , Mg^{2+} , NH_4^+ , Cl^- , SO_4^{2-} , and NO_3^-) with 10 bins (in \log_{10} concentration space) used for each dimension. Approximate ranges represented in the data for each ion are given in Table 7.1, and the bin ranges used to create the joint PDF are provided in Table A.1.

Table 7.1: Approximate concentration ranges for ions of interest in New England waters ($\log_{10}[\text{M}]$).

Analyte	Min.	Mean	Max.
NH_4^+	-6.8	-5.0	-3.2
Ca^{2+}	-4.8	-3.5	-2.2
Na^+	-4.8	-2.7	-1.2
K^+	-5.8	-4.3	-3.2
Cl^-	-5.7	-2.8	-1.3
NO_3^-	-6.8	-5.1	-3.7
Mg^{2+}	-5.7	-3.9	-1.3
SO_4^{2-}	-5.3	-3.8	-1.3

As hypothesized, this approximated 8-D PDF is not uniform nor well represented by a multivariate normal distribution (many analyte distributions have long tails). In fact, the 3218 complete samples fell within only 1032 of the 10^8 bins, of which 152 have five or more samples/bin. The corresponding cumulative distribution function (CDF), created for bins ordered by decreasing density contribution, is shown in Fig. 7-2. This plot demonstrates that only approximately 400 of the 8-D bins are required to account for 80% of the original environmental water samples, i.e., these environmental characteristics are highly covarying.

The approximate joint PDF was then weighted independently per dimension to take into account the relative uncertainty of measurements for each analyte given the hardware to be used, i.e., analyte concentrations for which ISE response would be lower or analytes for which ISE selectivity would be worse were relatively magnified in statistical density. However, because most ion concentrations fell within the linear portion of the response curve, or were consistently in the non-linear portion of the curve, this magnification did not greatly alter the shape of the PDF. The original environmental joint PDF was thus used

for the subsequent selection of training samples.

7.2.2 Selection of training samples

A set of 75 environmental sample mixes was selected randomly from the joint PDF following the methodology for discrete random variables given in [177], Section 2.2. For each sample, a uniform random number on the interval [0,1] is generated and compared to the CDF; the PDF bin corresponding to the density contribution causing the CDF to surpass this value is selected. Concentrations for each ion are then chosen independently from within each corresponding bin (assuming a uniform distribution per bin). Final mix compositions are provided in Table A.2 and Fig. A-1.

In addition to the 75 mix samples, 5 single-salt calibration sets (0.1 μM - 100 mM) were used to independently quantify response of each ISE to each ion. The five calibration sets, chosen to represent each of the ions of interest at least once, were KNO_3 , Na_2SO_4 , $\text{Mg}(\text{NO}_3)_2$, NH_4Cl , and CaCl_2 .

7.2.3 Creation of ion-mix solutions

A set of 16 aqueous standards (Table 7.2) were used to create the ion mixes. Except for the HCl standard which was diluted from a more concentrated aqueous standard, all standards were created using reagent grade salts, dried overnight at 55°C if anhydrous or purchased new for hydrated salts, and weighed out using an Ohaus Precision Standard TS4KD balance. Salts were dissolved in Millipore Milli-Q water ($18.2 \text{ M}\Omega\cdot\text{cm}^{-1}$) and diluted to the appropriate volume (typically 2 L) in a class A volumetric flask. Glass and plasticware used in this process were first acid washed for at least 24-hours in 10% HNO_3 and rinsed 7–10 times in Milli-Q water.

Table 7.2: Salt solutions used in creation of ion mix samples (all standards at 100mM except for $\text{Ca}(\text{OH})_2$ and MgCO_3 which were 20mM and 1.2mM respectively).

NaCl		Na_2SO_4		Na_2CO_3
KCl	KNO_3			K_2CO_3
CaCl_2			$\text{Ca}(\text{OH})_2$	
MgCl_2	$\text{Mg}(\text{NO}_3)_2$	MgSO_4		MgCO_3
NH_4Cl				
HCl	HNO_3	H_2SO_4		

A custom Matlab script was used to calculate the volumes of each standard required to match the target ion concentrations for each of the 75 mixes. Specified volumes were added to Milli-Q water, diluted to 2 L in a class A volumetric flask, well mixed, and then transferred to 2 L LDPE bottles (acid cleaned and rinsed using the method specified above, after which they were stored filled with Milli-Q water until use, i.e., 1–4 months). Solutions were not pH adjusted or specifically managed with respect to equilibration with atmospheric CO_2 during this process, although calculations at the time of sampling demonstrated that exchange with atmospheric CO_2 could be virtually neglected.

Propagated errors due to weighing ($\pm 0.01 \text{ g}$) and standard/sample creation ($\pm 0.5 \text{ mL}$ flask accuracy) estimate maximum concentration errors in the final 75 samples at $\pm 0.8 \mu\text{M}$ (with highest errors expected for salts with low molecular weight or ions at low concentrations).

7.3 Sample measurement

Sample measurement with the electrode array required the following preparation steps:

- Condition hardness ISE overnight in CaCO_3 solution ($\sim 10^{-2}$ M)
- Condition Orion Na^+ ISE overnight (and when not being used) in commercial storage solution¹
- Condition SO_4^{2+} ISE for ~ 1 hour before sampling (Na_2SO_4 solution at $\sim 10^{-2}$ M)
- Condition all ELIT ISEs for ~ 10 minutes before sampling (~ 1000 ppm solutions)
- Condition Thomas Carbonate ISE for ~ 10 minutes before sampling (manufacturer specifications of “lowest concentration being measured”)
- Immediately before sampling, rinse all sensors with Milli-Q water, pat dry with Kim-wipes, and install in ELIT electrode head / custom mounting plate

Sample measurement procedure was identical to that described in Chapter 6, however the full set of hardware described in Chapter 3 was used. Custom LabView software recorded ISE potentials at approximately 1 Hz, and each sample time series was recorded for approximately 6.5 minutes. Seven replicates of each sample were measured. For single-salt standards, measurements were taken from low to high concentrations, with the full suite completed in a single day. For salt mixes, measurements were done in the order of sample number (i.e., an arbitrary order relative to concentrations), and approximately 15-20 could be measured in a single day. After ISE data was logged, each sample/replicate was also characterized using two electrical conductivity meters (Amber Science Model 604; VWR Handheld 21800-012 manufactured by Control Company) and a temperature sensor; these measurements were not taken simultaneously with the ISE measurements to minimize interference in ISE signals due to currents induced during EC measurement. EC meters and pH ISE were calibrated with commercial calibration fluids ($0.73\text{-}10,000 \mu\text{Siemens}\cdot\text{cm}^{-1}$ at 25°C ; pH 4, 7, 10) at the end of the sampling period (approximately two weeks). When compared with previous calibrations of the pH electrode, drift in this signal was determined to be within the uncertainty of the linear calibration.

7.4 Neural network training set

Time series data from ISE sample/standard measurements were processed using the methodology proposed in Chapter 4 to identify ‘steady state’ potentials. Data from the first of seven replicates was discarded (generally identifiable as an outlier), and the mean of the remaining six replicate values was recorded. Problems were identified in several samples (e.g., discontinuities in LabView sampling or disturbance of hardware during measuring) that prevented the automated algorithm from successfully identifying the steady state values; in these cases, the time series was processed manually following the same principles outlined in Chapter 4. Figures 7-3 and 7-4 show measured mean values plotted against target concentration and demonstrate the degree to which interference is experienced on each of the ISE channels.

The neural network input data set was comprised of the ISE responses for the 75 mixes (base case) or 75 mixes plus single-salt standards (extended case), after removal of any problem samples/signals as noted here. The CO_3^{2+} ISE reached a steady potential for fewer than 5 of the mixes and was omitted from subsequent experimental stages. Several

¹Orion 841101 Na^+ Electrode Storage Solution: 5M NaCl, 0.08M NH_4OH , 0.08M NH_4Cl

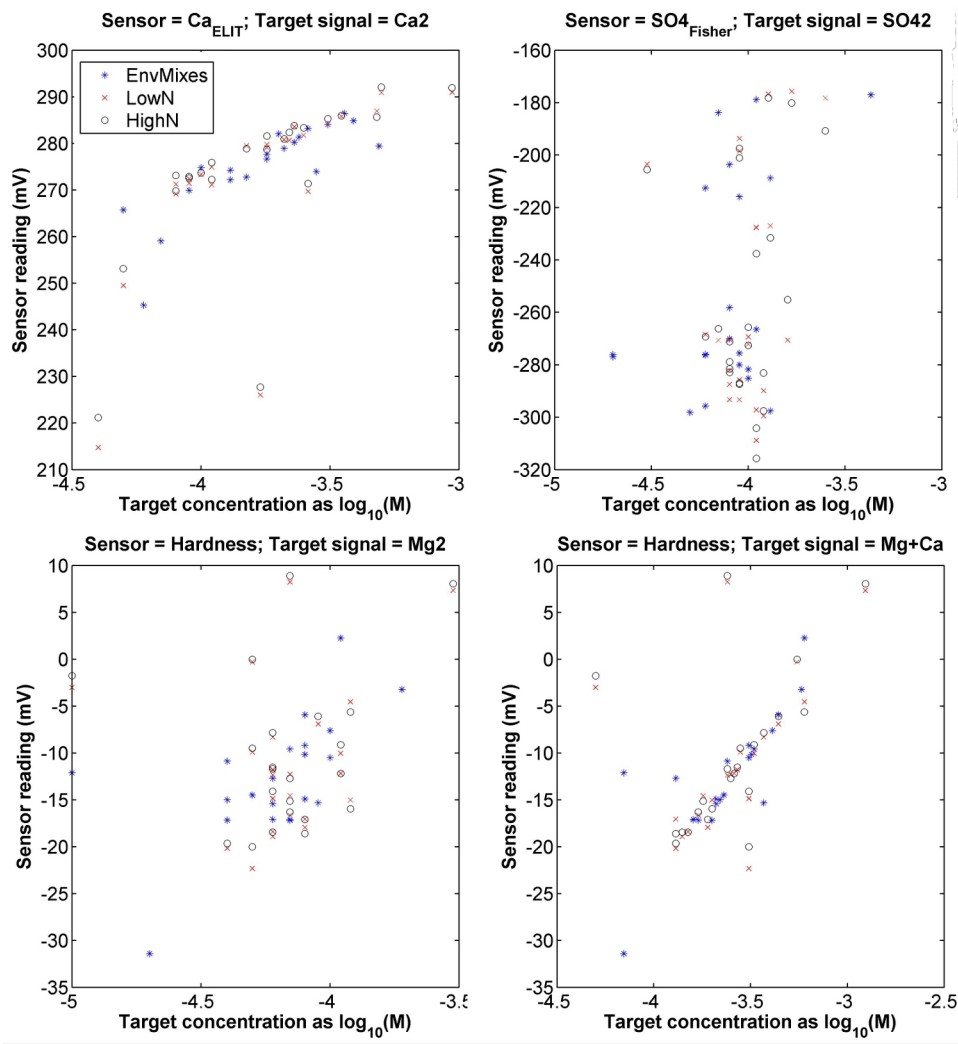


Figure 7-3: Mean response of divalent ISEs as a function of primary analyte concentration.

mix standards were visually identified as potential outliers (i.e., due to visual precipitation (increased turbidity) of salts in the bottle overnight or anomalously low/anomalously high measured conductivity values). These samples (numbers 4, 7, 22, 33, and 58; 17; 48 and 73) were included in training algorithms originally but later removed to improve the overall fit. Of the 65 single-salt standards, 42 had corresponding steady state values identified for the entire ISE set (excepting the carbonate electrode). The remaining 23 (mostly low concentration standards) were omitted from the input data set. Finally, training was found to improve when the highest of the single-salt standards were omitted, likely because these concentrations (100 mM) were significantly higher than those for the mix samples. This resulted in a total of 67 mix samples and 31 single-salt standards making up the ANN input data set. All results given in this chapter are for the fully pruned set described here.

Electrical conductivity and temperature signals were also included as input data. Results were not substantially seen to improve with inclusion of both EC signals, so only the Amber EC meter data were used in the final set. In total, this method resulted in an input data set with 13 signals (11 ISEs, EC, Temperature) and 98 samples.

The target output set was constructed for these 98 samples for up to 19 outputs, detailed

Table 7.3: Required (for calculation of chemical constraints) and additional (quantities that can be calculated or inferred given provided information) target outputs for the neural network architecture.

Ions	Required		Additional	
	Carbonate system	pH system	Ions	Other
NH_4^+	HCO_3^-	H^+	HSO_4^-	Ionic Strength
Ca^{2+}	CO_3^{2-}	OH^-	TOT_SO4	γ_1
Na^+			TOT_NH4	γ_2
K^+				pH
Cl^-				
NO_3^-				
Mg^{2+}				
SO_4^{2-}				

in Table 7.3. Required targets are necessary for accurate computation of the electrical conductivity while additional targets simply provide the user with more information about the overall system. Assignment of values to the neural network target set was done in several stages:

1. Electrical conductivity signals were corrected for temperature (using coefficient from [178]) and calibration.
2. Data from the pH ISE was converted to pH using the calibration developed above.
3. Concentrations for strong acid/strong base ions (all except NH_4^+ and SO_4^{2+}) were retrieved from the original sample specifications (Section 7.2.2).
4. Speciation of pH-dependent analytes ($\text{NH}_3/\text{NH}_4^+$ and $\text{HSO}_4^-/\text{SO}_4^{2+}$) was calculated based on the measured pH.
5. Carbonate system concentrations (along with H^+ and OH^-) were calculated based on the measured pH and assuming that only limited carbon exchange with the atmosphere had occurred since creation of the samples.
6. Ionic strength and activity coefficients were calculated based on the full ion distribution. (Note that this occurs with the previous item in a simultaneous solver procedure to assure values are mutually consistent.)

Calculations demonstrated that $[\text{HSO}_4^-]$ was zero on most channels and was expected to contribute less than 0.5% of the magnitude of EC for any sample. Because of the negligible additional information available in this signal, and because of the difficulties encountered in training a network for which a target signal has very little variance, HSO_4^- was not included as a target signal.

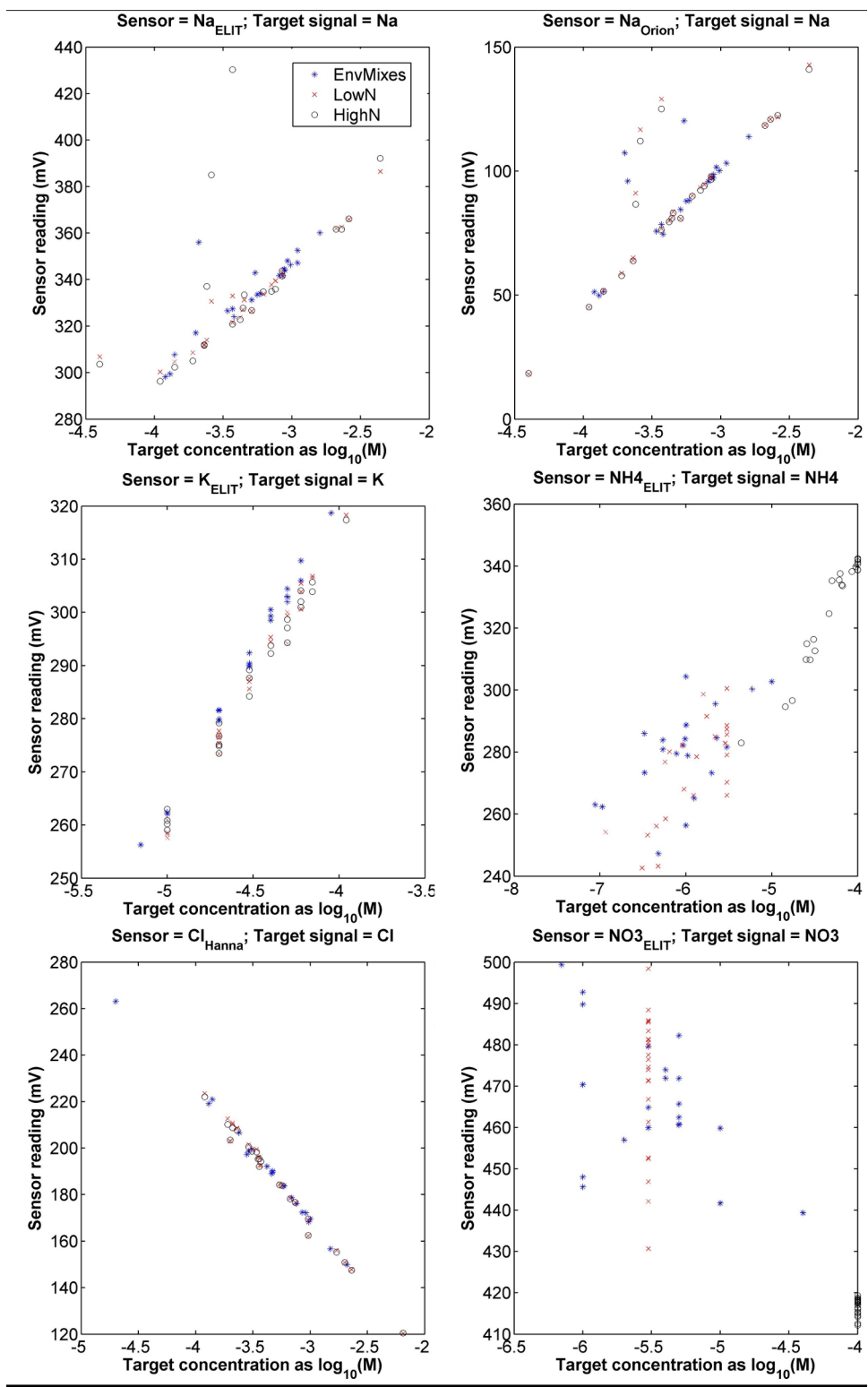


Figure 7-4: Mean response of monovalent ISEs as a function of primary analyte concentration.

At this juncture, the expected electrical conductivity based on these ion sets was calculated and compared to the measured (temperature corrected and calibrated) EC. Good agreement was found with the Amber EC meter measurements, as is demonstrated in Fig. 7-5 for the mix sample data. As such, the Amber EC measurements were used as the target EC output for the neural network.

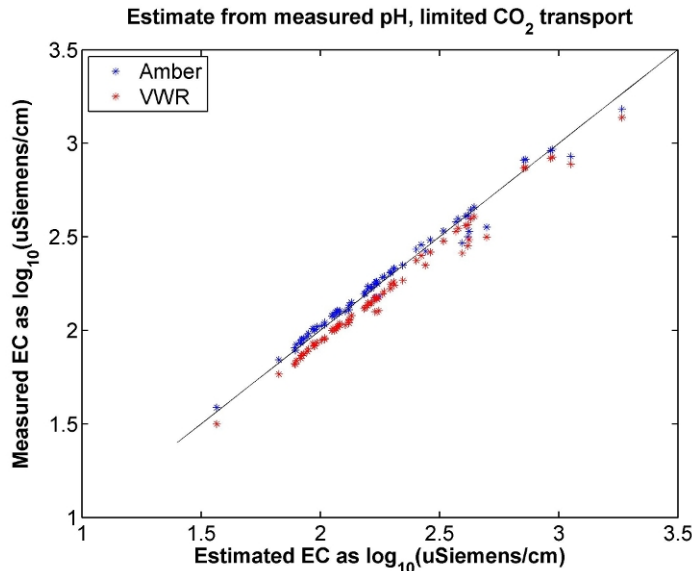


Figure 7-5: Estimated EC based on ion makeup of water samples as compared to measured EC (temperature corrected and calibrated). Measurements from VWR meter were highly correlated with but consistently lower than those produced by the Amber meter; it is expected this is related to the built in temperature correction software which was disabled for these experiments but does not always completely disable correctly.

Finally, the target charge balance (CB) was determined following the method proposed in Appendix 5, i.e., alkalinity less carbonate alkalinity or $\mathbf{CB} = [\mathbf{OH}^-] - [\mathbf{H}^+]$.

7.5 Neural network architectures tested

A range of neural network architectures was proposed and trained using the data sets described above. Design of the architecture and test set took place at two levels, which I will refer to here as ‘external’ (layout of constraint layers, number of outputs used, whether target data is log-transformed, etc.) and ‘internal’ (setting of individual parameters to control the ANN training, number of hidden layers or nodes per hidden layer, etc.). The full range of ‘internal’ options was explored for each combination of ‘external’ options to ensure that the optimal combination was found. For clarity, these two design stages are discussed here independently. Implementation of this functionality was done using the Matlab Neural Network Toolbox V7.0 (R2010b), and the key scripts can be found in Appendix B.

7.5.1 Internal ANN parameter space

The parameter space explored for the internal ANN settings was informed by the results of Chapter 6. Table 7.4 shows the values for each parameter; in cases where previous

experience had already identified the superior parameter choice, only the single selected value is listed.

Table 7.4: Range of parameters explored for design of neural networks.

Parameter	Values
Hidden layer size	6, 9, 12, 15, 18
μ	0.001, 0.1
μ_{inc}	1.5, 10, 50
μ_{dec}	0.1, 0.5, 0.9
Training goal	10^{-6}
Max. # hidden layers	3
Max. # of epochs	10,000
Hidden layer transfer function	tansig
Output layer transfer function	purelin
Training fraction	0.7
Validation fraction	0.15
Testing fraction	0.15

This set of parameterizations resulted in creation and training of 2790 independent ANNs for each of the architectures specified in the following section. Resulting ANNs were relatively ranked on several metrics, including MSE (mean squared error), NRMSE (normalized root mean squared error), MRE (mean relative error) (see Table 7.5), and these same metrics calculated using only the 8 target ion concentration results and/or using only the mix data. Specific methods used to select the ‘optimal’ network are discussed further in the Section 7.6.

7.5.2 External ANN parameter space (ANN architecture)

Design of the ANN architecture was done independently of selection of internal parameters. Architecture decisions determine the number of network inputs and outputs, the number of output layers, and the form of the data used to train the network. They may also include controlling the number of samples used for training, re-balancing the weighting of output errors, or altering training data to take uncertainties in chemical measurements into account. The primary architecture options explored are listed in Table 7.6, and implementation of these options follows recommendations in Appendix 5.

Use of the ‘logsin’ transfer function between the last hidden layer and the output layer (also requiring use of the range [0,1] for the target mapminmax function) was explored as an alternative to log-transformation of the data to ensure that all outputs are non-negative (and thus physically relevant in the case of concentration values). When compared, however,

Table 7.5: Formulae for metrics used to rank ANN results, including MSE (mean squared error), NRMSE (normalized root mean squared error), MRE (mean relative error).

Metric	MSE	NRMSE	MRE
Formula	$\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x})^2$	$\frac{1}{\bar{x}} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x})^2}$	$\frac{1}{n} \sum_{i=1}^n \frac{x_i - \hat{x}}{x_i}$

Table 7.6: Range of parameters explored for design of neural network architecture.

Parameter	Values
Output layer design	One output layer of 12 neurons (only 12 required ions) One output layer of 19 neurons (all required and additional targets) Two output layers: (1) 12 ions and (2) CB constraint Two output layers: (1) 12 ions and (2) EC constraint Three output layers: (1) 12 ions, (2) CB constraint, (3) EC constraint
Training data set	Environmental mix data only (67 samples) Environmental mix and single-salt data (98 samples)
Target normalization	Targets as absolute concentration values Targets as log-transformed concentration values
Error weighting	Weight errors on nitrogen channels (NH_4^+ , NO_3^-) more heavily than other outputs Weight errors on nitrogen channels and constraint layers (EC and/or CB) more heavily than other outputs

results indicated that the log-transformation produced significantly better results and was thus worth the increased complexity.

7.6 Results

Ion concentrations were predicted using (1) ISE calibration curves, assuming ISEs were used as stand-alone sensors (i.e., interferences were not taken into account) and (2) approximately 90,000 different ANNs based on the parameter sets outlined above. The ISE-only results are presented first as a ‘baseline’ case, after which the ANN results are evaluated and compared to the baseline to ascertain level of improvement.

7.6.1 ISE-only concentration prediction

Based on the single-salt standard measurements, calibration curves were created for each ISE relative to its primary ion (or secondary, in cases such as Mg^{2+} where no ISE was available for a given target ion). These curves, plotted along with response to interfering ions, are provided in Figs. A-2 – A-4, and corresponding parameters of the linear fits are given in Table A.3. These calibrations were then used to estimate ion concentrations from ISE responses. Scatter plots showing the resulting ion concentration predictions for the mix data as a function of the target concentrations are given in Figs. 7-6 and 7-7. Note that **all** sample concentrations for sulfate fall below the detection limit for the Pb-specific electrode as used for measuring sulfate; in addition, over half of the concentrations for the nitrate ions fall below the respective detection limits for the appropriately matched selective electrodes.

While certain ISEs are relatively selective and predict concentrations close to the correct values (Cl^- , K^+), most experience a significant amount of interference from the other ions in solution. In particular, errors tend to be biased, as is demonstrated by the displacement of the bulk of points from the 1:1 line in a single direction. In some cases the bias of error appears to be a function of concentration, however these trends may be confounded by

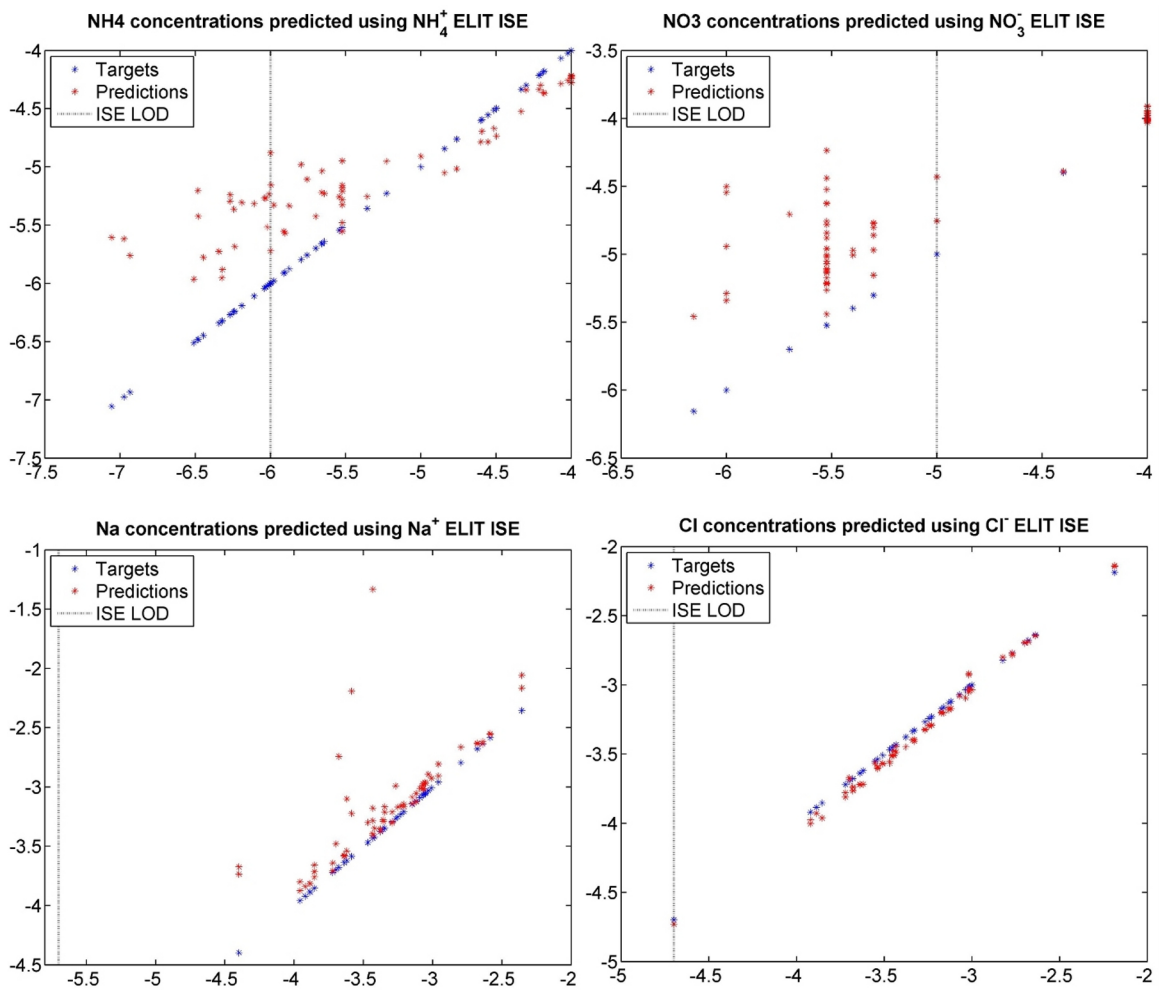


Figure 7-6: ISE-based predictions of ion concentrations (prediction vs. target) for NH₄⁺, NO₃⁻, Na⁺, and Cl⁻. Limit of detection (LOD) is plotted as a vertical dotted line.

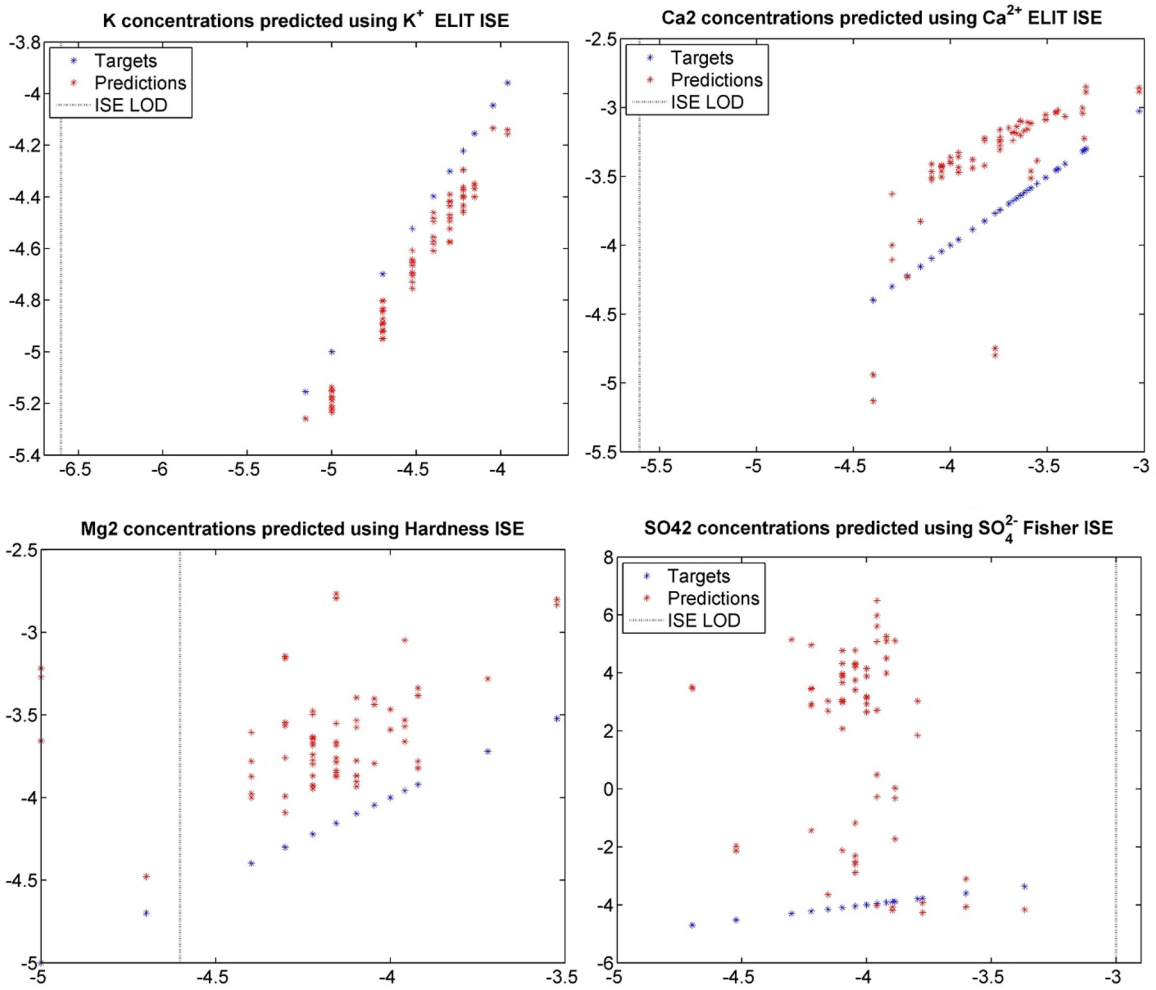


Figure 7-7: ISE-based predictions of ion concentrations (prediction vs. target) for K^+ , Ca^{2+} , hardness, and SO_4^{2-} . Limit of detection (LOD) is plotted as a vertical dotted line.

Table 7.7: Errors for ISE-based ion concentration predictions.

	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}
NRMSE	0.750	1.80	7.42	0.395	0.176	0.363	5.388	$3.9 \cdot 10^9$
MRE	3.31	1.94	2.71	0.323	0.110	3.02	4.94	$7.99 \cdot 10^8$

covarying properties, e.g., a similarly varying ionic strength. These concepts are further illuminated in Fig. A-5. The concentration predictions were evaluated using three metrics (MSE, NRMSE, MRE), and the two unitless metrics are provided in Table 7.7. The two relatively-selective ISEs show a mean relative error of 11% and 32% respectively, however all other channels have over 100% error, and the sulfate predictions are meaningless since all sample concentrations were below the detection limit for this ISE.

7.6.2 ANN suite evaluation

Each ‘External’/‘Internal’ parameter set combination produced a total of 2790 trained ANNs. It was then necessary to select the best from among these trained networks, requiring use of an appropriate ‘goodness’ metric. Several such metrics were considered, including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean relative error (MRE), and the normalized versions NMSE, NRMSE, and NMAE. (MRE is already a unitless metric.) For each network, the metrics were applied independently to each output signal, and the errors were summed. To determine which metric would most accurately encapsulate our standard for goodness (a balance of accurate estimation of the low-concentration nitrogen species and accuracy on the remaining ion channels), the resulting metrics were plotted against comparable metrics for the nitrogen species output signals. A subset of these plots are provided in Fig. A-6, and the corresponding correlation coefficients are given in Table A.4. Interpretation of these data led to selection of total NRMSE (calculated using only the 8 target ion output signals for the mix data samples) as the measure of goodness for ANN quality, and parameterizations for the best ANNs chosen for each External set are given in Table 7.8. Total MRE was also shown to be a goodness metric of comparable quality; because these two metrics identified substantially different optimal parameterization for cases where single-salt solutions were not used in training, corresponding results using the MRE metric are presented in Table A.5.

NRMSE and MRE were calculated for individual ion concentration outputs for each of the network architectures listed in Table 7.8; results are given in Tables 7.9 - 7.12. The minimum NRMSE and MRE for each channel are highlighted in the tables to provide the reader with a sense of the trade-offs required in choosing an optimal network configuration from among these. It is rare that a single configuration produces optimal results *on all channels simultaneously*, and thus one must inherently weigh trade-offs between accuracy (or precision) on one channel against others. (Note that it is possible that a larger training set could improve results on all channels, but this is beyond the scope of this present work.) In this case, the minimum sum of NRMSE for the 8 target ions was used to select the optimal ANN; the best among those listed in each table is highlighted, with the optimal overall network highlighted in gray in Tables 7.8 and 7.9. It is significant to note that networks trained to concentration data (i.e., without log-normalization, Tables 7.9 - 7.10) produced lower total NRMSEs by a factor of 2–4 compared to other configurations, though this was not true of the corresponding MREs. Interestingly, while most ions have errors in roughly

Table 7.8: Parameterizations for best ANN (chosen using NRMSE metric) as a function of ‘External’ architecture.

Architecture			Parameterization			
Outputs	Data	Normalization	Hidden Layers	μ	μ_{dec}	μ_{inc}
12 ions	mixes	none	[9,9]	0.1	0.9	50
19 params	mixes	none	[12,15]	0.1	0.1	50
12 ions+CB	mixes	none	[9,18,12]	0.001	0.5	50
12 ions+EC	mixes	none	[15,15,15]	0.001	0.1	50
12 ions+CB+EC	mixes	none	[6,18,15]	0.001	0.1	10
12 ions	all	none	[9,15,12]	0.1	0.5	50
19 params	all	none	[12,18,18]	0.001	0.9	50
12 ions+CB	all	none	[9,12]	0.1	0.1	10
12 ions+EC	all	none	[12,18,12]	0.001	0.1	10
12 ions+CB+EC	all	none	[18,15]	0.1	0.9	1.5
12 ions	mixes	\log_{10}	[18,12,18]	0.1	0.1	50
19 params	mixes	\log_{10}	[18,12]	0.1	0.5	50
12 ions+CB	mixes	\log_{10}	[12,9]	0.1	0.9	1.5
12 ions+EC	mixes	\log_{10}	[9,18,9]	0.1	0.9	1.5
12 ions+CB+EC	mixes	\log_{10}	[15,18,12]	0.001	0.9	10
12 ions	all	\log_{10}	[18,15,15]	0.001	0.1	50
19 params	all	\log_{10}	[12,18,12]	0.001	0.5	50
12 ions+CB	all	\log_{10}	[9,15,9]	0.1	0.1	1.5
12 ions+EC	all	\log_{10}	[15,12,6]	0.001	0.1	10
12 ions+CB+EC	all	\log_{10}	[18,18,12]	0.1	0.9	1.5
Errors on constraints weighted more heavily						
12 ions+CB	mixes	none	[12,12,15]	0.1	0.1	50
12 ions+EC	mixes	none	[9,15]	0.1	0.9	1.5
12 ions+CB+EC	mixes	none	[15,15,12]	0.001	0.9	1.5
12 ions+CB	all	none	[6,18]	0.1	0.1	1.5
12 ions+EC	all	none	[9,18,18]	0.1	0.9	10
12 ions+CB+EC	all	none	[15,15,15]	0.1	0.1	50
12 ions+CB	mixes	\log_{10}	[15,18]	0.1	0.9	50
12 ions+EC	mixes	\log_{10}	[12,15,6]	0.1	0.1	50
12 ions+CB+EC	mixes	\log_{10}	[9,12,15]	0.001	0.1	50
12 ions+CB	all	\log_{10}	[18,15,18]	0.1	0.1	1.5
12 ions+EC	all	\log_{10}	[12,12,18]	0.001	0.5	50
12 ions+CB+EC	mixes	\log_{10}	[9,18,9]	0.1	0.1	1.5

the same ranges, this bifurcation also corresponds with a clear trade-off between accuracy in predicting NH_4^+ and Mg^{2+} ; architectures trained to concentration data have lower Mg^{2+} MRE and higher NH_4^+ MRE while architectures trained to log-transformed data have inverse results. Inclusion of constraints systematically improves predictions for cases where the full data set (mixes and single-salt standards) was used for training, though it does not have the same effect for the mix-only cases, and weighting of the corresponding constraint error channels does not further improve results (see Fig. 7-8). Finally, architectures trained solely on the mix data produced results with lower NRMSE than corresponding networks trained with both mix and single-salt data. This was true *even when total NRMSE values used to identify optimal parameterizations were calculated using both mix and single-salt data sets*, implying that the less representative training data does not necessarily provide clear ‘piecewise’ information to the network about ISE calibration (e.g., how each electrode responds to a solution comprised of only NaCl) as had been expected. Corresponding results using the MSE metric are provided in Tables A.6 - A.7 and Fig. A-7.

Table 7.9: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **concentration values of mix data**; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	$\sum \text{Err}$
no	12	0.122 (80.1)	0.232 (25.7)	0.136 (27.1)	0.099 (7.6)	0.153 (28.4)	0.038 (37.2)	0.307 (34.2)	0.294 (23.4)	1.381 (263.7)
no	19	0.153 (184.9)	0.156 (18.7)	0.119 (21.2)	0.104 (9.4)	0.126 (21.4)	0.055 (39.2)	0.292 (27.3)	0.238 (31.7)	1.243 (353.8)
no	12+CB	0.147 (137)	0.266 (30.2)	0.164 (25)	0.142 (15)	0.126 (26.6)	0.101 (43.6)	0.298 (32.7)	0.32 (36.6)	1.564 (346.7)
no	12+EC	0.275 (111.5)	0.197 (19.4)	0.141 (17.4)	0.103 (7.8)	0.082 (17.5)	0.064 (22.8)	0.252 (23.9)	0.464 (21.8)	1.578 (242.1)
no	12+CB+EC	0.081 (54.3)	0.148 (20.2)	0.145 (16.9)	0.077 (8.8)	0.101 (23.4)	0.037 (21.4)	0.219 (20.1)	0.216 (21.4)	1.024 (186.5)
yes	12+CB	0.138 (190.6)	0.197 (25.7)	0.164 (31.2)	0.149 (15.7)	0.141 (28.7)	0.118 (66.7)	0.307 (32.5)	0.272 (35.9)	1.486 (427)
yes	12+EC	0.202 (175.9)	0.152 (18.4)	0.108 (18)	0.133 (13.1)	0.166 (28.9)	0.069 (40.6)	0.232 (23.5)	0.363 (33.2)	1.425 (351.6)
yes	12+CB+EC	0.177 (97.6)	0.198 (26.2)	0.186 (20.6)	0.074 (8.2)	0.187 (26)	0.141 (59.4)	0.241 (25.4)	0.269 (20.4)	1.473 (283.8)

Table 7.10: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **concentration values of mix and single-salt data**; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	$\sum \text{Err}$
no	12	0.139 (79.2)	0.201 (23.2)	0.178 (33.8)	0.107 (10.4)	0.195 (34.5)	0.098 (78.2)	0.31 (30)	0.683 (51.1)	1.911 (340.4)
no	19	0.158 (100.3)	0.186 (21.5)	0.287 (48.1)	0.14 (11.1)	0.153 (26.4)	0.135 (85.1)	0.284 (35.5)	0.681 (80)	2.024 (408)
no	12+CB	0.111 (87.2)	0.221 (26.5)	0.166 (21.3)	0.08 (7.9)	0.145 (19.3)	0.098 (65.5)	0.296 (27.5)	0.517 (57.8)	1.634 (313)
no	12+EC	0.119 (90.1)	0.16 (18.6)	0.246 (43.4)	0.168 (15.1)	0.174 (29)	0.072 (39.8)	0.254 (18.7)	0.658 (65.2)	1.851 (319.9)
no	12+CB+EC	0.201 (142.9)	0.221 (16.2)	0.149 (27.1)	0.158 (11.5)	0.087 (30)	0.082 (50.8)	0.266 (17.4)	0.466 (49.5)	1.63 (345.4)
yes	12+CB	0.209 (60.6)	0.205 (18.2)	0.195 (43.7)	0.095 (10.3)	0.141 (22.7)	0.057 (19.7)	0.36 (24.7)	0.734 (80.9)	1.996 (280.8)
yes	12+EC	0.122 (46.8)	0.201 (19.0)	0.350 (50.8)	0.130 (11.1)	0.176 (27.5)	0.060 (28.9)	0.258 (28.7)	1.148 (103.1)	2.445 (316.0)
yes	12+CB+EC	0.134 (204.1)	0.234 (26.6)	0.173 (33.7)	0.084 (8.5)	0.265 (48.1)	0.047 (28.6)	0.249 (21.3)	1.140 (91.0)	2.326 (461.9)

Table 7.11: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix data**; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻	∑Err
no	12	0.309 (40.4)	0.28 (20.4)	0.283 (18.1)	0.173 (14.7)	0.33 (27.6)	0.142 (39.1)	0.35 (24.2)	0.292 (28)	2.159 (212.5)
no	19	0.338 (31.5)	0.164 (13.2)	0.149 (12.5)	0.171 (9.3)	0.159 (16.8)	0.259 (28.5)	0.548 (41.6)	0.201 (19.1)	1.989 (172.5)
no	12+CB	0.383 (41.7)	0.35 (28)	0.554 (30.3)	0.187 (14)	0.5 (35.6)	0.467 (51.6)	0.374 (33.9)	0.457 (33.7)	3.272 (268.8)
no	12+EC	0.333 (25.7)	0.358 (32.1)	0.425 (40.4)	0.288 (22.6)	0.517 (45.8)	0.319 (42.2)	0.434 (47.6)	0.473 (40)	3.147 (296.4)
no	12+CB+EC	0.400 (63.8)	0.297 (27.5)	0.567 (41.3)	0.433 (23.4)	0.582 (37.0)	0.380 (30.4)	0.412 (39.9)	0.326 (32.2)	3.398 (295.5)
yes	12+CB	0.271 (40.2)	0.332 (26.4)	0.470 (33.4)	0.230 (17.4)	0.519 (44.8)	0.251 (31.2)	0.433 (44.9)	0.430 (35.7)	2.936 (273.9)
yes	12+EC	0.431 (31.7)	0.305 (26.2)	0.454 (52.3)	0.246 (17.6)	0.348 (33.3)	0.318 (15.6)	0.416 (36.7)	0.365 (29.4)	2.883 (242.8)
yes	12+CB+EC	0.392 (43.8)	0.471 (35.0)	0.580 (44.5)	0.364 (33.5)	0.526 (39)	0.206 (28.7)	0.438 (41.9)	0.456 (36.0)	3.433 (294.2)

Table 7.12: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix and single-salt data**; optimal network (highlighted in left column) selected using the NRMSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻	∑Err
no	12	0.208 (15.8)	0.355 (22.7)	0.351 (30)	0.258 (19.8)	0.545 (24.8)	0.215 (21.3)	0.407 (31.9)	0.533 (48.1)	2.872 (214.4)
no	19	0.202 (34.8)	0.16 (12.8)	0.298 (16.9)	0.25 (15)	0.225 (12.9)	0.539 (24.7)	0.387 (48.2)	0.315 (29.2)	2.376 (194.5)
no	12+CB	0.465 (37.2)	0.787 (56.4)	1.264 (86.9)	0.827 (55.3)	1.188 (80.8)	0.342 (25.9)	0.852 (74.4)	0.915 (80.6)	6.64 (497.5)
no	12+EC	0.429 (26.2)	0.753 (57.1)	1.147 (86.1)	0.782 (51.9)	0.967 (74.8)	0.171 (38.9)	0.756 (65)	1.025 (82.2)	6.03 (482.2)
no	12+CB+EC	0.557 (54.7)	0.639 (39.6)	1.23 (75.8)	0.76 (50.4)	1.148 (66.5)	0.397 (36.9)	0.712 (49.9)	0.92 (75)	6.363 (448.8)
yes	12+CB	0.848 (46.4)	1.103 (79)	1.514 (94.8)	0.994 (76.3)	1.673 (95.3)	0.479 (41.2)	1.034 (92.3)	1.072 (94)	8.717 (619.3)
yes	12+EC	0.516 (37.6)	0.632 (58.7)	1.15 (77.1)	0.72 (48.6)	1.058 (74.9)	0.449 (62.7)	0.728 (71.9)	0.843 (75.3)	6.096 (506.8)
yes	12+CB+EC	0.401 (39.5)	1.041 (56.8)	1.479 (88.5)	0.842 (56.8)	1.675 (87.5)	0.421 (56.9)	0.983 (72.8)	1.073 (91.7)	7.915 (550.5)

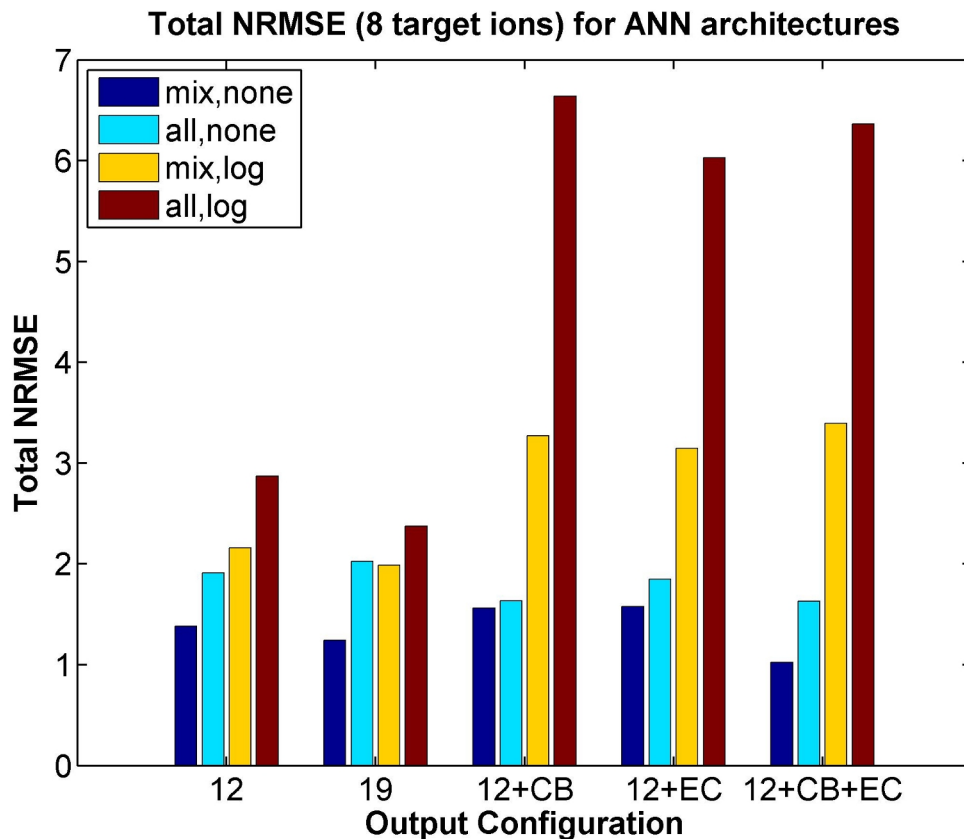


Figure 7-8: Total NRMSE as a function of ANN architecture. Horizontal axis shows output configuration while colored bars represent options for data used in training (mix data only / mix data plus single-salt standards) and data normalization (none / log).

The optimal ANN configuration was then used to predict the individual concentrations of the 8 target ions. Scatter plots of these results – with ISE predictions overlain for reference – are shown in Figs. 7-10–7-13; predictions for the carbonate system are included in Fig. 7-14, and results for the two constraint channels are provided in Fig. 7-15. ANN concentration estimates (blue) generally fall along the 1:1 lines (red), having **successfully removed the bias encountered** with the ISEs as stand-alone sensors. Table 7.13 shows the parameters for linear fits of the ANN-predicted concentrations against the target concentrations. The 95% confidence interval on the slope contains 1 (perfect agreement) for all target analytes except SO_4^{2-} and CO_3^{2-} , for which it is slightly less than 1. Intercepts are only statistically significantly different from zero for Cl^- and SO_4^{2-} . These facts essentially identify ANN estimates as an unbiased estimator (i.e., *accurate*), and therefore both the R^2 and RMSE values provide information about the magnitude of scatter around the targets (*precision*). RMSE values are $< 10 \mu\text{M}$ for all analytes excepting Na^+ , even for those analytes for which we have no specific sensor (Mg^{2+} , SO_4^{2-} , OH^- , and the carbonate species).

For clarity, a comparison of ANN and ISE prediction errors are given in Table 7.14 (note that ISE MRE is presented here as %). As is clear from this data **ANN predictions improve NRMSE by approximately an order of magnitude on most channels and by nearly a factor of five even for K^+** . Sulfate is now predicted reasonably well despite the fact that the ISE calibration is extremely non-linear, selectivity coefficients are

Table 7.13: Parameterization of the linear regression of ANN-predicted concentrations against target concentrations; in nearly all cases slope is statistically indistinguishable from 1 and intercept is statistically indistinguishable from 0.

Analyte	Slope	Intercept	R ²	RMSE
Na ⁺	0.980±0.030	(-0.96±3.54)·10 ⁻⁵	0.985	1.09·10 ⁻⁴
K ⁺	1.005±0.030	(0.57±1.33)·10 ⁻⁶	0.986	2.87·10 ⁻⁶
NH ₄ ⁺	0.996±0.013	(-2.84±5.14)·10 ⁻⁷	0.997	1.77·10 ⁻⁶
Ca ²⁺	0.991±0.045	(0.81±1.26)·10 ⁻⁵	0.967	3.22·10 ⁻⁵
Mg ²⁺	0.931±0.084	(7.23±7.69)·10 ⁻⁶	0.883	1.68·10 ⁻⁵
Cl ⁻	0.985±0.016	(3.87±2.29)·10 ⁻⁵	0.996	7.66·10 ⁻⁵
NO ₃ ⁻	1.000±0.007	(-2.15±4.34)·10 ⁻⁷	0.999	1.38·10 ⁻⁶
SO ₄ ²⁻	0.886±0.091	(1.45±1.08)·10 ⁻⁵	0.852	2.15·10 ⁻⁵
HCO ₃ ⁻	0.950±0.042	(0.20±1.57)·10 ⁻⁵	0.970	4.96·10 ⁻⁵
CO ₃ ²⁻	0.912±0.049	(1.93±2.65)·10 ⁻⁶	0.955	8.46·10 ⁻⁶
H ⁺	1.000±0.019	(0.54±1.20)·10 ⁻⁶	0.994	4.65·10 ⁻⁶
OH ⁻	1.006±0.031	(1.48±3.04)·10 ⁻⁶	0.985	1.07·10 ⁻⁵

Table 7.14: Ion concentration prediction errors for optimal ANN compared to results using ISEs as stand-alone sensors.

	Err Type	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻
ISE	NRMSE	0.750	1.80	7.42	0.395	0.176	0.363	5.388	3.9·10 ⁹
	MRE (%)	331	194	271	32.3	11.0	302	494	7.99·10 ⁶
ANN	NRMSE	0.081	0.148	0.145	0.077	0.101	0.037	0.219	0.216
	MRE (%)	54.3	20.2	16.9	8.8	23.4	21.4	20.1	21.4

not good, and all concentrations fall below the detection limit of the Pb-ISE proposed for measurement of sulfate. Concentrations of both nitrate ions are successful un-biased even for points below the detection limits of their respective electrodes. These gains were, however, incurred with an increase in MRE for Cl⁻ despite the corresponding decrease in NRMSE. This signals a case where fewer predictions are extreme outliers but the variance around the 1:1 line is increased. As expected, absolute error increases with decreasing concentration, though Fig. 7-9 shows that the error is generally still scattered about zero (i.e., unbiased).

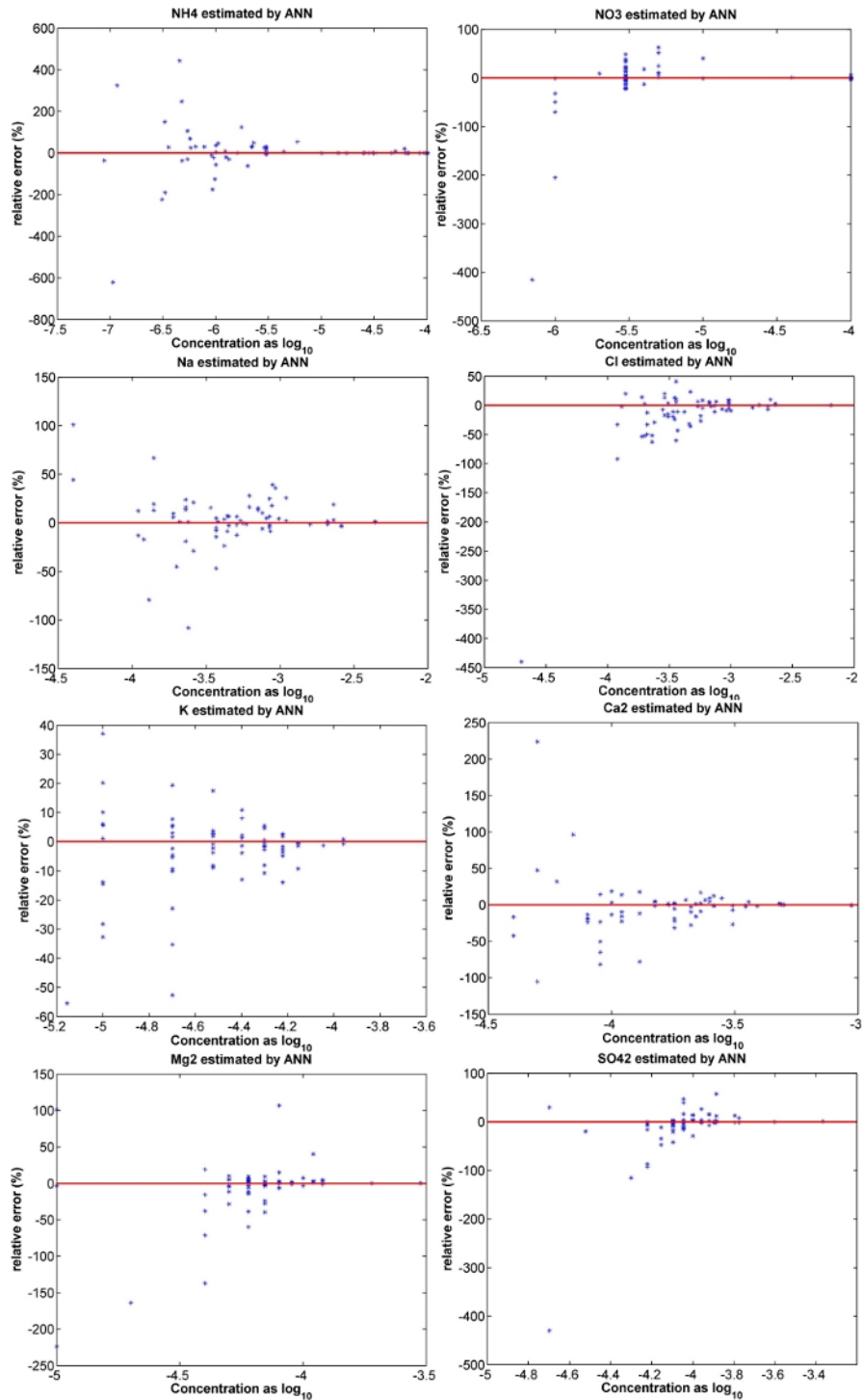


Figure 7-9: Relative percent error for optimal ANN predictions as a function of analyte concentration. Results are shown for mix data only.

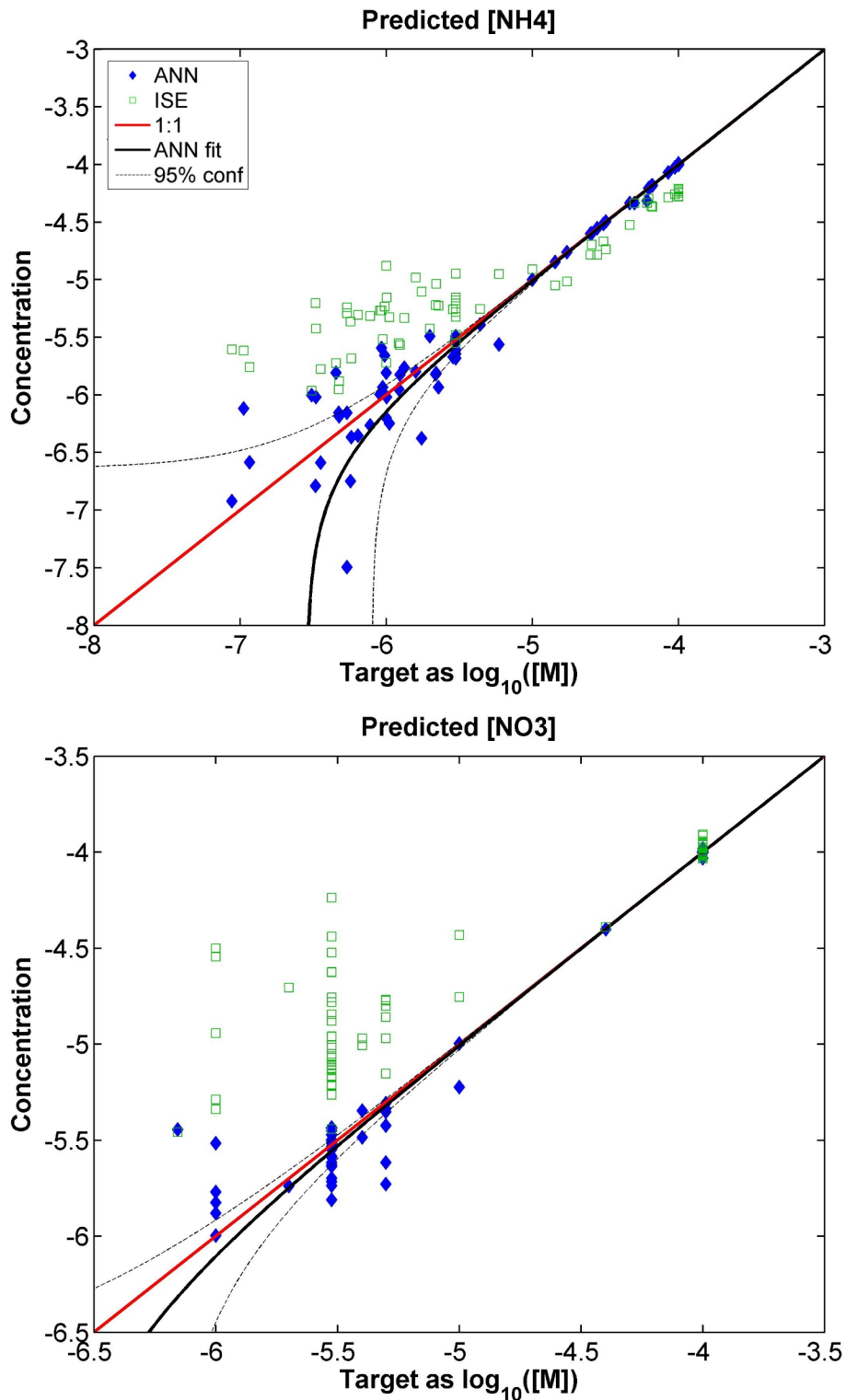


Figure 7-10: Scatter plots of nitrogen ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.

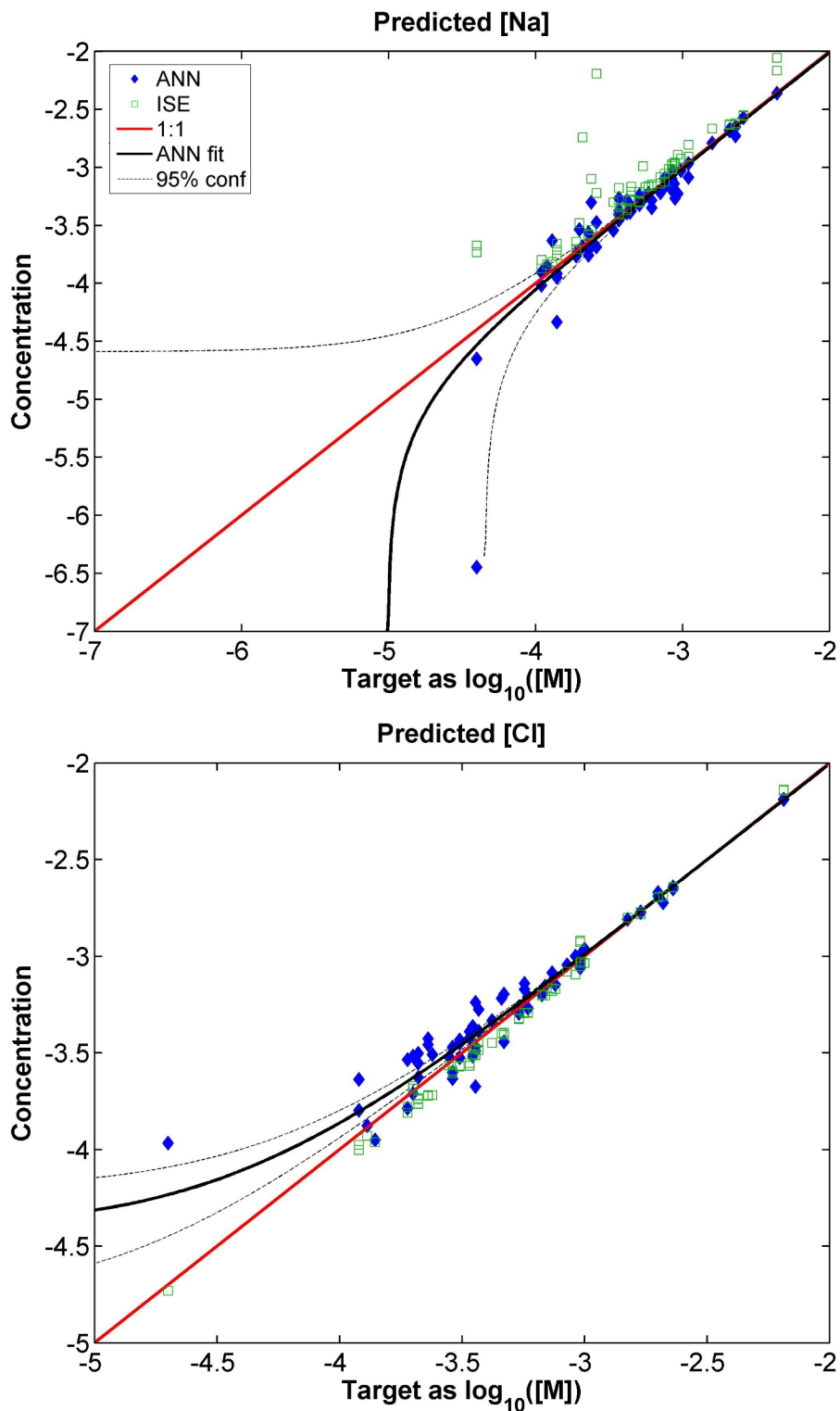


Figure 7-11: Scatter plots of Na^+ and Cl^- ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.

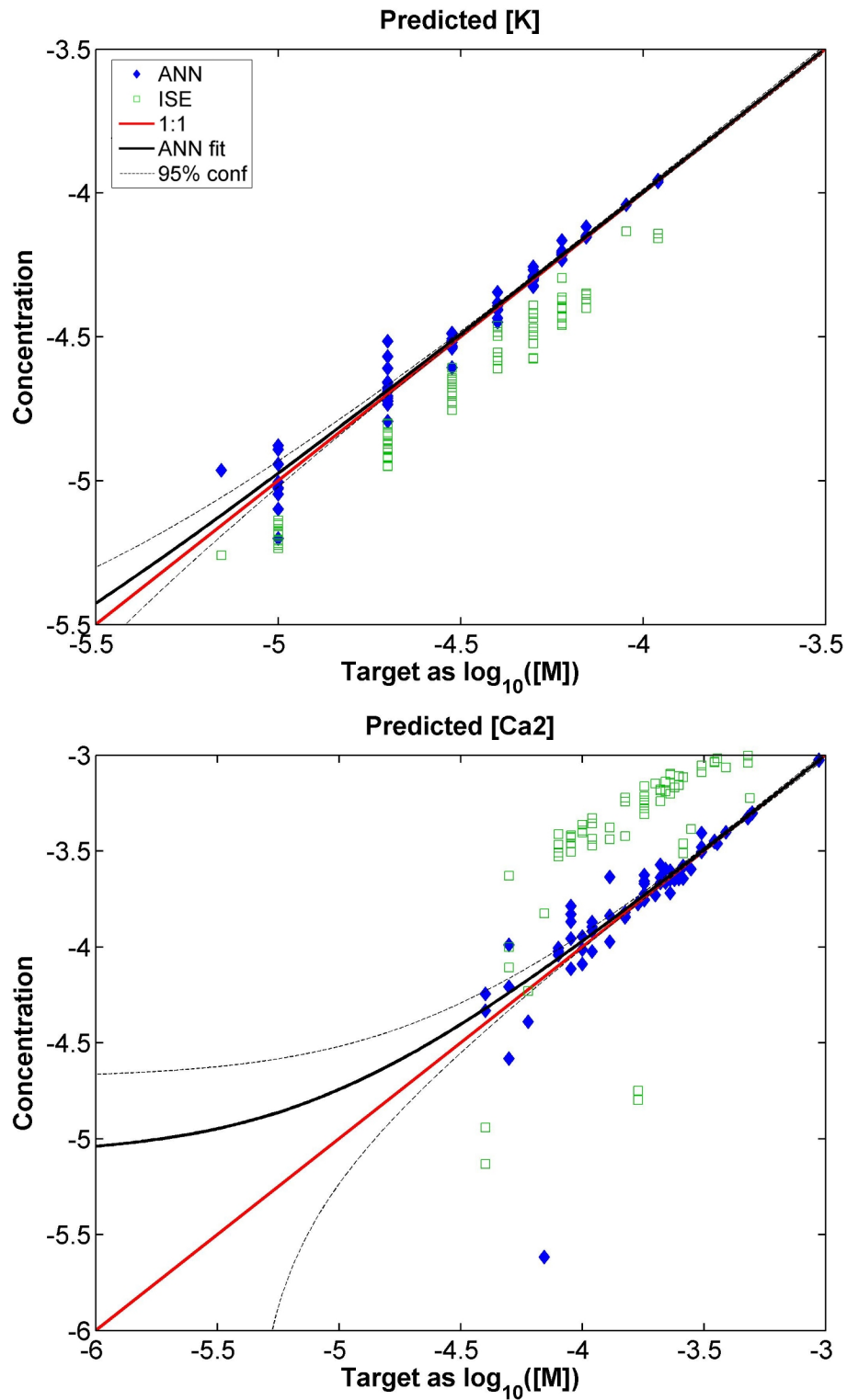


Figure 7-12: Scatter plots of K^+ and Ca^{2+} ion concentrations predicted using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.

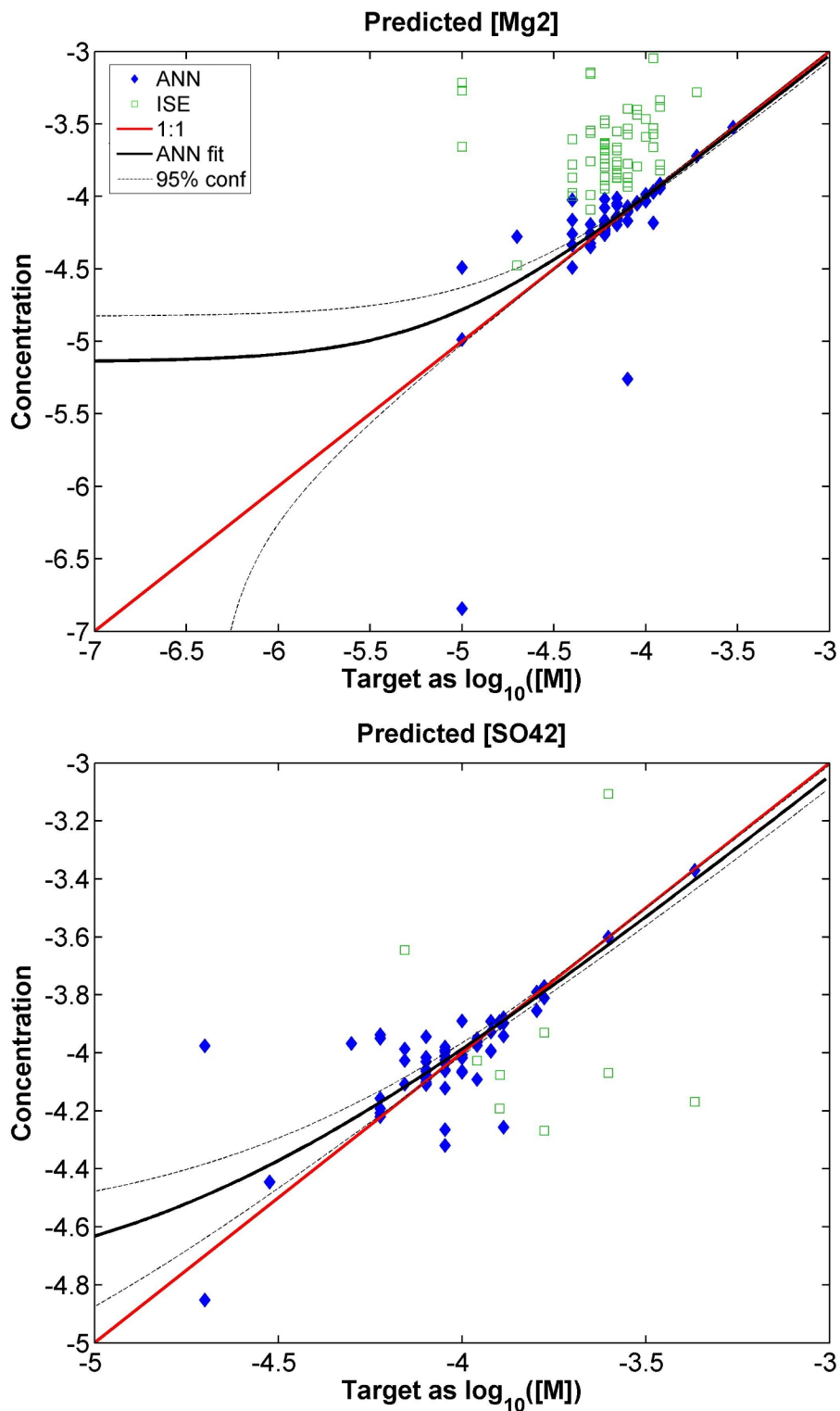


Figure 7-13: Scatter plots of Mg^{2+} and SO_4^{2-} ion concentrations predicted using the optimal ANN as a function of target concentration. (Note most ISE predictions do not fit on graph at this scale.) One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black.

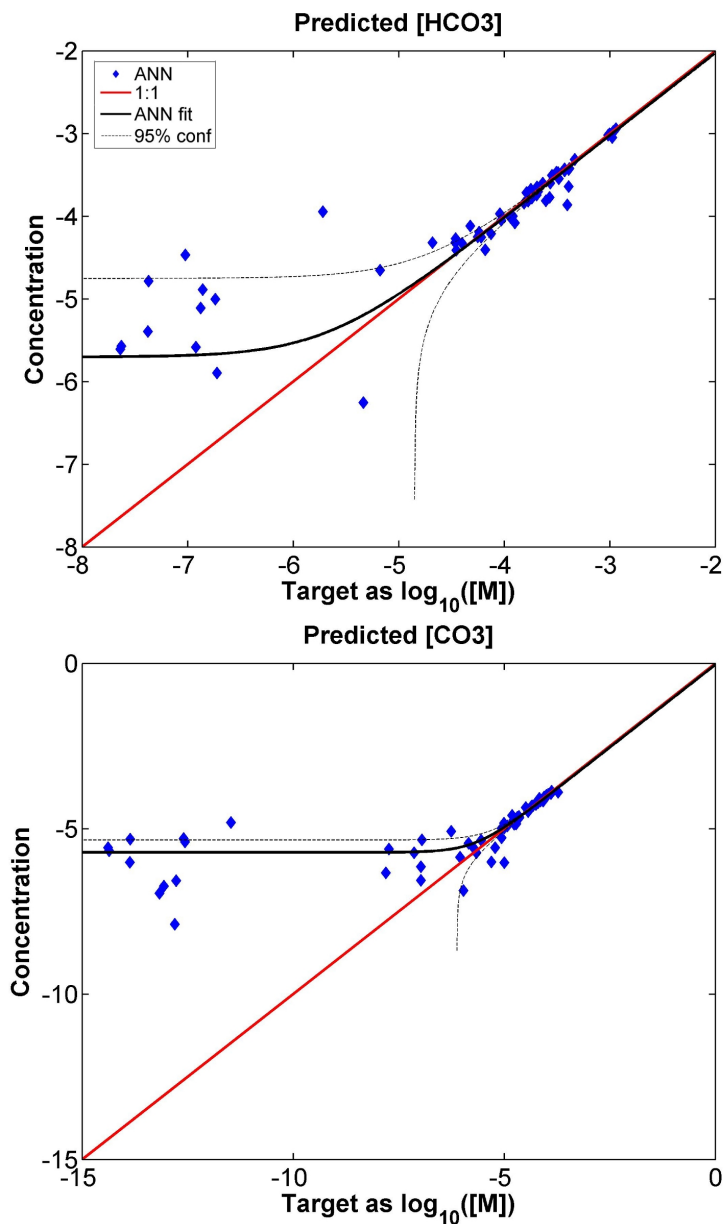


Figure 7-14: Scatter plots of carbonate system ion concentrations predicted using the optimal ANN as a function of target concentration. Note that there is significant bias in these estimates at increasingly small concentrations; it is expected that improvement in sulfate concentrations (which have the similar magnitude contribution as carbonate concentrations in the charge balance equation) would further reduce the uncertainty in these low-concentration predictions, while the same can be stated for further simultaneous improvement of bicarbonate and chloride predictions. One-to-one line shown in red; regression of estimates against targets (concentration data - note low concentrations do not contribute significantly to this fit) and 95% confidence interval on the linear fit shown in black.

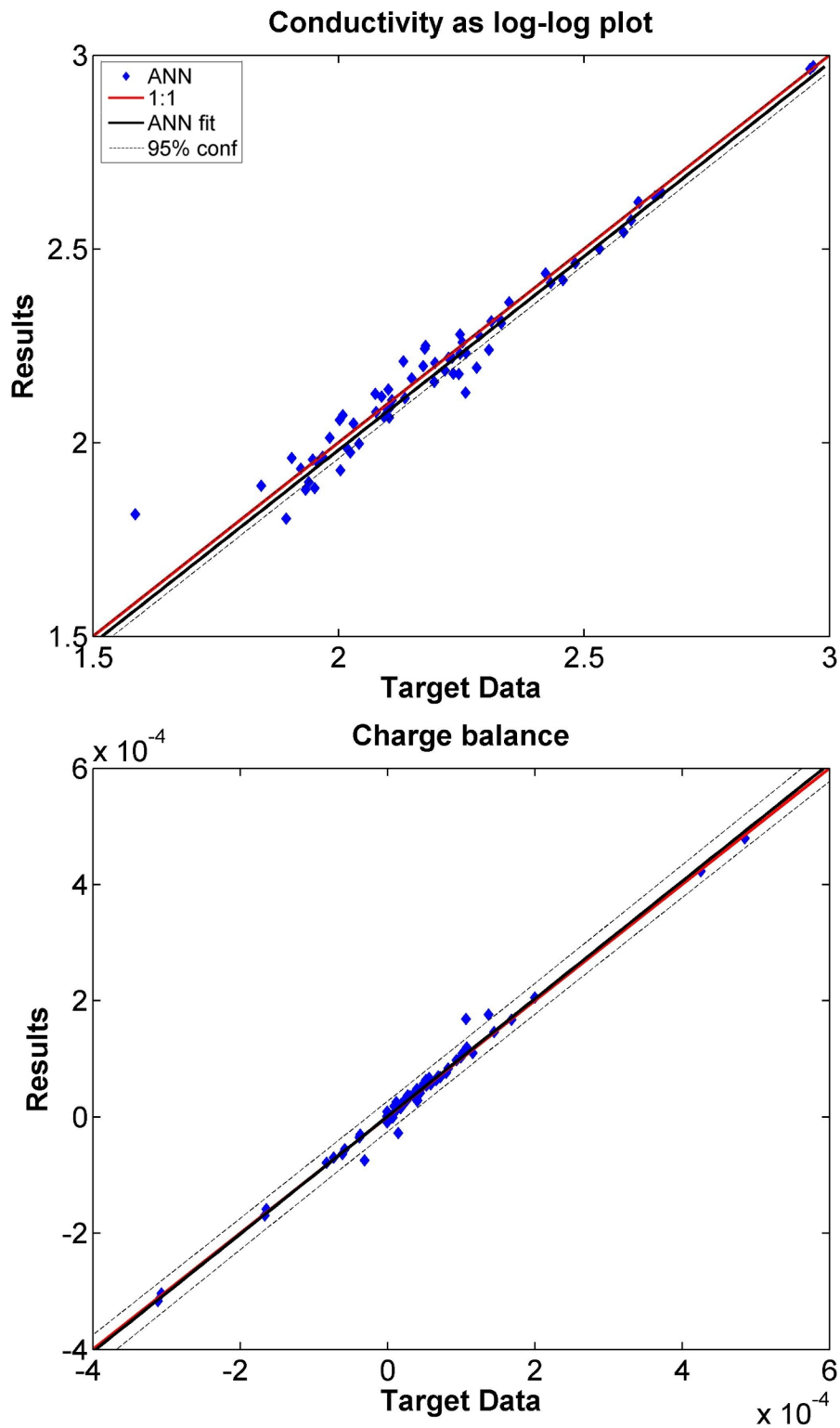


Figure 7-15: Scatter plot of constraint predictions of optimal ANN as a function of target value. One-to-one line shown in red; regression of estimates against targets (data before log-transformation) and 95% confidence interval on the linear fit shown in black.

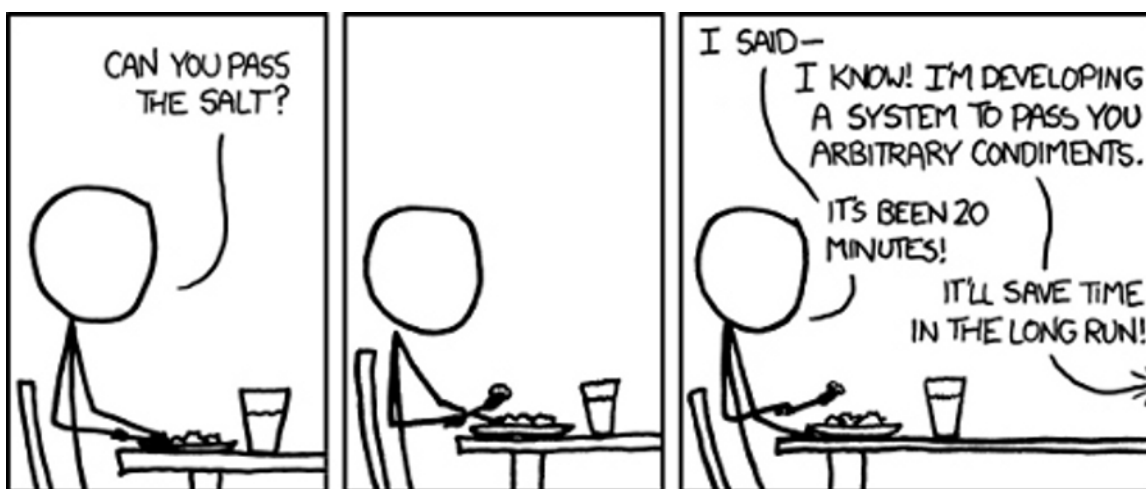
7.7 Conclusions

Overall, these results demonstrate the successful estimation of target ion concentrations, including nitrate and ammonium, **at environmental levels and in environmentally representative mixtures** through use of a novel ANN structure that implements constraints based on both conductivity and charge neutrality. All ion concentration estimates are unbiased with mean relative errors of approximately 20% on most channels (approximately 50% for NH_4^+). This is true **even for analytes for which no specific commercial sensor exists**, i.e., SO_4^{2-} and Mg^{2+} . This is a major step in the direction of direct in-situ quantification of these analytes in real time. It is important to note that the waters being considered here - i.e., New England fresh waters - are bimodal, containing both hard and soft waters, and results could potentially be improved by focusing on these separately, e.g., using two independent neural network structures trained on subsets of the data and between which one can select based on, for example, the hardness or pH electrode reading.

Of course there remain many additional directions to explore and characterize, including different training regimes for the ANNs, optimization of the training set, and quantification of the effects of the background matrix of real waters, and these will be expounded upon further in Chapter 8.

Chapter 8

Conclusions



The essence of doing my PhD. (I'm developing a system to plot arbitrary data in Matlab but have it look nice. - It's been three hours!) (I'm developing a system to select arbitrary environmentally relevant ion concentration sets. - It's been two months!) (I'm developing a system to measure arbitrary ions in fresh waters. - It's been 8 years! - It'll save time in the long run!!) *Image from xkcd.com with gratitude.*

8.1 Project summary and conclusions

The work presented in this thesis demonstrates a method for **real-time, in-situ determination of a suite of relevant environmental ions**, including nitrate and ammonium, using a novel signal processing algorithm coupled with a full suite on ion selective electrodes (11) and other sensors (3). The combined multi-sensor / ANN architecture is portable, low-power ($\ll 1$ W), and provides unbiased concentration predictions in un-preprocessed sample mixtures at environmental levels (down to $<10 \mu\text{M}$ for the NO_3^- and NH_4^+) in ~ 5 min. Relative errors on most channels are $\leq 20\%$ for the optimum architecture chosen here, even for analytes for which no specific sensor is available (i.e., Mg^{2+} , SO_4^{2-} , and HCO_3^-), and numerous pathways are available for further improvement or relative optimization (see below). Ability to quantify this analyte set *in the field* and *in real time* can now enable source tracking - systematic sampling in a contaminated area in order to travel up-gradient

and identify contamination provenance - for example, in cases where multiple factories, properties, or agricultural fields could be suspected sources. Similarly, analysis of the relative ion concentrations (ion ‘fingerprinting’) in a large river may provide information about which of its tributary streams (or, potentially, groundwater) is likely to be the source of an identified anomaly. Real time sampling is also guaranteed to improve sample density, decrease our uncertainty about the true characteristics of a given ecosystem, and provide key data necessary to inform environmentally conservative and fiscally responsible management decisions.

Development of the instrumentation described here also led to a number of other valuable contributions to the field, including:

- Design of custom ISE interface hardware to provide measure stable, noise-free voltage potential signals from a suite of ISEs
- Proposal of a standardized method for automation of ISE potential measurements to promote measurement replicability, facilitate inter-comparison of data, and allow unmonitored simultaneous measurement using a number of ISEs with measurements generally completed in <6.5 min.
- Description and validation of a novel ANN architecture optimized for use on chemical data via incorporation of constraints based on a priori chemical knowledge

This ANN technique could easily be extended from use in New England waters to other water bodies around the US and the globe through creation of appropriate training data sets. This does not necessarily require inclusion of water samples from every environment but would require inclusion of mixes that are representative of many different geographies. It will also be interesting to explore the trade-offs between creating a single ANN trained with data from all geographies of interest and creating several ‘specialized’ ANNs (expected to have fewer hidden nodes and thus a faster training time and smaller required training set) for regional waters, the outputs of which could be multiplexed based on certain environmental characteristics or specified directly by the user.

The ANN architecture presented also has the potential to become a powerful tool for analysis of any number of environmental datasets, namely those for which the target relationship is non-linear and poorly constrained by current models (or highly dependent on parameterization of the model) and for which *a priori* knowledge of the system can be integrated. For example, the charge balance constraint implemented here could easily be generalized to mass balance for any system components. Especially in cases where it is relatively inexpensive to collect training data, this method could prove much faster and less costly than performing the lab work required to exactly parameterize known physical models.

8.2 Suggestions for future work

Exciting potential extensions of this work exist at many scales, from data optimization to exploration of a number of interesting field sites. This section provides details of just a few representative opportunities.

8.2.1 Design for prolonged and in-situ use

Work presented here serves as a proof of concept for the measurement mechanism for a novel field instrument, however as yet the hardware and software have not been optimized

for all conditions that will be encountered during long-term and in-situ use. Software will require an additional pre-processing unit that compensates mV measurements for any drift in ISE signals; this will necessarily be interposed between the data acquisition module and the ANN software. A single calibration standard, e.g., run at the beginning of each sampling day, will allow for changes in voltage offset. While eventual degradation of ISE membranes is expected to affect slope of responses as well, continued accurate predictions under these conditions could require a complete re-training of the ANN software rather than an offset adjustment, and as such replacement of the affected components would instead be recommended.

These requirements tie in nicely with capabilities for scaling for eventual construction of multiple systems, e.g., as a commercial product. Training of the neural network need only be completed for a single prototypical set of ISEs; ISEs installed in a particular field instrument can then be calibrated in reference to the original sensor to take advantage of the pre-trained ANN.

Construction of a field-ready (e.g., stainless steel) housing will also be required to protect sensor membranes from abrasion during measurements and transport. This housing will also need to provide electrical isolation (essentially act as a Faraday cage) and should be connected to the power supply ground.

Finally, targeted improvements to the base system can be identified to direct primary research. Improved specificity and lowering of detection limits for key ISEs (SO_4^{2-} , CO_2 or preferably HCO_3^-) would serve to improve concentration estimates for all analytes by providing additional information to the system and reducing uncertainty on significant terms in the conductivity and charge balance calculations. Miniaturization and isolation may also be facilitated through use of techniques that are becoming cheaper and more reliable every year: chemfet construction and/or screen-printing of ISEs. By reducing the length of physical wires connecting electrodes to post-processing electronics, coupling of electronic noise will likely be lessened. This method would also have the potential to reduce overall power consumption and provide sampling in smaller sample volumes.

8.2.2 ANN optimization

There are a number of additional dimensions for optimization that were beyond the scope of this thesis. For example:

- Quantification of the sensitivity of results to training set size (i.e., to find the minimum number of samples required to produce accurate results)
- Quantification of the sensitivity of results to training set *contents*, i.e., what types of samples provide the most information during training and which could be omitted to streamline the process?
- Optimization of values in the error weighting vector (or adaptive error weighting)
- Incorporation of uncertainty into data (e.g., on EC targets or ISE inputs) by adding Gaussian random noise on these channels and including these samples in the training set (following [179, 180])

8.2.3 The environmental matrix

It is also extremely important to consider the effects of taking measurements under actual, i.e., in-situ, environmental conditions, i.e., in waters where temperatures will vary, other charged species may exist (e.g., DOC), or ionic strengths may vary over several orders of

magnitude (e.g., estuarine). In some cases, such characteristics may be compensated for by expansion of the training set (e.g., temperature compensation has been demonstrated using this technique), however it is not necessarily clear that this will always be the case. It is thus necessary to bound the range of applicability for this technique by exploring the gradient from relatively ‘benign’ to more ‘complex’ waters. A few interesting cases include:

- Range from oligotrophic to eutrophic waters
- Comparison of results in waters from different geographic areas
- Range from fresh-to-salt waters, e.g., in an estuarine environment
- Range from low-DOC to high-DOC waters

Finally, once challenge cases have been identified, it will be interesting to explore the possibilities for incorporation of additional sensors of different types (e.g., optical, spectrometric) to provide the information required to restore accurate predictions. Overall, however, results of the initial tests presented in this thesis are already promising for the most ‘benign’ end of the spectrum and thus strongly motivate further development of this in-situ instrument.

Appendix A

Supplementary Materials for Chapter 7



This appendix contains additional tables and figures as referenced from Chapter 7, including more details describing the environmental 8-ion joint PDF, the environmentally-representative samples used in ANN training, and extended ANN results.

A.1 Sample creation and characteristics

Table A.1: Delineation of bin edges (as $\log_{10}(M)$) for PDF based on USGS-recorded environmental samples.

Bin edge	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}
1-2	-6.78	-4.83	-4.78	-5.83	-5.72	-6.81	-5.72	-5.25
2-3	-6.33	-4.50	-4.33	-5.50	-5.17	-6.42	-5.17	-4.75
3-4	-5.89	-4.17	-3.89	-5.17	-4.61	-6.03	-4.61	-4.25
4-5	-5.44	-3.83	-3.44	-4.83	-4.06	-5.64	-4.06	-3.75
5-6	-5.00	-3.50	-3.00	-4.50	-3.50	-5.25	-3.50	-3.25
6-7	-4.56	-3.17	-2.56	-4.17	-2.94	-4.86	-2.94	-2.75
7-8	-4.11	-2.83	-2.11	-3.83	-2.39	-4.47	-2.39	-2.25
8-9	-3.67	-2.50	-1.67	-3.50	-1.83	-4.08	-1.83	-1.75
9-10	-3.22	-2.17	-1.22	-3.17	-1.28	-3.69	-1.28	-1.25

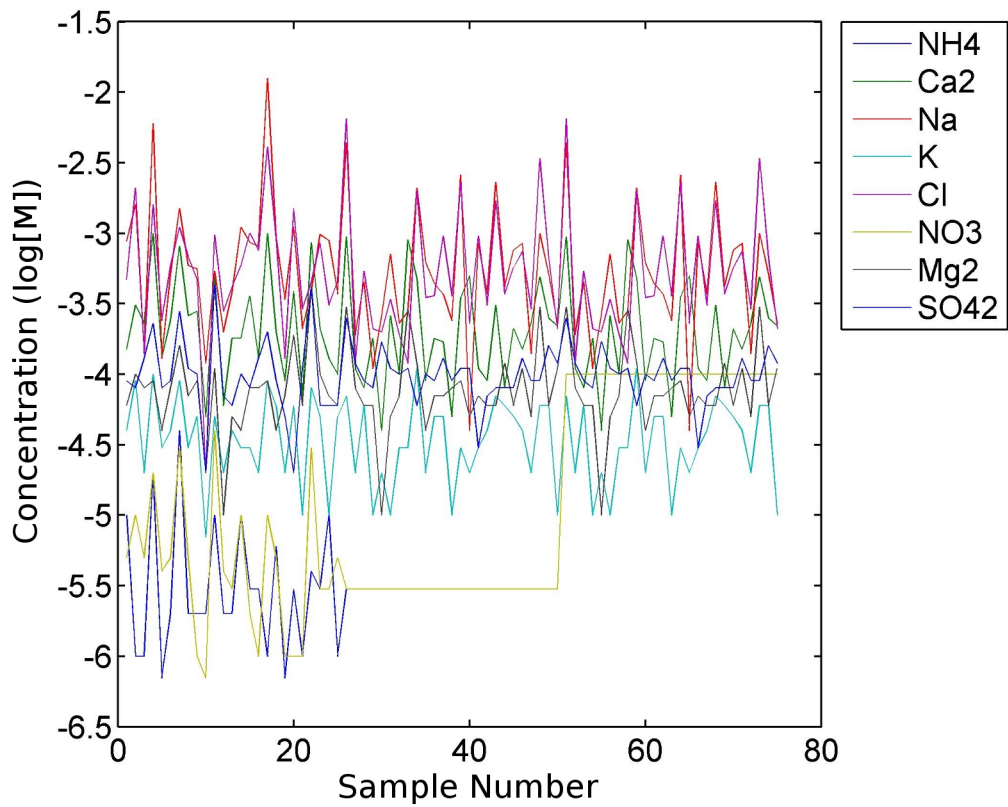


Figure A-1: Ion concentrations in 75 training samples, plotted against sample number. Recall that 'low' and 'high' nitrogen conditions were imposed on, respectively, samples 26-50 and samples 51-75.

Table A.2: Ion concentrations (μM , except for alkalinity which is given in mM) for chosen training sample mixtures. Recall that ‘low’ and ‘high’ nitrogen conditions were imposed on, respectively, samples 26-50 and samples 51-75; in some cases, equilibrium pH will lead to decrease in NH_4^+ as equilibrium shifts toward NH_3 . Calculations are given for alkalinity, pH, and carbonate species, for cases of (1) full equilibration with atmospheric CO_2 (392 ppm, recorded at Mauna Loa Dec. 2011) or (2) limited carbon exchange (carbonate system limited by standards used for sample creation).

#	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	Equilibrium				Limited exchange				
									Alk	pH	HCO_3^-	CO_3^{2-}	EC	pH	HCO_3^-	CO_3^{2-}	EC
1	10	150	880	40	470	5	60	90	0.694	8.1	685	4.3	154	10	253	156.6	184
2	1	310	1600	90	2100	10	100	80	0.241	7.6	240	0.5	319	5.9	242	0	319
3	1	230	140	20	140	5	80	130	0.376	7.8	373	1.3	93	10.5	10	17.2	145
4	20	1000	6000	100	1600	20	90	230	6.214	8.9	5549	327.3	902	10.8	466	2433.7	1202
5	0.7	130	130	30	240	4	40	80	0.097	7.2	96	0.1	64	9.7	30	7.6	73
6	2	240	510	40	590	5	80	90	0.417	7.8	413	1.6	144	9.9	166	78.6	164
7	40	810	1500	90	1100	30	160	280	1.875	8.4	1807	32.4	415	10.7	142	518.5	552
8	2	260	590	30	690	5	70	110	0.367	7.8	364	1.2	157	9.7	199	56	170
9	2	280	560	50	470	1	90	100	0.681	8	671	4.2	158	10.3	136	158.8	204
10	2	50	120	7	20	0.7	20	20	0.208	7.6	207	0.4	29	10.1	40	23.4	49
11	10	490	540	50	970	40	110	430	-0.07	4.2	0	0	272	4.2	0	0	272
12	2	60	200	20	280	4	10	70	-0.062	4.2	0	0	74	4.2	0	0	74
13	2	180	380	40	420	3	50	60	0.339	7.7	336	1	106	10	120	59.6	126
14	10	180	1100	30	590	10	40	100	0.779	8.1	767	5.5	181	10	315	174.8	210
15	3	360	860	30	1000	2	80	80	0.611	8	603	3.4	214	9.8	318	111.7	232
16	3	130	810	20	760	1	80	130	0.232	7.6	231	0.5	156	6.6	232	0.1	156
17	1	1000	12600	90	4100	10	90	200	10.361	9.1	8642	851.8	1644	10.3	2284	3906.3	1927
18	6	200	930	60	920	5	40	90	0.371	7.8	368	1.3	181	7.2	370	0.4	181
19	0.7	90	340	20	130	1	70	50	0.45	7.9	445	1.8	75	10.2	94	86.2	110
20	3	390	1100	60	1500	1	190	20	0.782	8.1	769	5.7	280	9.8	394	155.5	303
21	1	70	210	10	290	1	60	110	-0.03	4.5	0	0	76	4.5	0	0	76
22	4	860	390	80	450	30	420	400	1.754	8.4	1694	28.4	354	11	29	206.3	570
23	3	210	980	50	850	3	100	60	0.68	8	670	4.2	194	9.8	367	123.2	213
24	10	130	890	10	310	3	70	100	0.876	8.2	861	6.9	142	10.3	220	224.8	189
Low N concentrations																	
26	3	940	4400	70	6500	3	300	250	-0.05	4.3	0	0	923	4.3	0	0	923
27	3	110	190	20	120	3	80	120	0.23	7.6	229	0.5	72	10.1	38	26	95
28	3	80	450	60	540	3	60	90	0.07	7.1	70	0	102	6.1	71	0	102
29	3	180	110	10	210	3	60	80	0.23	7.6	229	0.5	74	10.2	21	17.8	101
30	3	40	260	20	200	3	10	170	-0.158	3.8	0	0	118	3.8	0	0	118
31	3	260	710	10	340	3	50	110	0.78	8.1	768	5.5	152	10.4	118	181	211
32	3	100	230	30	190	3	70	100	0.21	7.5	209	0.4	73	9.9	70	28.4	88

Table A.2 – continued from previous page.

#	Equilibrium										Limited exchange						
	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻	Alk	pH	HCO ₃ ⁻	CO ₃ ²⁻	EC	pH	HCO ₃ ⁻	CO ₃ ²⁻	EC
33	3	900	280	30	120	3	290	110	2.35	8.5	2247	49.4	290	11.2	25	249.8	587
34	3	480	2100	110	2000	3	120	60	1.29	8.3	1257	15.5	405	9.6	853	196.2	425
35	3	90	620	20	350	3	40	100	0.35	7.8	347	1.1	106	9.7	201	49.3	117
36	3	180	440	50	360	3	70	90	0.45	7.9	445	1.8	117	10.1	133	90.5	144
37	3	170	370	50	960	3	70	130	-0.317	3.5	0	0	255	3.5	0	0	255
38	3	50	240	10	360	3	80	90	-0.03	4.5	0	0	80	4.5	0	0	80
39	3	350	2600	30	2300	3	90	110	0.99	8.2	970	9.2	422	6.8	989	0.4	421
40	3	500	40	20	230	3	50	136	0.71	8.1	700	4.6	136	10.8	2	6.6	243
41	3	110	850	30	960	3	70	30	0.22	7.6	219	0.4	154	6.3	220	0	154
42	3	90	370	40	310	3	60	70	0.26	7.6	258	0.6	86	9.7	141	34.3	95
43	3	310	2300	70	1700	3	60	80	1.25	8.3	1219	14.4	364	9.2	1027	101.5	373
44	3	80	420	60	370	3	120	80	0.35	7.8	347	1.1	105	9.9	150	59.5	121
45	3	210	760	50	570	3	60	80	0.62	8	612	3.5	158	10	222	137.6	187
46	3	150	850	40	740	3	110	130	0.41	7.8	406	1.5	171	8.4	396	5.4	172
47	3	230	140	20	290	3	50	90	0.25	7.6	248	0.6	89	10.2	20	18.8	120
48	3	490	1000	60	3400	3	300	90	-0.934	3	0	0	745	3	0	0	745
49	3	250	510	60	670	3	60	160	0.2	7.5	199	0.4	152	9.4	137	17.9	157
High N concentrations																	
51	100	940	4400	70	6500	100	300	250	-0.05	4.3	0	0	938	4.3	0	0	937
52	100	110	190	20	120	100	80	120	0.228	7.6	227	0.5	86	9.9	39	15.6	90
53	100	80	450	60	540	100	60	90	0.07	7.1	70	0	116	6.1	71	0	116
54	100	180	110	10	210	100	60	80	0.228	7.6	227	0.5	87	10	23	11.7	94
55	100	40	260	20	200	100	10	170	-0.158	3.8	0	0	132	3.8	0	0	132
56	100	260	710	10	340	100	50	110	0.774	8.1	761	5.4	165	10.3	134	160.9	203
57	100	100	230	30	190	100	70	100	0.208	7.5	207	0.4	87	9.6	70	14.7	87
58	100	900	280	30	120	100	290	110	2.334	8.5	2232	48.8	303	11.2	27	248.5	575
59	100	480	2100	110	2000	100	120	60	1.28	8.3	1248	15.3	418	9.4	899	150.1	427
60	100	90	620	20	350	100	40	100	0.347	7.8	344	1.1	120	9.3	225	25	119
61	100	180	440	50	360	100	70	90	0.446	7.9	442	1.8	131	10	133	67.2	142
62	100	170	370	50	960	100	70	130	-0.317	3.5	0	0	269	3.5	0	0	269
63	100	50	240	10	360	100	80	90	-0.03	4.5	0	0	94	4.5	2	0	95
64	100	350	2600	30	2300	100	90	110	0.982	8.2	962	9.1	435	6.8	989	0.4	436
65	100	500	40	20	230	100	50	110	0.704	8.1	694	4.5	149	10.7	0	0	233
66	100	110	850	30	960	100	70	30	0.218	7.6	217	0.4	168	6.3	220	0	168
67	100	90	370	40	310	100	60	70	0.258	7.6	256	0.6	99	9.3	159	15.5	98
68	100	310	2300	70	1700	100	60	80	1.24	8.3	1210	14.2	377	9	1057	71.3	380
69	100	80	420	60	370	100	120	80	0.347	7.8	344	1.1	119	9.6	175	34.5	120
70	100	210	760	50	570	100	60	80	0.615	8	607	3.4	171	9.8	257	102.8	182
71	100	150	850	40	740	100	110	130	0.406	7.8	403	1.5	185	8.1	396	2.9	185
72	100	230	140	20	290	100	50	90	0.248	7.6	246	0.6	103	10.1	19	11.4	113
73	100	490	1000	60	3400	100	300	90	-0.934	3	0	0	759	3	0	0	759
74	100	250	510	60	670	100	60	160	0.198	7.5	197	0.4	166	8.9	148	6.5	164

Table A.2 – continued from previous page.

#											Equilibrium		Limited exchange				
	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	Alk	pH	HCO_3^-	CO_3^{2-}	EC	pH	HCO_3^-	CO_3^{2-}	EC
75	100	220	230	10	210	100	110	120	0.446	7.9	442	1.8	120	10.3	52	53.1	144

A.2 ISE calibrations and results

Figures A-2–A-4 show the individual response of each ISE to each of five single-salt calibration standards over the concentration range $0.1 \mu\text{M}$ - 100 mM . Note that NH_4Cl and CaCl_2 standards were made ~ 18 months earlier than the other standards and differences in baseline values can thus be accounted for partially by equilibration with atmospheric CO_2 (changes in pH, interferences due to HCO_3^-). Elevated EC and depressed pH measurements in low concentration samples also indicate that these samples have been subject to an additional influx of H^+ ions over this period of time, although the source of these ions is unknown. (Calculations show that $\sim 10 \mu\text{L}$ of 10% concentrated acid, e.g., HNO_3 , could be responsible for the pH change, and experiments are thus being run to determine whether leaching of acid into the LDPE bottles during acid washing, and out of the bottles when filled with low-ionic strength waters, may have taken place.) Table A.3 shows the relevant calibration parameters for the most selective (or only available) electrode for each of the target ions. These parameters were used to calculate ion concentrations in the mix samples; the relative error (as %) is given as a function of concentration in Figure A-5.

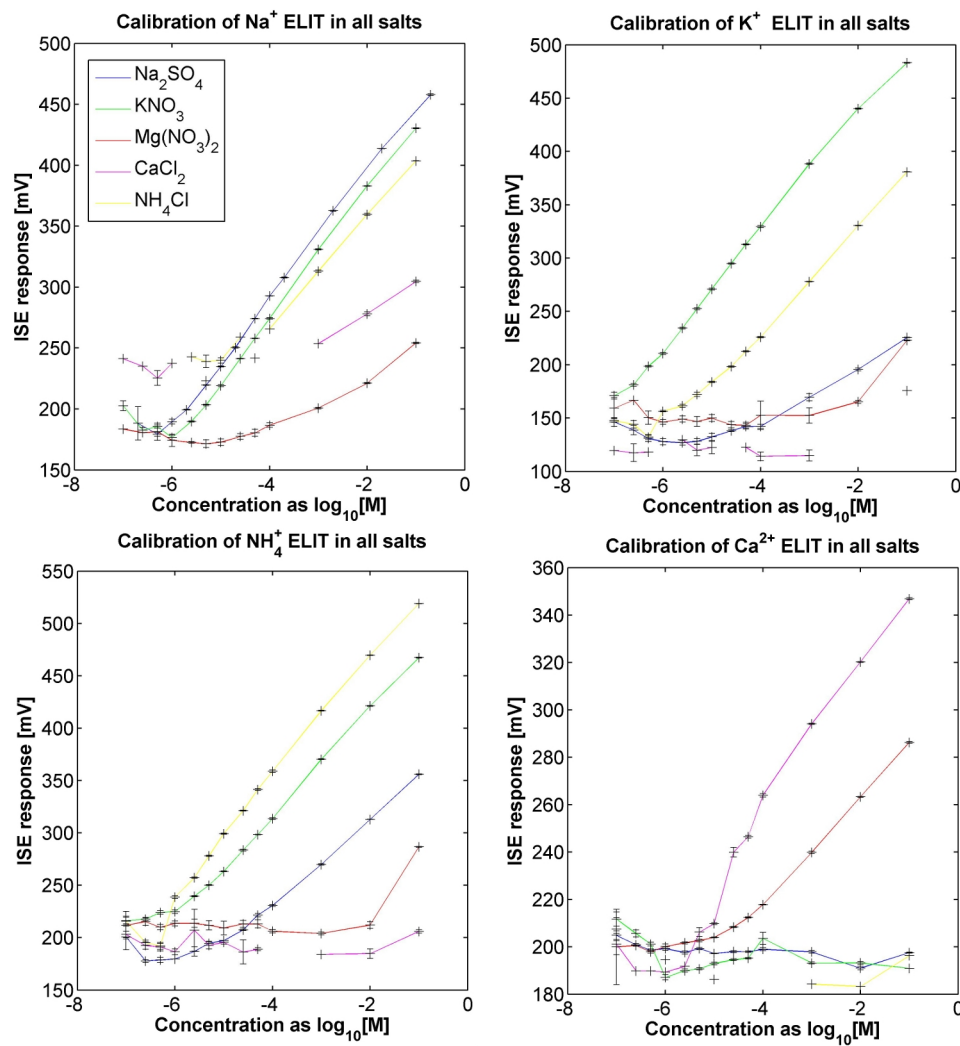


Figure A-2: Response of ELIT ISEs to each of five single-salt calibration standards.

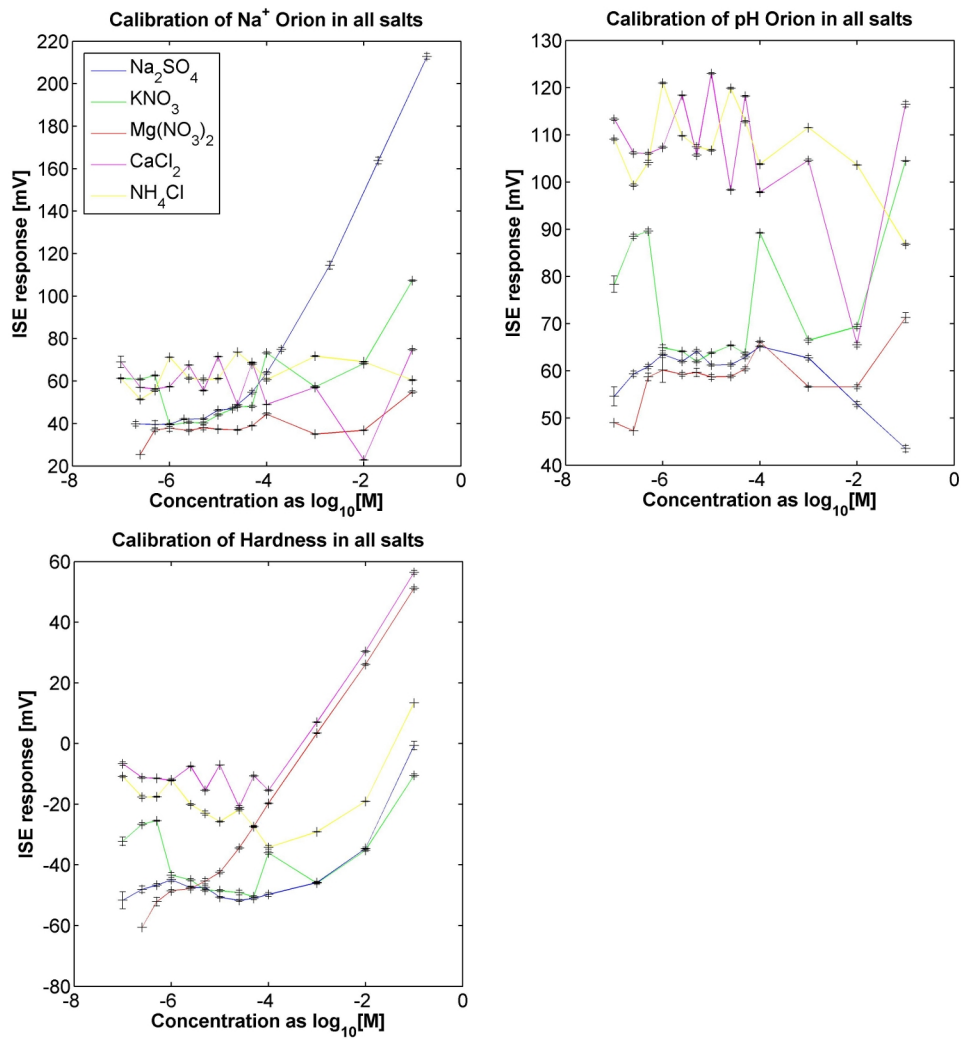


Figure A-3: Response of glass and divalent cation ISEs to each of five single-salt calibration standards.

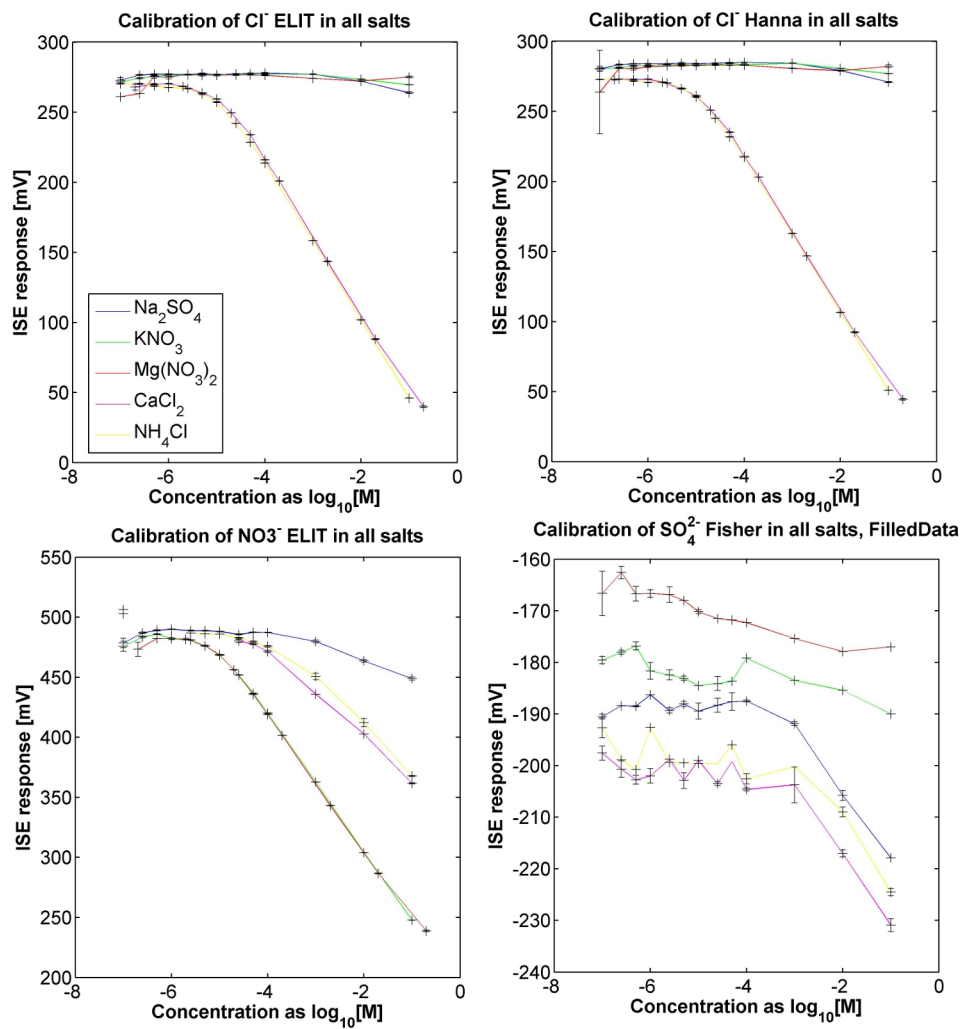


Figure A-4: Response of anion ISEs to each of five single-salt calibration standards.

Table A.3: Most informative calibration curve for prediction of target ions directly using ISEs as stand-alone sensors

Ion	ISE	slope	intercept	LOD [M]	R ²	RMSE
Na ⁺	ELIT Na ⁺	52.7	500.6	2·10 ⁻⁶	0.999	3.31
K ⁺	ELIT K ⁺	55.5	548	2.5·10 ⁻⁷	0.998	4.53
Ca ²⁺	ELIT Ca ²⁺	33.9	388.9	2.5·10 ⁻⁶	0.981	7.95
Mg ²⁺	Hardness	23.6	74.1	2.5·10 ⁻⁵	>0.999	0.6
NH ₄ ⁺	ELIT NH ₄ ⁺	57	582.7	1·10 ⁻⁶	0.998	4.4
Cl ⁻	ELIT Cl ⁻	-53.8	-0.014	2·10 ⁻⁵	0.998	2.85
SO ₄ ²⁻	Fisher SO ₄ ²⁻	-13	-231.2	0.001	0.998	0.73
NO ₃ ⁻	ELIT NO ₃ ⁻	-56.2	192.6	1·10 ⁻⁵	0.999	2.85

A.3 ANN evaluation and extended results

As discussed in the text, determination of the best metric for selection of the ‘winning’ neural network was based on an analysis of the relationship between single-value metrics (necessary for comparison of different ANNs) and accuracy of prediction of the nitrogen species. The results given here are for a single ‘External’ parameter set (12 ion outputs; no EC or CB constraints; mix data only in training set (no single-salt data); training to log-transformed data) but are representative of results for any such set. Figure A-6 shows the correlation between single-value metrics used to assess ANN quality (i.e., MSE, NRMSE, and MRE calculated using the full output set or only the 8 target ions) and the metrics used to assess mean response quality for each of the nitrogen species ions. Table A.4 shows the pairwise correlation coefficients for each of these metrics. Based on these data it is clear that a goodness metric calculated using only errors for the 8 target ions should be used. Both NRMSE and MRE calculations show excellent correlation to the respective values for NH₄⁺ and NO₃⁻; visual inspection and inter-comparison of results demonstrated that networks selected using these two metrics are both of good quality but are optimized differently, e.g., fewer outliers but wider overall spread. As such the NRMSE results are presented in the primary text while MRE results are included here.

Table A.4: Pairwise correlation coefficients between net (whole data set) goodness metrics and those calculated individually for nitrogen ions. Methods are: MSE (mean squared error), NRMSE (normalized root mean squared error), MRE (mean of absolute value of relative error).

Outputs used	Method	NH ₄ ⁺		NO ₃ ⁻	
		NRMSE	MRE	NRMSE	MRE
All	MSE	0.3864	0.4166	0.3561	0.3881
All	NRMSE	0.4165	0.4317	0.3774	0.3964
All	MRE	0.1492	0.1960	0.1703	0.2104
8 ions	MSE	0.6631	0.6784	0.6076	0.6105
8 ions	NRMSE	0.8505	0.7858	0.7598	0.6953
8 ions	MRE	0.7236	0.8591	0.6845	0.7910

Table A.5: Parameterizations for best ANN (chosen using MRE metric) as a function of ‘External’ architecture.

Architecture			Parameterization			
Outputs	Data	Normalization	Hidden Layers	μ	μ_{dec}	μ_{inc}
12 ions	mixes	none	[18,12,9]	0.1	0.5	50
19 params	mixes	none	[12,18]	0.1	0.1	10
12 ions+CB	mixes	none	[12,6,15]	0.1	0.1	50
12 ions+EC	mixes	none	[18,18,18]	0.1	0.1	50
12 ions+CB+EC	mixes	none	[6,18,15]	0.001	0.1	10
12 ions	all	none	[9,15,12]	0.1	0.5	50
19 params	all	none	[12,18,18]	0.001	0.9	50
12 ions+CB	all	none	[9,12]	0.1	0.1	10
12 ions+EC	all	none	[12,18,12]	0.001	0.1	10
12 ions+CB+EC	all	none	[18,15]	0.1	0.9	1.5
12 ions	mixes	log ₁₀	[15,12,18]	0.1	0.9	50
19 params	mixes	log ₁₀	[12,18,18]	0.1	0.1	10
12 ions+CB	mixes	log ₁₀	[15,15,15]	0.1	0.1	50
12 ions+EC	mixes	log ₁₀	[18,15,9]	0.1	0.5	50
12 ions+CB+EC	mixes	log ₁₀	[9,9,12]	0.1	0.1	10
12 ions	all	log ₁₀	[18,15,15]	0.001	0.1	50
19 params	all	log ₁₀	[6,18,9]	0.001	0.1	1.5
12 ions+CB	all	log ₁₀	[9,15,9]	0.1	0.1	1.5
12 ions+EC	all	log ₁₀	[15,12,6]	0.001	0.1	10
12 ions+CB+EC	all	log ₁₀	[18,18,12]	0.1	0.9	1.5
Errors on constraints weighted more heavily						
12 ions+CB	mixes	none	[18,18,18]	0.001	0.5	50
12 ions+EC	mixes	none	[9,9]	0.001	0.5	10
12 ions+CB+EC	mixes	none	[18,15]	0.1	0.9	50
12 ions+CB	all	none	[6,18]	0.1	0.1	1.5
12 ions+EC	all	none	[18,12]	0.1	0.1	10
12 ions+CB+EC	all	none	[12,12]	0.001	0.5	50
12 ions+CB	mixes	log ₁₀	[15,9,12]	0.001	0.9	1.5
12 ions+EC	mixes	log ₁₀	[9,18,18]	0.1	0.9	10
12 ions+CB+EC	mixes	log ₁₀	[15,15,15]	0.1	0.1	50
12 ions+CB	all	log ₁₀	[18,15,18]	0.1	0.1	1.5
12 ions+EC	all	log ₁₀	[12,12,18]	0.001	0.5	50
12 ions+CB+EC	mixes	log ₁₀	[9,18,9]	0.1	0.1	1.5

Table A.6: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **concentration values of mix data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	$\sum \text{Err}$
no	12	0.271 (55.8)	0.266 (24.1)	0.205 (20)	0.183 (10.1)	0.183 (23.7)	0.097 (29.9)	0.326 (32.2)	0.489 (34.9)	2.02 (230.7)
no	19	0.188 (52.6)	0.167 (18.4)	0.174 (15.8)	0.159 (10.7)	0.184 (15.3)	0.125 (46.3)	0.396 (21.9)	0.288 (20.3)	1.681 (201.3)
no	12+CB	0.335 (58.3)	0.289 (26)	0.138 (17.1)	0.158 (10.4)	0.17 (27.1)	0.34 (39.4)	0.312 (26)	0.267 (28.1)	2.009 (232.4)
no	12+EC	0.308 (61.4)	0.326 (21.3)	0.105 (15.4)	0.135 (9.7)	0.148 (16.3)	0.31 (54)	0.327 (19.6)	0.452 (24.7)	2.111 (222.4)
no	12+CB+EC	0.081 (54.3)	0.148 (20.2)	0.145 (16.9)	0.077 (8.8)	0.101 (23.4)	0.037 (21.4)	0.219 (20.1)	0.216 (21.4)	1.024 (186.5)
yes	12+CB	0.608 (63.6)	0.346 (17)	0.166 (17.1)	0.108 (7.7)	0.19 (17)	0.13 (21.5)	0.324 (15.6)	0.51 (28.5)	2.382 (188)
yes	12+EC	0.523 (55.7)	0.293 (21.6)	0.125 (17.3)	0.128 (11.9)	0.222 (24.4)	0.296 (30.6)	0.319 (39.7)	0.465 (33.4)	2.371 (234.6)
yes	12+CB+EC	0.404 (51.2)	0.208 (16.9)	0.182 (21.4)	0.162 (11.7)	0.147 (28.5)	0.132 (34.5)	0.39 (22.7)	0.479 (31)	2.104 (217.9)

Table A.7: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **concentration values of mix and single-salt data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								
EW	Outputs	NH_4^+	Ca^{2+}	Na^+	K^+	Cl^-	NO_3^-	Mg^{2+}	SO_4^{2-}	$\sum \text{Err}$
no	12	0.139 (79.2)	0.201 (23.2)	0.178 (33.8)	0.107 (10.4)	0.195 (34.5)	0.098 (78.2)	0.31 (30)	0.683 (51.1)	1.911 (340.4)
no	19	0.158 (100.3)	0.186 (21.5)	0.287 (48.1)	0.14 (11.1)	0.153 (26.4)	0.135 (85.1)	0.284 (35.5)	0.681 (80)	2.024 (408)
no	12+CB	0.111 (87.2)	0.221 (26.5)	0.166 (21.3)	0.08 (7.9)	0.145 (19.3)	0.098 (65.5)	0.296 (27.5)	0.517 (57.8)	1.634 (313)
no	12+EC	0.119 (90.1)	0.16 (18.6)	0.246 (43.4)	0.168 (15.1)	0.174 (29)	0.072 (39.8)	0.254 (18.7)	0.658 (65.2)	1.851 (319.9)
no	12+CB+EC	0.201 (142.9)	0.221 (16.2)	0.149 (27.1)	0.158 (11.5)	0.087 (30)	0.082 (50.8)	0.266 (17.4)	0.466 (49.5)	1.63 (345.4)
yes	12+CB	0.209 (60.6)	0.205 (18.2)	0.195 (43.7)	0.095 (10.3)	0.141 (22.7)	0.057 (19.7)	0.36 (24.7)	0.734 (80.9)	1.996 (280.8)
yes	12+EC	0.122 (46.8)	0.201 (19)	0.35 (50.8)	0.13 (11.1)	0.176 (27.5)	0.06 (28.9)	0.258 (28.7)	1.148 (103.1)	2.445 (315.9)
yes	12+CB+EC	0.134 (204.1)	0.234 (26.6)	0.173 (33.7)	0.084 (8.5)	0.265 (48.1)	0.047 (28.6)	0.249 (21.3)	1.14 (91)	2.326 (461.9)

Table A.8: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								\sum Err
EW	Outputs	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻	
no	12	2.003 (31.6)	0.203 (11.1)	0.506 (13.1)	0.116 (6.4)	0.518 (16.2)	0.527 (13.7)	0.587 (19.1)	0.708 (23)	5.168 (134.2)
no	19	0.577 (30.6)	0.225 (12.1)	0.207 (13.4)	0.188 (10.7)	0.229 (15.9)	0.517 (22.3)	0.72 (19.6)	0.561 (15.2)	3.224 (139.8)
no	12+CB	0.703 (35.3)	0.286 (22.6)	0.439 (29.2)	0.198 (13.8)	0.547 (41.1)	0.507 (24.8)	0.373 (38.1)	0.363 (32.2)	3.416 (237.1)
no	12+EC	0.55 (27.2)	0.298 (28.1)	0.556 (29.5)	0.283 (18.8)	0.487 (31)	0.692 (18.9)	0.345 (36.9)	0.432 (30.2)	3.643 (220.6)
no	12+CB+EC	0.73 (33.4)	0.392 (22.5)	0.769 (29.3)	0.258 (19.4)	0.878 (36.0)	0.444 (21.2)	0.526 (37.0)	0.418 (31.0)	4.415 (229.8)
yes	12+CB	0.401 (30.7)	0.364 (24.5)	0.429 (35.7)	0.218 (17.5)	0.463 (41.6)	0.492 (25.7)	0.402 (45.1)	0.47 (30.4)	3.239 (251.2)
yes	12+EC	0.575 (23.4)	0.273 (20.7)	0.57 (36)	0.247 (17.3)	0.562 (44.1)	0.765 (23.8)	0.417 (29.6)	0.445 (31.7)	3.854 (226.6)
yes	12+CB+EC	0.467 (34.2)	0.364 (29.8)	0.803 (30.7)	0.254 (21)	0.743 (35.2)	0.241 (31.9)	0.431 (25.7)	0.546 (33.7)	3.849 (242.2)

Table A.9: Concentration NRMSE and (MRE) (as %, mean of absolute value of relative errors) for each of 8 target ions. ANN architecture defined by outputs (12 ions, 19 outputs, or 12 ions with 1 or 2 constraints) and error weighting on constraints (EW). Architectures trained to **logarithm-transformed mix and single-salt data**; optimal network (highlighted in left column) selected using the MSE metric. Optimal results for each concentration are individually highlighted in the corresponding columns.

Architecture		Ion concentration NRMSE (MRE (%))								\sum Err
EW	Outputs	NH ₄ ⁺	Ca ²⁺	Na ⁺	K ⁺	Cl ⁻	NO ₃ ⁻	Mg ²⁺	SO ₄ ²⁻	
no	12	0.208 (15.8)	0.355 (22.7)	0.351 (30)	0.258 (19.8)	0.545 (24.8)	0.215 (21.3)	0.407 (31.9)	0.533 (48.1)	2.872 (214.4)
no	19	1.422 (361.3)	0.615 (47.9)	0.747 (61.3)	0.624 (51.1)	0.842 (60.9)	1.321 (186.6)	0.587 (56.7)	0.587 (44.8)	6.745 (870.6)
no	12+CB	0.465 (37.2)	0.787 (56.4)	1.264 (86.9)	0.827 (55.3)	1.188 (80.8)	0.342 (25.9)	0.852 (74.4)	0.915 (80.6)	6.64 (497.5)
no	12+EC	0.429 (26.2)	0.753 (57.1)	1.147 (86.1)	0.782 (51.9)	0.967 (74.8)	0.171 (38.9)	0.756 (65)	1.025 (82.2)	6.03 (482.2)
no	12+CB+EC	0.557 (54.7)	0.639 (39.6)	1.23 (75.8)	0.76 (50.4)	1.148 (66.5)	0.397 (36.9)	0.712 (49.9)	0.92 (75)	6.363 (448.8)
yes	12+CB	0.848 (46.4)	1.103 (79)	1.514 (94.8)	0.994 (76.3)	1.673 (95.3)	0.479 (41.2)	1.034 (92.3)	1.072 (94)	8.717 (619.3)
yes	12+EC	0.516 (37.6)	0.632 (58.7)	1.15 (77.1)	0.72 (48.6)	1.058 (74.9)	0.449 (62.7)	0.728 (71.9)	0.843 (75.3)	6.096 (506.8)
yes	12+CB+EC	0.401 (39.5)	1.041 (56.8)	1.479 (88.5)	0.842 (56.8)	1.675 (87.5)	0.421 (56.9)	0.983 (72.8)	1.073 (91.7)	7.915 (550.5)

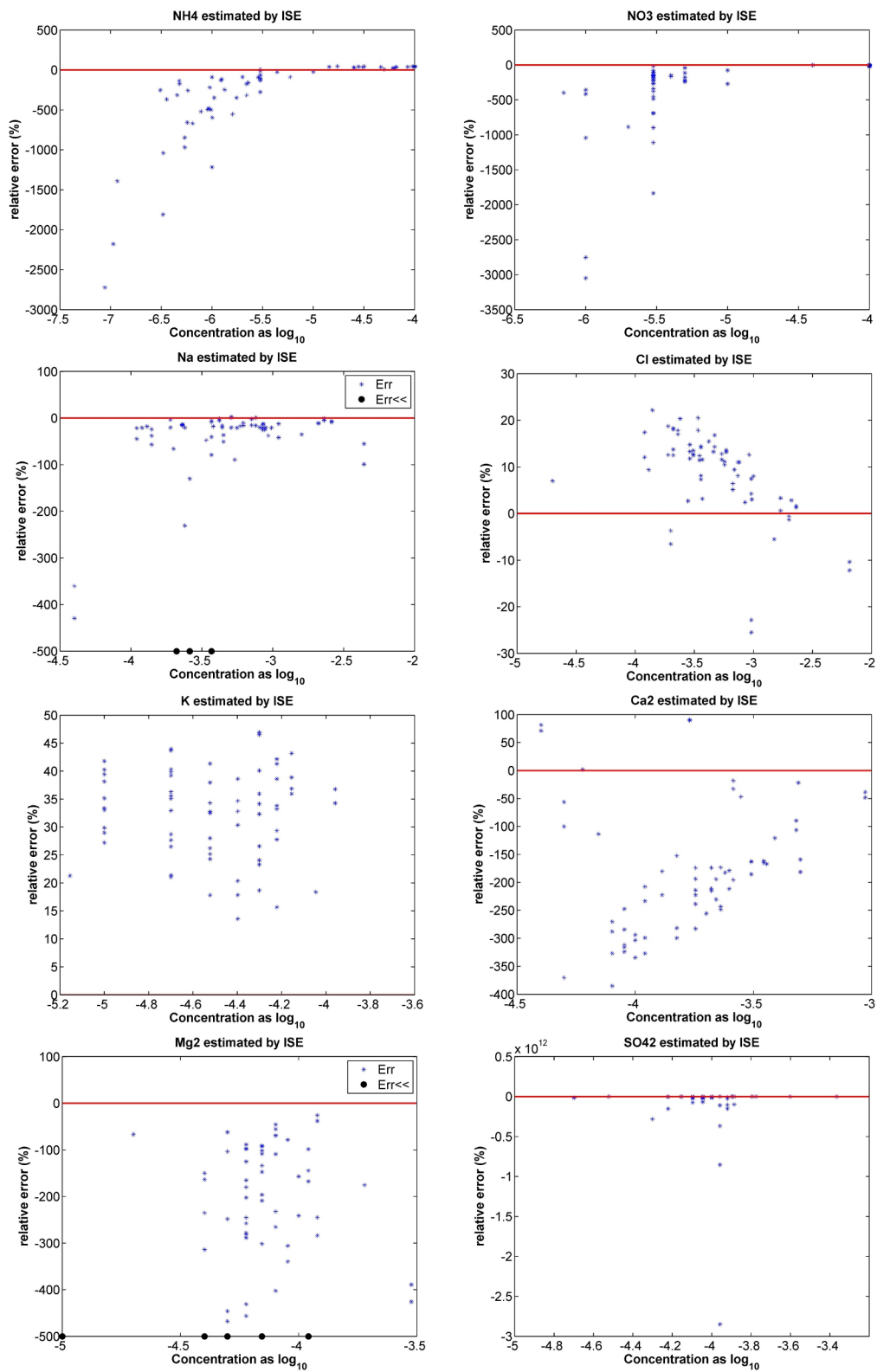


Figure A-5: Relative error (as %) for ISE-based predictions of ion concentrations.

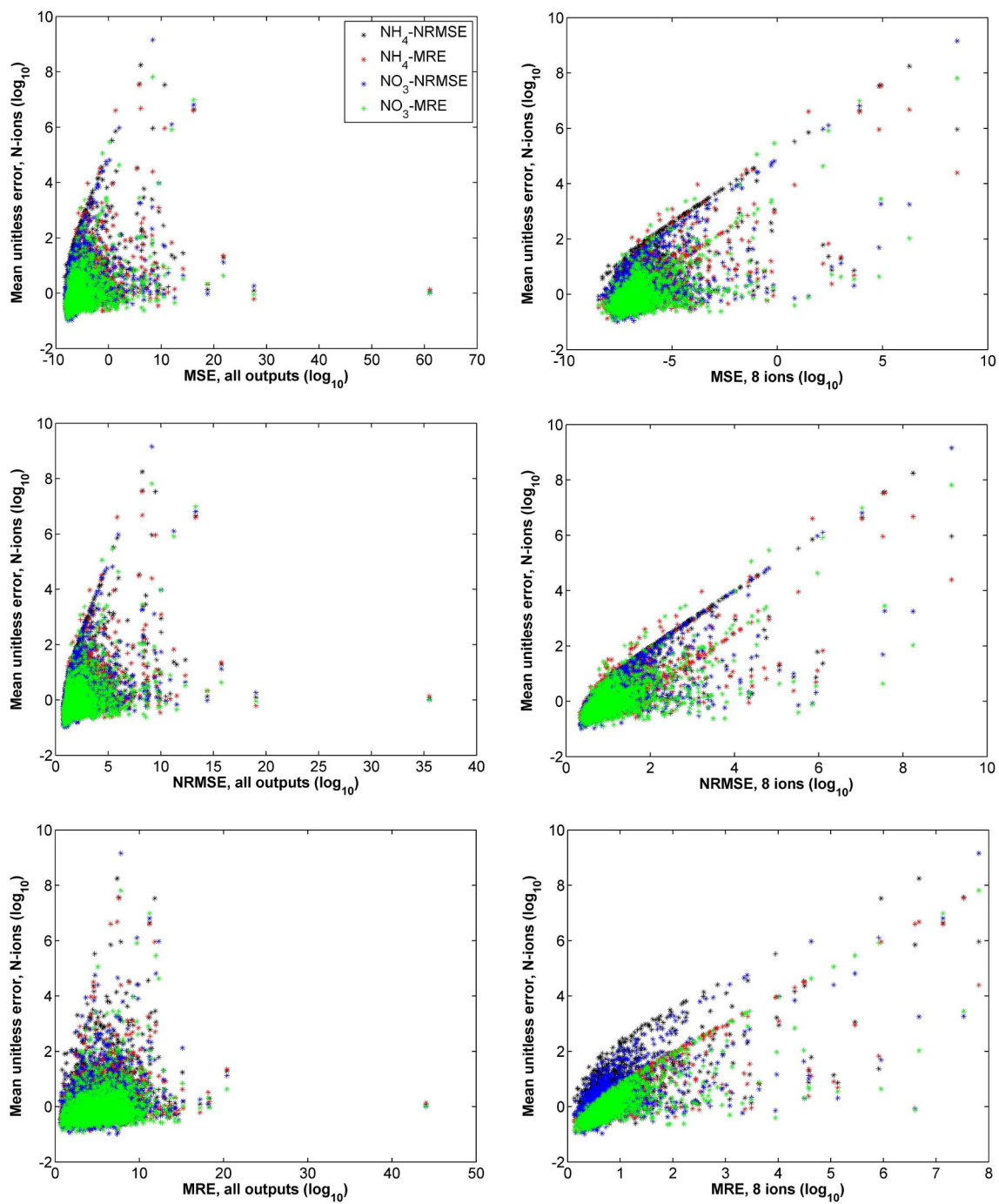


Figure A-6: Correlation of net goodness parameters with error calculations for nitrogen ions.

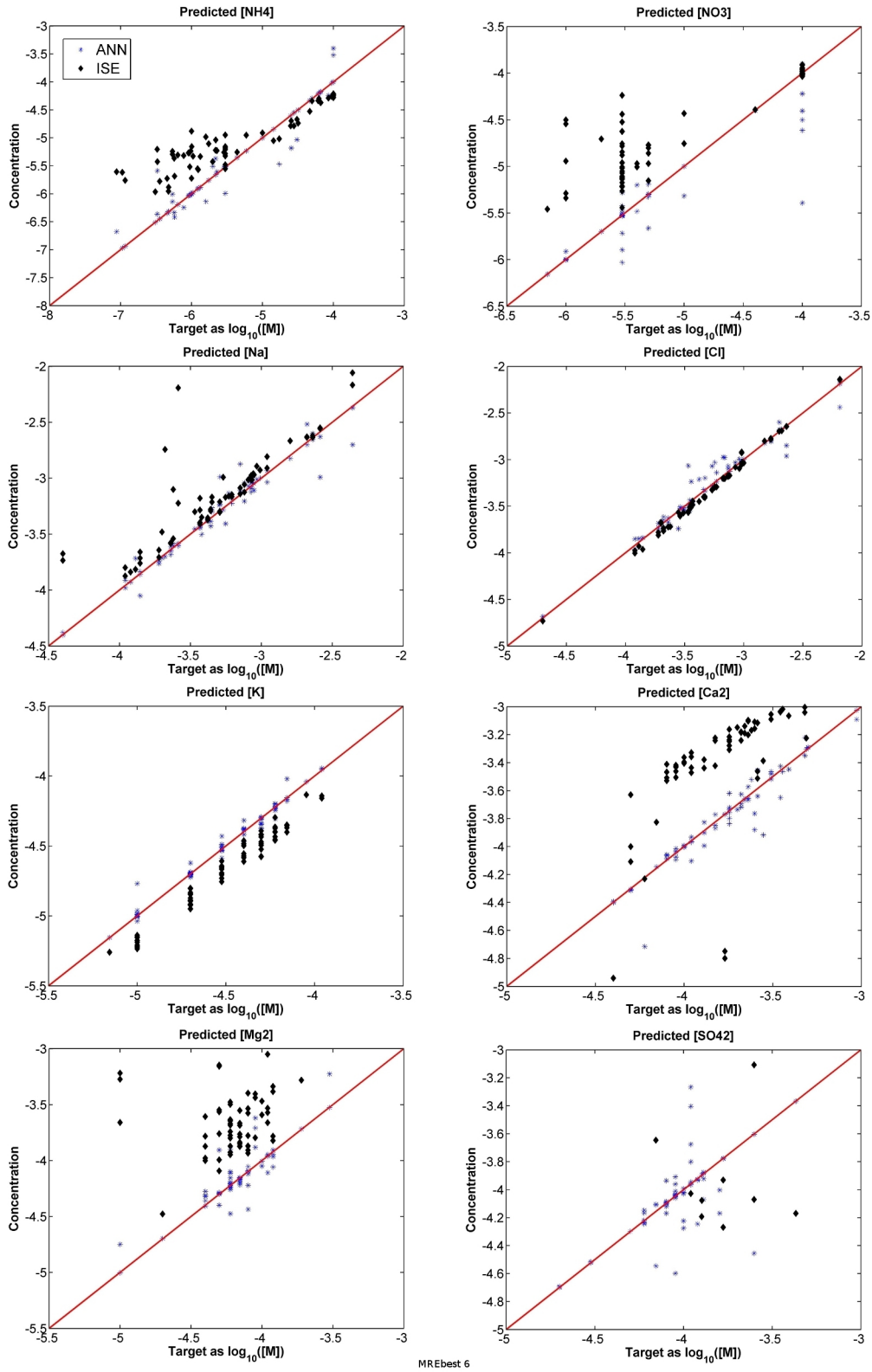


Figure A-7: Scatter plots of ion concentrations predicted using the optimal ANN (chosen using the MRE metric) as a function of target concentration.

Appendix B

Matlab code



B.1 Equilibrium determination

FindEquilib.m: Implements IUPAC recommendations for determining equilibrium.

```
1 function [status, values, dE_dt_series, d2E_dt2_series, starts] = ...
    FindEquilib(smoothSignal, sample_period, time_window, lim_dE_dt, ...
        dirtySignal, so, title_info, ...
            do_plots, single_plot_point, deriv_smoothing,
                smooth_win)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function finds equilibrium values following IUPAC recommendations as detailed in
% Macca, Analytica Chimica Acta 512 (2004)
% i.e. take reading when  $\Delta E / \Delta t = \pm\{0.1, 0.2, \text{ or } 0.4\}$  mV/min
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 % INPUTS
% smoothSignal of size [numP x smoothed_length]
% sample_period: in minutes/sample
% time_window: length (in seconds) of window over which lim_dE_dt must be maintained
% for equilib.
% lim_dE_dt: mV/min required for equilibrium to be declared
% dirtySignal: original signal (un-smoothed) – for graphing
% so: "signal_offset" = offset of smoothSignal from original signal – for graphing
% do_plot: 1 or 0, controls whether the plotting function is called (generally)
% single_plot_point: if only one graph is desired, this is [probe_num]; [-1] does
% all plots
21 %
% OUTPUTS
% status – indicates whether equilibrium was reached for all probes; size [1 x 1]
% values – equilibrium values; size [numP x 1]
```

```

% dE_dt_series - mean value of 1st derivative in region where "equilibrium" is
% determined
% d2E_dt2_series - mean value of 2nd derivative in region where "equilibrium" is
% determined

% options for plotting
if (do_plots)
    do_subplots = 0;
31   do_int_plots = 1;
    do_unsmoothed = 0;
else
    do_subplots = 0;
    do_int_plots = 0;
    do_unsmoothed = 0;
end % if (do_plots)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FUNCTION CONSTANTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41   if (lim_dE_dt < 0.1 | lim_dE_dt > 2)
        lim_dE_dt = 0.4; % mV/min
    end; % if (lim_dE_dt)

% how long does signal have to be stable for 'equilibrium' to be determined?
if (time_window < 1 | time_window > 60) % given in seconds
    time_window = 30;
end; % if (time_window)
equilib_seq_len = floor((time_window/60)/sample_period); % integer # samples in "
    time_window" seconds
51
    if (~exist('smooth_win'))
        smoothing_window = 41; % this choice is somewhat arbitrary (~30 sec)
    else
        smoothing_window = smooth_win;
    end % if

% indicate whether to do smoothing on derivative signals before calculating mean
% values for output
d2E_okay_limit = 0.25; % this choice is based on frequency plots from 4/22/10, page
    11-12 in lab book

61 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up variables
numP = size(smoothSignal,1);
values = NaN(numP,1);
dE_dt_series = NaN(numP,1);
d2E_dt2_series = NaN(numP,1);
starts = NaN(numP,1);
status = zeros(numP,1);

71 for p=1:numP
    % calculate first, second derivatives
    dE_dt = diff(smoothSignal(p,:)) / sample_period;
    d2E_dt2 = diff(dE_dt) / sample_period;

    % calculate smoothed versions of the derivative functions
    % appropriate smoothing_window length??
    w = window(@rectwin, smoothing_window);

    % these signals are offset from dE_dt and d2E_dt2 by (smoothing_window-1)/2
81   sm_dy = (1/smoothing_window)*conv(dE_dt,w,'valid');
    sm_d2y = (1/smoothing_window)*conv(d2E_dt2,w,'valid');
    offset = (smoothing_window-1)/2;

% for testing
if (0)
    figure;

```

```

    plot(dE_dt);
    figure;
    plot(d2E_dt2);
91    pause(3)
    return;
end % if (testing)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% find equilibrium value, if any %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% find regions where dE_dt < lim_dE_dt
if (deriv_smoothing) % use the smoothed version of the derivative for equilibrium
    determination
101    ind = find(abs(sm_dy) < lim_dE_dt);
    ind = ind + offset; % index into original signal
else % use unsmoothed version
    ind = find(abs(dE_dt) < lim_dE_dt);
end % if (deriv_smoothing)
l = length(ind);

% require signal remains < lim_dE_dt for 'equilib_seq_len' points in a row...
j = 0;
count = 0;
111    equilib_reached = 0;

if (0) % testing
    sm_dy
    ind
    smoothSignal(ind)
    pause(5)
end % if testing

% leave loop if/when 'count' hits 'equilib_seq_len'
121    while ( (j < l) & (~equilib_reached) )
        j = j+1;

        if (count == 0)
            count = 1;
            start = ind(j); % updates 'start' point of equilibrium sequence
        elseif (ind(j) == ind(j-1)+1) % index consecutive to previous
            count = count+1;
        else % index not consecutive
            count = 0;
131    end; % if

if (count >= equilib_seq_len)
    % check second derivative condition
    dE_mean_len = equilib_seq_len; % appropriate length for calculating mean of
    % derivs?
    if (~deriv_smoothing)
        [dE_mean, d2E_dt2_mean] = getD2mean(dE_mean_len, start, 0, dE_dt, d2E_dt2);
    else
        [dE_mean, d2E_dt2_mean] = getD2mean(dE_mean_len, start, offset, sm_dy,
            sm_d2y);
    end % if (~deriv_smoothing)
141
    d2E_dt2_mean = 0; % do not use checking limits
    % if (d2E_dt2_mean < d2E_okay_limit) % second_deriv within okay limits (+ side
    only)
    if (abs(d2E_dt2_mean) < d2E_okay_limit) % second_deriv within okay limits (+
    and -)
        equilib_reached = 1;
    else
        % go back halfway through this period & start looking again
        j = j - floor(equilib_seq_len/2);
        count = 0;
    end % if (second_deriv_okay)

```

```

151     end; % if
end; % while

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% output variables updated only if equilibrium was reached %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (equilib_reached)
    % equilibrium value determined by 'start' index above

    % average of start and end points of sequence of 'equilib_seq_len'
161     % tmp = (smoothSignal(p,start) + smoothSignal(p,start+equilib_seq_len))/2;

    % average of all points of sequence of 'equilib_seq_len'
    %display([num2str(start) ', ' num2str(equilib_seq_len)]);
    tmp = mean(smoothSignal(p,start:start+equilib_seq_len-1));

    % report with accuracy to 0.1mV
    values(p) = round(10*tmp)/10;

    % report how many samples it took to equilibrate
171     starts(p) = start;

    if (do_subplots | do_int_plots)
        if ( (single_plot_point == -1) | (single_plot_point == p) )
            % for displaying plots of signal & derivatives w/ equilb section marked
            non_var = 'we shouldnt be here'; % for testing
            win_end = start + equilib_seq_len - 1; % ending point in original signal
            sig_deriv_plots(do_subplots, do_int_plots, do_unsmoothed, ...
                dirtySignal, smoothSignal, dE_dt, sm_dy, d2E_dt2, sm_d2y, ...
                so, offset, start, win_end, tmp, p, lim_dE_dt, title_info);
181         end % if display plots
        end % if (displaying plots)

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % want to report something about behavior of dE/dt and dE2/dt2 %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % smooth vs. unsmoothed derivatives?

        % appropriate length for calculating mean of derivs?
        dE_mean_len = equilib_seq_len;
191         if (~deriv_smoothing)
            [dE_dt_series(p), d2E_dt2_series(p)] = getD2mean(dE_mean_len, start, 0, dE_dt
                , d2E_dt2);
        else
            [dE_dt_series(p), d2E_dt2_series(p)] = getD2mean(dE_mean_len, start, offset,
                sm_dy, sm_d2y);
        end % if (~deriv_smoothing)

    elseif (~equilib_reached & do_plots) % no equilibrium but want to plot anyway -
        for diagnostics
            if ( (single_plot_point == -1) | (single_plot_point == p) )

                non_var = 'no equilibrium was found'; % for testing
201
                xlim = size(smoothSignal,2);
                dirty_xlim = size(dirtySignal,2);
                probeset1 = {'Cl- ELIT', 'NH4 ELIT', 'Ca+ ELIT', 'Na+ ELIT', 'K+ ELIT', 'Na+ Or-n
                    ', 'pH Orion', ...
                    'Cl- Hann', 'Hardness'};
                    probeset1 = {'Cl- ELIT', 'NH4 ELIT', 'Ca+ ELIT', 'Na+
                        ELIT', 'K+ ELIT', 'Na+ Or-n', 'pH Orion', ...
                            'NO3 ELIT', 'Cl- Hann', 'Hardness', 'SO4 Fshr', 'CO3 Thom'}; % Feb
                                2012, extended set
                title.text = [title_info ': probe ' probeset1{p}];

                figure; % display data with smoothed derivative signals, no equilibrium
                    window shown

```

```

211     [AX,H1,H2] = plotyy(so+[1:xlim],smoothSignal(p,:),so+offset+1+[1:length(sm_dy
        )],sm_dy,'plot');
        set(get(AX(1),'Ylabel'),'String','ISE response [mV]');
        set(get(AX(2),'Ylabel'),'String','dE/dt [mV/min]');
        xlabel('Index within sample window');
        set(H1,'LineStyle','-');
        set(H1,'Color','b');
        set(H2,'LineStyle','-');
        set(H2,'Color','r');

        hold(AX(1),'on');
221     hold(AX(2),'on');
        plot(AX(1),dirtySignal(p,),'k-');
        plot(AX(2),so+offset+2+[1:length(sm_d2y)],0.2*sm_d2y,'g-');
        set(AX(2),'YLim',[-1 1]);
        legend('SmoothSig','OrigSig','sm-dE/dt','0.2*sm-d2E/dt2');
        title([title_text ' — smoothed derivatives']);

        plot(AX(2),zeros(dirty_xlim,1),'k-','LineWidth',2); % zero limit for
            derivatives
        plot(AX(2),lim_dE_dt*ones(dirty_xlim,1),'k-','LineWidth',1); % upper limit
            for dE.dt
        plot(AX(2),-1*lim_dE_dt*ones(dirty_xlim,1),'k-','LineWidth',1); % lower
            limit for dE.dt
231     end % if single plot point var is ok
        end; % if equilib_reached
    end; % for p

    if (length(find(isnan(values))) > 0)
        status = 0;
    else
        status = 1;
    end; % if

241 end % function FindEquilib()

function [d1_mean, d2_mean] = getD2mean(dE_mean_len, start, offset, deriv1, deriv2)
    % start points into original signal
    % offset = 0 for unsmoothed arrays
    % offset = offset for smoothed arrays

    % where 'start' point should reference in these arrays' indeces
    % note: window is smaller than dE_mean_len if st==1 because of following statements
    st = max((start-1) - offset, 1); % account for case where equilibrium is very
        early
251     st2 = max(st-1, 1);

    % where end of the window should be, in these arrays' indeces
    win_end = start + dE_mean_len - 1; % ending point in original signal
    win_end = win_end - offset - 1; % ending point in the (first) derivative signals

    if (win_end > length(deriv1))
        d1_mean = mean(deriv1(st:end));
    else
        d1_mean = mean(deriv1(st:win_end));
261     end % if

    % offset from signal by 2 (offset from 1st deriv by 1)
    if (win_end-1 > length(deriv2))
        d2_mean = mean(deriv2(st2:end));
    else
        d2_mean = mean(deriv2(st2:win_end-1));
    end % if
end % function getD2mean()

```

B.2 Environmental PDF and sample creation

makeFreshwaterPDF2.m: Creates an 8-D probability distribution function for ions of interest based on USGS data from 5 New England states.

```

1 % Goal: Make a joint pdf for all of the freshwater samples I have
% (CT, MA, NH, etc.) such that we'll be able to sample out of that
% joint pdf.
% Also: Weight the pdf based on the uncertainty values of
% electrodes we have to measure those analytes and at those levels.

% Amy Mueller, starting 12/16/11 - continuing 12/29/11, 1/3/12

% 1/3/12 - jointPDF representation changed for more efficient
% representation, i.e., not N-D sparsely filled matrix. Now
11 % represented as a structure with fields ind1, ind2, ... (one per
% analyte) and counts - for "height" of PDF at that index location

clear;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% See procStateData2_simultOnly.m
analyte_list = {'conductivity', 'pH', 'NH4+', 'Ca2+', ...
               'Na+', 'K+', 'Cl-', 'NO3-', 'Mg2+', 'SO42-', 'H+'};
regunits = {'uSiemens/cm', 'pH', 'M', 'M', 'M', 'M', ...
            'M', 'M', 'M', 'M', 'M'};
21 units = {'log_1_0(uSiemens/cm)', 'pH', 'log_1_0([M])', 'log_1_0([M])', ...
            'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])', ...
            'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])'};

param_codes = {95, 403, 608, 915, 930, 935, 940, 618, 925, 945, 191};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

numbins = 9;

31 cond_bins = [1:0.25:6]; % log uSiemens/cm
pH_bins = [4:0.25:8]; % pH units -> we should use either pH or H+,
% not both !! Seems like H+ has more data
% than pH, so we'll use that one
env_datapath = ['C:\Documents and Settings\Amy\My Documents\Research\' ...
               'thesis\usgs data - for range detection\'];

% get USGS data for New England - simultaneous samples only - from file
load([env_datapath 'NewEnglandAll-Simultaneous-29-Dec-2011.mat'], 'analytedata');

41 % for testing
%analyte_list = {'Ca2+', 'Na+', 'K+', 'Cl-'};
%param_codes = {915, 930, 935, 940};

% First - create the joint pdf
% Using only the 8 ions - assume conductivity & pH will fall out
% from there.
analyte_list = {'NH4+', 'Ca2+', 'Na+', 'K+', 'Cl-', 'NO3-', 'Mg2+', 'SO42-'};
regunits = {'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'};
51 units = {'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])', ...
            'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])', 'log_1_0([M])'};
param_codes = {608, 915, 930, 935, 940, 618, 925, 945};

% get fieldnames
all_fieldnames = fieldnames(analytedata);

% create fieldnames (subset of all fields) from analyte list
th_fieldnames = {};
for tmp = 1:length(analyte_list)
61 th_name = analyte_list{tmp};
    if (isequal(th_name(end), '+') | isequal(th_name(end), '-'))

```

```

        th_name = th_name(1:end-1); % matlab won't allow names ending with these
            characters
    end % if
    th_fieldnames = {th_fieldnames{:} th_name};
end % for

for tmp = 1:length(analyte_list)
    [mult minmax] = getEnvRange(analyte_list{tmp});
    % NEED TO PUT DATA IN THE CORRECT UNITS!!!!
71    % Changing them all from mg/L to mol/L
    th_name = th_fieldnames{tmp};
    eval(['analytedata = setfield(analytedata, '' th_name '' , mult*analytedata.'
        th_name ');']);
    concbincenters(tmp,:) = minmax(1) + [0:numbins]*(minmax(2)- minmax(1))/numbins;
    concbinedges(tmp,:) = minmax(1) + (0.5+[0:numbins-1]*(minmax(2)- minmax(1))/
        numbins);
end % for

% Remove any invalid data points - right now this means ANY missing
% data on ANY channel
for tmp2 = 1:length(th_fieldnames)
81    th_str = th_fieldnames{tmp2};
    eval(['keepers = find(~isnan(analytedata.' th_str ') & ~isempty(analytedata.'
        th_str ');']);
    for tmp3 = 1:length(all_fieldnames)
        th_str = all_fieldnames{tmp3};
        eval(['analytedata = setfield(analytedata, '' th_str '' , analytedata.' ...
            th_str '(keepers));']); % keep only the appropriate subset
    end % for
end % for

display(['Number of valid data points: ' num2str(length(analytedata.conductivity))]
);
91

% Create the joint PDF.
% Instead of a N-D matrix (which is sparsely populated),
% create a struct with fields: ind1, ind2, ..., indN + counts
% Save each populated bin indexed by ind1, ind2, ..., with
% corresponding counts.
% e.g. PDF.ind1 = [1 1]
%     PDF.ind2 = [2 1]
%     PDF.ind3 = [3 3]
%     PDF.counts = [45 44]
101 % And it is assumed that all unlisted index combos have counts = 0

jointPDF = struct; % empty struct
for tmp_ind = 1:length(th_fieldnames)
    fname = ['ind' num2str(tmp_ind)];
    eval(['jointPDF = setfield(jointPDF, '' fname '' ,[]);']);
end % for
jointPDF = setfield(jointPDF, 'counts', []);

jointPDF = findPDFvalue2(analytedata, th_fieldnames, 1, jointPDF, ...
111     zeros(size(analyte_list)), numbins+1, concbinedges); %
    recursive call

for q=1:length(jointPDF.counts)
    display(['Value of PDF at: (' num2str(jointPDF.ind1(q)) ' ' num2str(jointPDF.ind2
        (q)) ' ' ...
            ' num2str(jointPDF.ind3(q)) ' ' num2str(jointPDF.ind4(q)) '
            , ...
            num2str(jointPDF.ind5(q)) ' ' ...
            num2str(jointPDF.ind6(q)) ' ' ...
            num2str(jointPDF.ind7(q)) ' ' ...
            num2str(jointPDF.ind8(q)) ') - number of ' ...
            'counts = ' num2str(jointPDF.counts(q))]);
121 end % for

```

```

numPointsStored = sum(jointPDF.counts);
display(['Number of valid points counted: ' num2str(numPointsStored)]);

envPDF = jointPDF;
envPDF.counts = envPDF.counts/numPointsStored; % sums to 1

save(['env_datapath 'NewEngland_Simultaneous_PDF.mat'], 'analytedata', ...
    'envPDF', 'numPointsStored'); % staged saving
131
% clear things we don't need
clear analytedata keepers envPDF

multipliers = NaN(size(jointPDF.counts));
mult_count = zeros(size(jointPDF.counts)); % indicates how many
                                         % multipliers have been averaged

% Now we need to include scaling due to magnitude, uncertainty of ISE measurements
for j = 1:length(analyte_list)
141     tmp_param = param_codes{j}; % USGS parameter code

    % response for named ISE
    [th_mag th_stderr] = get_ise_resp(tmp_param, 10.^concbinccenters(j,:), 'main');

    % don't want negative numbers from anions
    th_mag = abs(th_mag);

    if (length(find(isnan(th_mag)))==0) % we had a corresponding sensor
        th_multiplier = th_stderr;
        % for non-baseline cases, want to include magnitude of response
        % in multiplication factor
        non_bl = find(th_mag ~= 0);
        tmp_mults = th_stderr./(th_mag/25); % note: 20 is an arbitrary
                                         % choice, is matching order
                                         % of magnitude of std_error #s
        th_multiplier(non_bl) = tmp_mults(non_bl); % length = num_bins+1
        clear tmp_mults non_bl;

        display(['Multipliers for analyte = ' num2str(j) ': ' mat2str(th_multiplier)
            ]);
161
        % update the main 'multipliers' matrix
        for k = 1:length(th_multiplier)
            mult = th_multiplier(k);
            eval(['pdf_subset = find(jointPDF.ind ' num2str(j) '== ' ...
                num2str(k) ');']);

            for n = 1:length(pdf_subset)
                ind = pdf_subset(n);
                if (mult_count(ind)==0)
171                    multipliers(ind) = mult;
                    mult_count(ind) = 1;
                elseif (mult_count(ind)==1)
                    multipliers(ind) = sqrt(mult^2 + multipliers(ind)^2); % root of
                        squares
                    %multipliers(ind) = (multipliers(ind)+mult)/3; % average+
                    mult_count(ind) = mult_count(ind)+1;
                else
                    mult_count(ind) = mult_count(ind)+1;
                    multipliers(ind) = sqrt(mult^2 + multipliers(ind)^2); % root of
                        squares
                    %multipliers(ind) = (multipliers(ind)*mult_count(ind)+mult)/(
                        mult_count(ind)+1); %average+
181                end % if
            end % for n
            clear pdf_subset;
        end % for
    end % if
end % for

```



```

% new variables so we can compare results
multISE_multipliers = multipliers;
multISE_mult_count = mult_count;
191 for j=1:4 % be sure we only do this for cations we have measurements for
    % response for non-named ISEs
    tmp_param = param_codes{j}; % USGS parameter code
    [th_mags th_stderrs] = get_ise_resp(tmp_param, 10.^concbinceners(j,:), 'others')
    ;

    for k2=1:size(th_mag,1)
        th_mag = squeeze(th_mags(k2,:));
        th_stderr = squeeze(th_stderrs(k2,:));

        th_multiplier = th_stderr;
201 % for non-baseline cases, want to include magnitude of response
        % in multiplication factor
        non_bl = find(th_mag ~= 0);
        tmp_mults = th_stderr./(th_mag/25); % note: 20 is an arbitrary
                                           % choice, is matching order
                                           % of magnitude of std_error #s
        th_multiplier(non_bl) = tmp_mults(non_bl);

        for k = 1:length(th_multiplier)
            mult = th_multiplier(k);
211 eval(['pdf_subset = find(jointPDF.ind' num2str(j) '==' ...
                num2str(k) ');']);

            for n = 1:length(pdf_subset)
                ind = pdf_subset(n);
                if (multISE_mult_count(ind)==0)
                    multISE_multipliers(ind) = mult;
                    multISE_mult_count(ind) = 1;
                elseif (multISE_mult_count(ind)==1)
                    multISE_multipliers(ind) = sqrt(mult^2 + multISE_multipliers(ind)
                    ^2); % root of squares
221 %multpliers(ind) = (multipliers(ind)+mult)/3; % average+
                    multISE_mult_count(ind) = multISE_mult_count(ind)+1;
                else
                    multISE_mult_count(ind) = multISE_mult_count(ind)+1;
                    multISE_multipliers(ind) = sqrt(mult^2 + multISE_multipliers(ind)
                    ^2); % root of squares
                    %multpliers(ind) = (multipliers(ind)*mult_count(ind)+mult)/(
                    mult_count(ind)+1); %average+
                end % if
            end % for n
            clear pdf_subset;
        end % for k
231 end % for k2
    end % for j

    figure; hold on;
    plot(multipliers, 'r');
    plot(multISE_multipliers, 'b');
    legend('Single ISE multipliers', 'All ISE multipliers');

% apply the multipliers to the original PDF
pdfIntegral = sum(jointPDF.counts .* multipliers);
241 jointPDF.scaled1 = (jointPDF.counts .* multipliers)/pdfIntegral; % sums to 1

pdfIntegral = sum(jointPDF.counts .* multISE_multipliers);
jointPDF.scaled2 = (jointPDF.counts .* multISE_multipliers)/pdfIntegral; % sums to 1

figure; hold on;
plot(jointPDF.counts/numPointsStored, 'r');
plot(jointPDF.scaled1, 'b');
plot(jointPDF.scaled2, 'k');
legend('No scaling', '1 ISE', 'All ISEs');

```

```

251 save([env_datapath 'NewEngland_Simultaneous_PDF.mat'], ...
      'jointPDF', '-append');

save([env_datapath 'NewEngland_Simultaneous_PDF.mat'], ...
     'env_datapath', 'analyte_list', 'regunits', 'th_fieldnames', ...
     'concbincenters', 'concbinedges', 'multipliers', 'multISE_multipliers', '-append
     ');

261 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Post-processing area - just want to get some stats & record what %
% I did. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:8
    eval(['all_indices(i,:) = jointPDF.ind' num2str(i) ' ']);
end

maxheight = max(jointPDF.ind1);

271 figure; hold on;
h = bar3(all_indices);
for i=1:length(h)
    % set(h(i), 'MarkerEdgeColor', 'none');
    set(h(i), 'EdgeAlpha', [0]);
    cdata = get(h(i), 'CData');
    zdata = get(h(i), 'ZData');
    for j=1:length(cdata)/6
        cdata((j-1)*6+1:6*j, :) = zdata((j-1)*6+2,2)/maxheight;
281    end % for
    set(h(i), 'Cdata', cdata);
end % for
a = axis;
axis([a(1) length(jointPDF.ind1) a(3)+0.5 a(4)-0.5]);
nicify_graph(gcf, gca, 16);

figure;
hist(all_indices ' ');
291 legend(th_fieldnames);
nicify_graph(gcf, gca, 16);

figure;
plot(jointPDF.counts, '.');
nicify_graph(gcf, gca, 16);

% let's look at where the bulk of the probability lies
cutoff = numPointsStored/100;
301 majorPoints = find(jointPDF.counts>cutoff); % have > 1% of PDF @
% each point
frac = sum(jointPDF.counts(majorPoints))/numPointsStored % fraction of PDF
% contained in
% these points alone

for i=1:length(majorPoints)
    th_str = '';
    for j=1:length(th_fieldnames)
        th_tmp = eval(['jointPDF.ind' num2str(j) '(majorPoints(i))']);
311    th_str = [th_str num2str(th_tmp) ' '];
    end % if
    display(['Indicies: (' th_str '), count = ' ...
            num2str(jointPDF.counts(majorPoints(i)))]);
end % for

while (frac < 0.8)

```

```

        cutoff = cutoff - 0.5;
        th_points = find(jointPDF.counts > cutoff);
        frac = sum(jointPDF.counts(th_points))/numPointsStored
321 end % while

% now we have points accounting for 90% of the PDF
figure; hold on;
h = bar3(all_indices(:, th_points));
for i=1:length(h)
    % set(h(i), 'MarkerEdgeColor', 'none');
    set(h(i), 'EdgeAlpha', [0]);
    cdata = get(h(i), 'CData');
    zdata = get(h(i), 'ZData');
331 for j=1:length(cdata)/6
        cdata((j-1)*6+1:6*j, :) = zdata((j-1)*6+2,2)/maxheight;
    end % for
    set(h(i), 'Cdata', cdata);
end % for
a = axis;
axis([a(1) length(th_points) a(3)+0.5 a(4) -0.5]);
nicify_graph(gcf, gca, 16);
title(['Points accounting for ' num2str(frac) ' of PDF weight']);

341 saveas(gcf, [env_datapath '\figures\' 'SubPoints_80perc_PDFweight.jpg']);
print(gcf, '-djpeg', '-r300', [env_datapath '\figures\' 'SubPoints_80perc_PDFweight-
HighRes.jpg']);

th_points = [];
while (length(th_points) < 50)
    cutoff = cutoff - 0.5;
    th_points = find(jointPDF.counts > cutoff);
    frac = sum(jointPDF.counts(th_points))/numPointsStored
end % while

351 % now we have points accounting for 90% of the PDF
figure; hold on;
h = bar3(all_indices(:, th_points));
for i=1:length(h)
    % set(h(i), 'MarkerEdgeColor', 'none');
    set(h(i), 'EdgeAlpha', [0]);
    cdata = get(h(i), 'CData');
    zdata = get(h(i), 'ZData');
    for j=1:length(cdata)/6
        cdata((j-1)*6+1:6*j, :) = zdata((j-1)*6+2,2)/maxheight;
361 end % for
    set(h(i), 'Cdata', cdata);
end % for
a = axis;
axis([a(1) length(th_points) a(3)+0.5 a(4) -0.5]);
nicify_graph(gcf, gca, 16);
title(['Points accounting for ' num2str(frac) ' of PDF weight']);

saveas(gcf, [env_datapath '\figures\' 'SubPoints_80perc_PDFweight2.jpg']);
print(gcf, '-djpeg', '-r300', [env_datapath '\figures\' 'SubPoints_80perc_PDFweight2-
HighRes.jpg']);
371

% man points have only a single count
length(find(jointPDF.counts==1)) % these are points of SMALL PROBABILITY

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% want to look at how much the PDF changes due to ISE-based scaling

381 no_scaling = jointPDF.counts/sum(jointPDF.counts);
main_scaling = jointPDF.scaled1;

```

```

all_scaling = jointPDF.scaled2;

figure; hold on;
plot(log10(no_scaling), 'r');
plot(log10(main_scaling), 'g');
plot(log10(all_scaling), 'b');
nicify_graph(gcf, gca, 16);
391 legend('none', 'main', 'all');

mean(abs(main_scaling-no_scaling))

mean(abs(all_scaling-no_scaling))

mean(abs(all_scaling-main_scaling))

401 % Does sorting illuminate anything?
[no_scaling Z] = sort(no_scaling);
main_scaling = main_scaling(Z);
all_scaling = all_scaling(Z);

figure; hold on;
plot(log10(no_scaling), 'r');
plot(log10(main_scaling), 'g');
plot(log10(all_scaling), 'b');
nicify_graph(gcf, gca, 16);
411 legend('none', 'main', 'all');

```

selectPDFpoints.m: Randomly chooses points from the 8-D joint PDF.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The goal of this file is to choose the training sample mixtures
% for the ANN training, based on joint PDF of environmental data.
%
% use either jointPDF.scaled1 - just primary ISEs taken into account
% or      jointPDF.scaled2 - non-primary effects accounted for
% or      envPDF - no ISE scaling taken into account
9 %
% Use method of: choose uniform random number [0-1]
% Use this to reference into the CDF
% Back out where that was in the PDF
% Use that to choose ranges for each of the parameters.
% From there choose within the bin for each parameter based on that
% parameter's individual PDF.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% load the data
19 clear;
env_datapath = ['C:\Documents and Settings\Amy\My Documents\Research\' ...
               'thesis\usgs data - for range detection\'];
load([env_datapath 'NewEngland.Simultaneous.PDF.mat']);

% set up variables
numAnalytes = length(th_fieldnames);
numbins = size(concbincenters, 2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 % SET BY USER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of random samples we want to choose
numSamples = 50;

% choose which scaled PDF to use

```

```

usedPDF = jointPDF.scaled1; % this is just scaling due to primary ISE effects

testing = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39

% Make the CDF
jointCDF = NaN(size(usedPDF));
for i=1:length(usedPDF)
    if (i==1)
        jointCDF(i) = usedPDF(i);
    else
        jointCDF(i) = usedPDF(i) + jointCDF(i-1);
    end % if
49 end % for

% Choose random numbers & index them into the CDF
pdfSamples = zeros(1,numSamples);
for j=1:numSamples
    th_rand = rand;
    cdfbin = 1;
    while (th_rand > jointCDF(cdfbin))
        cdfbin = cdfbin + 1;
59 end % while
    pdfSamples(j) = cdfbin;
end % for

% Identify the bin ranges in from the PDF
all_ranges = NaN(numAnalytes, numSamples, 2); % (min,max)
all_inds = NaN(numAnalytes, numSamples);
all_outputs = NaN(numAnalytes, numSamples);
for j=1:numSamples
69     for k=1:numAnalytes
        eval(['th_index = jointPDF.ind' num2str(k) '(pdfSamples(j));']);

        % determine range for this analyte
        if (th_index==1)
            th_bin_start = 0; % [M], there shouldn't be any <0 (non-physical)
        else
            th_bin_start = 10^concbinedges(k,th_index-1); % [M]
        end % if
        if (th_index==(numbins+1))
79             th_bin_end = 10^8; % [M]
        else
            th_bin_end = 10^concbinedges(k,th_index); % [M]
        end % if

        all_ranges(k,j,:) = [th_bin_start th_bin_end];
        all_inds(k,j) = th_index;
    end % for k
end % for j

89

% Use that to choose ranges for each of the parameters.
% Use paramter (independent) PDF to choose sub-bin.
% From there choose uniformly within the sub-bin for each parameter
numsubbins = 9;
for k=1:numAnalytes
    % first pull out details of data
    eval(['subdata = analytedata.' cell2mat(th_fieldnames(k)) ';'']);
    th_inds = all_inds(k,:);
99     for bin_index = 1:numbins+1
        % find which samples of this analyte correspond to this bin
        th_samples = find(th_inds == bin_index);

```

```

if (~isempty(th_samples)) % continue processing
    % get the PDF for only that analyte for only that bin

    % determine range for that bin
    if (bin_index==1)
        bin_start = 0; % [M], there shouldn't be any <0 (non-physical)
    109     else
        bin_start = 10^concbinedges(k,bin_index-1); % [M]
    end % if
    if (bin_index==(numbins+1))
        bin_end = 10^8; % [M]
    else
        bin_end = 10^concbinedges(k,bin_index); % [M]
    end % if

    % find subset of data to sub-bin
    119     keepers = find((subdata >= bin_start) & (subdata < bin_end));
    subdata2 = subdata(keepers);

    % building the PDF by sub-bin
    subbinedges = log10(bin_start) + ...
        (0.5+[0:numsubbins-1])*(log10(bin_end) - log10(bin_start))/numsubbins
        ;
    subbinhist = zeros(1,numsubbins+1);
    for m=1:numsubbins+1
        if (m==1)
            sub_bin_start = 0; % [M], there shouldn't be any <0 (non-
            129                physical)
        else
            sub_bin_start = 10^subbinedges(m-1); % [M]
        end % if
        if (m==(numsubbins+1))
            sub_bin_end = 10^8; % [M]
        else
            sub_bin_end = 10^subbinedges(m); % [M]
        end % if
        keepers = find((subdata2 >= sub_bin_start) & (subdata2 < sub_bin_end)
            );
        subbinhist(m) = length(keepers);
    139     end % for

    if (testing)
        figure; plot(sort(subdata2))
        hold on;
        plot((10.^subbinedges'*ones(1,length(subdata2))))'
    end % if

    % set up the CDF
    subbinhist = subbinhist / sum(subbinhist);
    149     subbinCDF = NaN(size(subbinhist));
    for i=1:length(subbinhist)
        if (i==1)
            subbinCDF(i) = subbinhist(i);
        else
            subbinCDF(i) = subbinhist(i) + subbinCDF(i-1);
        end % if
    end % for i

    % Need to choose concentrations for all cases where
    159     % this analyte has this bin number assigned to it
    % (those represented in 'th_samples').

    % choose a random number, index into the CDF
    randConcs = zeros(1,length(th_samples));
    for j=1:length(th_samples)
        th_rand = rand;
        cdfbin = 1;
        while (th_rand > subbinCDF(cdfbin))

```

```

169         cdfbin = cdfbin + 1;
        end % while

        % Now we know the subbin to use,
        % choose uniformly from this sub-bin
        if (cdfbin==1)
            th_start = log10(bin_start); % log10[M], there shouldn't be any
            <0 (non-physical)
        else
            th_start = subbinedges(cdfbin-1); % log10[M]
        end % if
        if (cdfbin==(numsubbins+1))
179         th_end = log10(bin_end); % log[M]
        else
            th_end = subbinedges(cdfbin); % log[M]
        end % if
        randConcs(j) = th_start + rand*(th_end - th_start);
    end % for

    all_outputs(k,th_samples) = randConcs;

    end % if ~isempty
189 end % for bin_index
end % for k

% change to [M]
finalSamples = 10.^all_outputs;

% round to integer uM, 10uM, etc.
for i=1:numAnalytes
    for j=1:numSamples
        tmp = all_outputs(i,j);
199         if (tmp < -6)
            finalSamples(i,j) = (10^-7)*floor(finalSamples(i,j)*10^7); % to 0.1uM
            precision
        elseif (tmp < -5)
            finalSamples(i,j) = (10^-6)*floor(finalSamples(i,j)*10^6); % to 1uM
            precision
        elseif (tmp < -3)
            finalSamples(i,j) = (10^-5)*floor(finalSamples(i,j)*10^5); % to 10uM
            precision
        else
            finalSamples(i,j) = (10^-4)*floor(finalSamples(i,j)*10^4); % to 100uM
            precision
        end % if
    end % j
209 end % for

% for second 25, want to set NH4, NO3 = 10^-5.5 or 10^-4 (low and
% high) w/ same other concentrations
finalSamples(:,numSamples+1:numSamples+25) = finalSamples(:, numSamples-24:numSamples
);
finalSamples([1 6], numSamples+1:numSamples+25) = 10^-4; % high
finalSamples([1 6], numSamples-24:numSamples) = 3*10^-6; % low = rounded(10^-5.5) to
1uM

figure; hold on; plot((log10(finalSamples)))')
219 %figure; hold on; plot(all_outputs')
legend(th_fieldnames, 'location', 'NorthEastOutside')
nicify_graph(gcf,gca,16);
saveas(gcf,[env_datapath '\figures\' 'chosenPoints.jpg']);
print(gcf, '-djpeg', '-r300', [env_datapath '\figures\' 'chosenPoints-HighRes.jpg']);

display('Printing concentrations for points:')
display([' as: ' th_fieldnames]);
for i=1:numSamples-25

```

```

229     th_str = [];
        for j=1:numAnalytes
            th_str = [th_str num2str(finalSamples(j,i)) ', '];
        end % for
        display(['          (' th_str(1:end-2) ')']);
    end % for

    display([' ']);
    display('Low-N concentrations:');
    for i=numSamples-24:numSamples
239     th_str = [];
        for j=1:numAnalytes
            th_str = [th_str num2str(finalSamples(j,i)) ', '];
        end % for
        display(['          (' th_str(1:end-2) ')']);
    end % for

    display([' ']);
    display('High-N concentrations:');
    for i=numSamples+1:numSamples+25
249     th_str = [];
        for j=1:numAnalytes
            th_str = [th_str num2str(finalSamples(j,i)) ', '];
        end % for
        display(['          (' th_str(1:end-2) ')']);
    end % for

    save([env_datapath 'NewEngland_TrainingSamples.mat'], 'finalSamples', ...
        'th_fieldnames', 'all_ranges', 'all_inds', 'all_outputs');

```

stdRecipes3.m: Specifies ‘recipes’ for creating environmentally-representative ion mixes selected from the joint PDF.

```

% This version updates from use of Mg(OH)2 —> too low of
% solubility (oops) to use of MgSO4 and Mg(NO3)2
3 %
% 1/19/12 – solubility of CaCO3 is also TOO LOW
% (But solubility of Na2CO3 is really high – so prioritize CaCl2
% over NaCl.)
%
% 2/2/12 – Used 1.2mM MgCO3 instead of 4mM – updating.

function [saltVolumes saltList H OH CO3 error] = ...
    stdRecipes3(Na, K, Ca, Mg, NH4, Cl, SO4, NO3, Vtotal, useOldWay)

13 % Inputs:  [uM] target concentrations for each analyte
%           Vtotal is the target volume in [mL]
% Outputs:  [mL] of 100mM standard required to achieve target conc's

% Note:  need to have more standards than otherwise required
% (for individual standard creation) to account for all relative concentrations.

% Way to calculate this:
% C_target [uM] = V_std [mL] * Conc_std [uM] / Vtotal [mL]
% or (inverted)
23 % V_std [mL] = C_target [uM] * Vtotal [mL] / Conc_std [uM]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    verbose = false;
    if (~exist('useOldWay'))
        useOldWay = false; % goes through the true-false logic (not matrix math)
    end % if
    if (useOldWay)

```



```

warning(['If-then logic not implemented for last salt change ' ...
33      '(Mg salt additions) - aborting.']);
return;
end % if

useNaCl = true;

my_ions = {'Na', 'K', 'Ca', 'Mg', 'NH4', 'Cl', 'SO4', 'NO3', ...
          'H+', 'CO3', 'OH'};

43 saltList = {'NaCl', 'KCl', 'CaCl2', 'MgCl2', 'NH4Cl', 'KNO3', ...
            'Na2SO4', 'H2SO4', 'HNO3', 'CaOH2', 'Na2CO3', 'HCl', ...
            'MgSO4', 'MgNO32', 'K2CO3', 'MgCO3'}; % MgSO4, Mg(NO3)2

% std_conc = 100*10^3; % 100mM = 10^5 uM
std_conc = 100*10^3 * ones(length(saltList),1);
std_conc(find(strcmp(saltList, 'CaOH2')==1)) = 20*10^3; % this one has lower
solubility
std_conc(find(strcmp(saltList, 'MgCO3')==1)) = 1.2*10^3; % this one has lower
solubility

conc_vector = [Na K Ca Mg NH4 Cl SO4 NO3]';

53 % check initial charge balance gap
if (0)
    charge_check = Na + K + 2*Ca + 2*Mg + NH4 - Cl - 2*SO4 - NO3;
    display(['CB check - init : ' num2str(charge_check)]);
end % if

% initialize outputs
saltVolumes = zeros(1,length(saltList));
H = NaN;
OH = NaN;
63 CO3 = NaN;

% ACTUAL CALCULATIONS TO INVERT MATRIX:
% SALT_MATRIX * [salt volume list]' * std_conc / Vtotal = [conc_vector]'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Construct PInv by hand for saltMatrix - controls outputs > 0
% (which we could NOT do by using Matlab pinv())
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

73 % NH4Cl
invSaltMatrix(5,:) = [0 0 0 0 1 0 0 0]; % NH4Cl

% MgCl2, MgSO4, Mg(NO3)2
d = (Mg >= SO4);
z = (Mg-SO4 >= NO3/2);

invSaltMatrix(13,:) = [0 0 0 ~d 0 0 d 0]; % MgSO4
invSaltMatrix(14,:) = d*[0 0 0 (~z) 0 0 -(~z) z/2]; % Mg(NO3)2

83 Cl_star = Cl - NH4;
q = (Cl_star / Cl);
t = (Cl_star <= 2*(Mg - ((~d)*Mg+d*SO4) - (d*(~z)*Mg-d*(~z)*SO4+d*z*NO3/2) )); %
not enough Cl to complete Mg
% invSaltMatrix(4,:) = d*z*[0 0 0 1 0 0 -1 -1/2]; % MgCl2
invSaltMatrix(4,:) = d*z*[0 0 0 (~t) 0 (t*q/2) -(~t) -(~t)/2]; % MgCl2
invSaltMatrix(16,:) = t*d*z*[0 0 0 1 0 -q/2 -1 -1/2]; % MgCO3

if (verbose)
    display(['Mg=' num2str(Mg) ', SO4=' num2str(SO4) ', NO3=' num2str(NO3)]);
93    display(['MgSO4: ' num2str((~d)) '*Mg, ' num2str(d) '*SO4']);
    display(['MgNO32: ' num2str(d*(~z)) '*Mg, ' num2str(d*(-~z)) '*SO4, ' ...
            num2str(d*z/2) '*NO3']);
    display(['MgCl2: ' num2str(d*z) '*Mg, ' num2str(-d*z) ...

```

```

        '*SO4, ' num2str(-d*z/2) '*NO3']];
end % if

% Re-prioritize Ca (use of CaCl)
% CaCl2, CaCO3
Cl_star = Cl_star - (2*(d*z*((~t)*(Mg-SO4-NO3/2)+t*q*Cl/2))); % Cl required after
    Mg, NH4 have been completed
103 e = (Cl_star < 2*Ca);
    f = (Cl_star / Cl);
% !!!!! Should not be multiplying by Cl - rather by Cl*
invSaltMatrix(3,:) = [0 0 (~e) 0 0 e*(f/2) 0 0]; % CaCl2
invSaltMatrix(10,:) = e * [0 0 1 0 0 (-f/2) 0 0]; % CaOH2
if (e)
    display(['Using CaOH2 in this solution... ' num2str((Ca-f*Cl/2)*Vtotal/
        std_conc(10))]);
end % if

% KCl, KNO3, HNO3
113 NO3left = NO3*(1 - d*z) - 2*d*(~z)*(Mg-SO4); % NO3-2*((z/2)*d*NO3 + (~z)*d*(Mg-
    SO4))
y = NO3left/NO3;
Cl_star = Cl_star - 2*((~e)*Ca + e*f*Cl/2); % Cl required after Mg, NH4, Ca have
    been completed
c = (K >= (NO3left)); % nitrate minus the amount that we used above for Mg(NO3)2
g = (K-NO3left > Cl_star);
v = Cl_star / Cl;
invSaltMatrix(6,:) = [0 (~c) 0 0 0 0 0 c*y]; % KNO3
invSaltMatrix(9,:) = (~c) * [0 -1 0 0 0 0 0 y]; % HNO3
invSaltMatrix(2,:) = c * [0 (~g) 0 0 0 (g)*v 0 -(~g)*y]; % KCl
invSaltMatrix(15,:) = c * g * 0.5 * [0 1 0 0 0 -v 0 -y]; % K2CO3 instead of KOH
123

% CHECK FOR ERROR
Cl_star = Cl_star - c*((~g)*K-(~g)*y*NO3+g*v*Cl); % constraint based on Cl !
    % Cl_star = Cl required after Mg,
    NH4, K have been completed
if (Cl_star < 0) % Can't make this sample w/ available salts
    warning(['Cannot make this sample with available salts ' ...
        '- error at K allocation: ' num2str(Cl_star)]);
    pause();
    saltVolumes = NaN(size(saltList));
    error = NaN;
133 return;
end % if

% Na2SO4, H2SO4
SO4left = SO4*(1 - d) - (~d)*Mg; % SO4 - (d*SO4+~d*Mg)
w = SO4left/SO4;
a = (Na >= 2*SO4left); % constraint based on SO4
invSaltMatrix(7,:) = [(~a)/2 0 0 0 0 0 a*w 0]; % Na2SO4
invSaltMatrix(8,:) = (~a) * [-1/2 0 0 0 0 0 w 0]; % H2SO4

143 % NaCl, Na2CO3
b = (Cl_star <= (Na-2*SO4left)); % need more Na than Cl - use Na2CO3
invSaltMatrix(1,:) = (useNaCl & ~b) * a * [1 0 0 0 0 0 -2*w 0]; % NaCl
invSaltMatrix(11,:) = (~useNaCl | b) * a * (1/2) * [1 0 0 0 0 0 -2*w 0]; % Na2CO3
% Note: (~useNaCl | b) = ~(useNaCl & ~b), i.e., one choice always true.

% HCl
Cl_star = Cl_star - useNaCl*(~b)*a*(Na-2*w*SO4);
% Cl_star = Cl required after all ions but Cl done
j = (Cl_star / Cl);
153 invSaltMatrix(12,:) = [0 0 0 0 0 j 0 0]; % HCl

% Need to normalize all of them to the correct units [mL]
conc_multiplier = ((std_conc.^-1)*ones(1,size(invSaltMatrix,2)));
invSaltMatrix = (invSaltMatrix .* conc_multiplier) * Vtotal;

```

```

%[NaCl KCl CaCl2 MgCl2 NH4Cl KNO3 Na2SO4 H2SO4 HNO3 CaCO3 Na2CO3 HCl MgSO4 Mg(NO3
) 2]
saltVolumes = invSaltMatrix*conc_vector;

163 error = (saltToConc(saltVolumes, std_conc, Vtotal) - conc_vector)';
H = 2*(~a)*(-Na/2+w*SO4) + (~c)*(y*NO3-K) + j*Cl;
OH = 2*(e)*(Ca-f*Cl/2); % CaOH2
CO3 = t*d*z*(Mg-q*Cl/2-SO4-NO3/2) + (~useNaCl | b)*a*(1/2)*(Na -2*w*SO4) + ...
c*g*(1/2)*(K-v*Cl-y*NO3);

ch_bal_error = (Na + K + 2*Ca + 2*Mg + NH4 + H) - ...
(Cl + 2*SO4 + NO3 + OH + 2*CO3);
if (abs(ch_bal_error) > 10^-12)
warning(['Problem with charge balance! Error = ' num2str(ch_bal_error)]);
173 display(['Errors: ' mat2str(error)]);
end % if

if (verbose)
display(['Errors: ' mat2str(error)]);
pause()
end % if
error = sum(error);

end % function

183

function ind = getInd(ion_str, my_ions)
% Returns the correct position index for this ion_str
ind = find(strcmp(my_ions, ion_str)==1);
end % function

193 function concs = saltToConc(salt_vector, std_conc, Vtotal)
% salt matrix - relates salt constituents to output ions
%
% NaCl KCl CaCl2 MgCl2 NH4Cl KNO3 Na2SO4 H2SO4 HNO3 CaCO3 Na2CO3 HCl MgSO4 Mg
(NO3)2 K2CO3 MgCO3
% Na
% K
% Ca
% Mg
% NH4
% Cl
203 % SO4
% NO3
%
% SALT_MATRIX * [salt volume list]' * std_conc / Vtotal = [conc_vector]'
%

saltMatrix(1,:) = [1 0 0 0 0 0 2 0 0 0 2 0 0 0 0 0]; % Na
saltMatrix(2,:) = [0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 2 0]; % K
saltMatrix(3,:) = [0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0]; % Ca
saltMatrix(4,:) = [0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1]; % Mg
213 saltMatrix(5,:) = [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]; % NH4
saltMatrix(6,:) = [1 1 2 2 1 0 0 0 0 0 0 0 1 0 0 0 0]; % Cl
saltMatrix(7,:) = [0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0]; % SO4
saltMatrix(8,:) = [0 0 0 0 0 1 0 0 1 0 0 0 0 2 0 0 0]; % NO3

% concs = saltMatrix * salt_vector .* std_conc / Vtotal;

multiplier = ones(size(saltMatrix,1),1) *std_conc';
concs = (saltMatrix .* multiplier) * salt_vector / Vtotal;

223 end % function

```

B.3 Creating ANN training set

makeNNsetFINAL.m: Puts together data and does carbonate system/pH/ionic strength calculations to create a self-consistent ANN training set based on environmental mix data.

```
function [] = makeNNsetFINAL(savedir)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loading data from different locations to create the input and
% target sets for ANN training
6 %
% 3/28/12
% Takes into account NH4, SO4 dependencies on pH.
% Extensive testing has produced base calibration for EC meter.
% Extensive looking at data has produced good estimates for all
% optional inputs.
% Must take into account that some ions (NH4, SO4) will NOT be the
% same as the values given in the original recipe file.
%
% inputs: 12 x ISEs (even though CO3 is probably useless)
16 %
%         2 x EC
%         temperature
%
% outputs: 8 x ion concentrations
%          NH3 or NH3+NH4 (optional)
%          HSO4 (optional – but might be needed for EC constraint)
%          H+ ion (optional)
%          OH- ion (optional)
%          pH (optional)
%          EC (optional)
26 %          CB (optional) – need to define what charge balance exactly
%          alkalinity (optional)
%          Ionic Strength (optional)
%          gamma1 (activity coefficient, monovalent ions)
%          gamma2 (activity coefficient, divalent ions)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up directories, filenames for Feb 2012 sampling

th_date = date;
36
% inputs – i.e., measurements from ISE dataset
datadir2 = ['C:\Documents and Settings\Amy\My Documents\MATLAB\code\feb2012\'];
figdirectory = ['C:\Documents and Settings\Amy\My Documents\MATLAB\code\feb2012\figs\
'];
freshdatafile = 'FinalMixData_compacted.mat';
freshdatafile2 = 'FinalMixData_compactedFillIn.mat';
% difference of mean values from the first file listed is 0.0049mV,
% but more have detected equilibrium values!
saltdatafile = 'FinalSalinityData_compacted.mat';

46 % inputs & calculated targets: see checkECpH_mixes2.m
ECdatafile = 'EC_EnvMixes_Mar2012_FINAL.mat';

% outputs – excepting those over-ridden by ECdatafile
env_datapath = ['C:\Documents and Settings\Amy\My Documents\Research\' ...
               'thesis\usgs data – for range detection\'];
MixTruthDatafile = 'NewEngland_TrainingSamples5.mat'; % "5" includes NH4, SO4
                dependency on pH

% pH calibration
PHdir = ['C:\Documents and Settings\Amy\My Documents\Research\' ...
56 'thesis\sampling\Feb 2012 data\pH calibration\'];
PHcalibFile = 'PHcalibs.mat';
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ISE measurements
load([datadir2 freshdatafile]); % m_all, Three_sig_all, num_ss_all
66 InputData = m_all;
InputDataHeaders = {'Cl_E_L_I_T', 'NH4_E_L_I_T', 'Ca_E_L_I_T', 'Na_E_L_I_T', 'K_E_L_I_T',
, 'Na_O_r_i_o_n', 'pH_O_r_i_o_n', ...
'NO3_E_L_I_T', 'Cl_H_a_n_n_a', 'Hardness', 'SO4_F_i_s_h_e_r', '
CO3_T_h_o_m_a_s'};

% for cases where results are NaN, pull data from freshdatafile2
load([datadir2 freshdatafile2], 'm_all'); % m_all, Three_sig_all, num_ss_all
backupInputData = m_all;
nanpoints = find(isnan(InputData));
InputData(nanpoints) = backupInputData(nanpoints);

76 % AVM - 3/11/12
% Need to deal with any NaNs in the inputs!
% still a problem even with time_window = 15sec, smooth_win = 11sec
prob_points = [30 4; ...
37 4; ...
40 11; ...
41 7; ...
63 4; ...
74 7];

86 % this is how to get matlab to identify these points for you
% (columns reversed from prob_points)
[th_i th_j] = ind2sub(size(InputData(1:11,:)), find(isnan(InputData(1:11,:))));

% update these manually - justification in notebook pages 65-66
InputData(4,30) = mean([339.4057 329.9363 327.7211 mean([325.4974 325.4730] )]);
InputData(4,37) = mean([mean([332.0876 331.0720 330.0385]) 336.3367 331.4260]);
InputData(11,40) = mean([-314.45 -311.41 -309.22 -307.17 -305.92 -304.88]);
InputData(7,41) = mean([-20.60 -22.73]);
InputData(4,63) = mean([mean([339.7902 339.7128 339.6405]) mean([334.5473 334.4479])
]);
96 InputData(7,74) = mean([mean([-127.0558 -125.8496 -126.3150]) ...
mean([-119.7950 -119.9260 -120.0497]) ...
-121.1953 ...
mean([-118.2133 -118.2980 -118.3851]) ...
mean([-117.5026 -117.6220 -117.7351]) ]]);

% Note that CO3 signal is pretty much useless, so drop it
InputData = InputData(1:end-1,:);
InputDataHeaders = InputDataHeaders(1:end-1);

106 % EC meters + temperature, also all updated targets based on
% pH/carb system calculations (see above).
load([datadir2 ECdatafile]); % load ALL variables for the moment
% EC meter data is TEMPERATURE AND CALIBRATION CORRECTED
% EC_Amber_m_all, EC_VWR_m_all, Temp_m_all
% also has all of these + Three_sig_all, num_ss_all
% along with expected values for EC,pH under different carbon assumptions
InputDataAncillary = [EC_Amber_m_all' EC_VWR_m_all' Temp_m_all']';
InputDataAncillaryHeaders = {'EC_Amber', 'EC_VWR', 'Temperature'};
% also loads the following targets:
116 % TargetCO3 TargetGamma1 TargetH TargetHSO4 TargetNH4
TargetSO42
% TargetEC TargetGamma2 TargetHCO3 TargetIS TargetOH

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OUTPUTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% outputs should be the actual concentrations of the ions of
% interest, plus others required for EC, charge balance calcs
126 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% start with ion concentrations from "recipes"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load([env_datapath MixTruthDatafile], 'finalSamples', 'th_fieldnames');
TargetFieldnames = th_fieldnames;
TargetData = finalSamples;
clear th_fieldnames finalSamples;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
136 % add carbonate system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TargetData(end+1,:) = TargetHCO3; % from ECfile above
TargetFieldnames(end+1) = {'HCO3'};
TargetData(end+1,:) = TargetCO3; % from ECfile above
TargetFieldnames(end+1) = {'CO3'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% add H+, OH- ions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
146 TargetData(end+1,:) = TargetH; % from ECfile above
TargetFieldnames(end+1) = {'H'};
TargetData(end+1,:) = TargetOH; % from ECfile above
TargetFieldnames(end+1) = {'OH'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% replace appropriate columns for NH4, SO4 - add columns for
% NH4-TOT, HSO4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SO4_index = find(strcmp(TargetFieldnames, 'SO42'));
156 TargetData(end+1,:) = TargetHSO4; % from ECfile above
TargetFieldnames(end+1) = {'HSO4'};

TargetData(end+1,:) = TargetData(SO4_index,:); % TOTAL SO4 (optional)
TargetFieldnames(end+1) = {'SO4_TOT'};
TargetData(SO4_index,:) = TargetSO42; % from ECfile above

NH4_index = find(strcmp(TargetFieldnames, 'NH4'));
TargetData(end+1,:) = TargetData(NH4_index,:); % NH4-TOT = NH4+NH3
TargetFieldnames(end+1) = {'NH4_TOT'};
166 TargetData(NH4_index,:) = TargetNH4; % from ECfile above

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pH to train to - based on pH measurements/calib
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (exist([PHdir PHcalibFile])==2) % check to be sure data is there
    load([PHdir PHcalibFile], 'p'); % p = [slope intercept]
end % for
pHdata = squeeze(InputData(7,:));
% convert pHdata from mV to pH
176 pHdata_asPH = (pHdata - p(2)*ones(size(pHdata)))./(p(1)*ones(size(pHdata)));

TargetData(end+1,:) = pHdata_asPH;
TargetFieldnames(end+1) = {'pH'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EC to train to - based on actual measurements,
% so in theory could be directly constrained by measurements.
% passed in above, called: TargetEC
186 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Also have optional targets for ionic strength and gammas
% all loaded from ECfile above, consistent with pH, carbonate sys, EC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TargetData(end+1,:) = TargetIS; % ionic strength
TargetFieldnames(end+1) = {'IS'};
TargetData(end+1,:) = TargetGamma1; % monovalent activity coeff
196 TargetFieldnames(end+1) = {'g1'};
TargetData(end+1,:) = TargetGamma2; % divalent activity coeff
TargetFieldnames(end+1) = {'g2'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Charge balance —> formulated here as [OH-]-[H+]
% so (in theory) this could be directly constrained by pH
% measurement
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% alkalinity = Na + K + 2*Ca2 + 2*Mg2 + NH4 - Cl - 2*SO42 -NO3
206 %           = OH- + HCO3- + 2*CO32- - H+
% move carbonate terms to left side for CB calc
TargetCB = TargetOH - TargetH;

% save data
if (length(usepoints) < size(TargetData,2))
    % some points were marked to be thrown away — need to take them
    % out of ALL Input, Target data vectors (excepting headers)
    % details in notebook, page 70

216 % keep points listed in variable 'usepoints' (loaded in ECfile)
    keepers = usepoints;

    InputData = InputData(:, keepers);
    InputDataAncillary = InputDataAncillary(:, keepers);

    TargetData = TargetData(:, keepers);
    TargetCB = TargetCB(keepers);
    TargetEC = TargetEC(keepers);

226 end % if

% sensitivity analysis to see how important inclusion of HSO4
% channel really is.
th_index = find(strcmp(TargetFieldnames, 'HSO4'));
TargetHSO4 = TargetData(th_index, :);
TargetHSO4_asEC = (52*1000)*TargetHSO4;
HSO4_ECfrac = TargetHSO4_asEC./TargetEC;
figure; hold on;
236 plot(HSO4_ECfrac, 'b*');
title('Total contribution of HSO4 to EC (fraction)');
if (max(HSO4_ECfrac) < 0.01) % less than 1%
    TargetDataAll = TargetData;
    TargetFieldnamesAll = TargetFieldnames;
    TargetData = TargetData([1:th_index-1 th_index+1:end], :);
    TargetFieldnames = TargetFieldnames([1:th_index-1 th_index+1:end]);
    display('HSO4 does not contribute significantly to EC — removing. ');
end % if

246 save([savedir th_date 'EnvMixesANNdata-FINAL.mat'], 'InputData*', 'Target*');

end % function

```

B.4 ANN with chemical constraints

FullsuiteNN_Wrapper.m: Sets up ANN architecture (constraints, training data, network parameters, etc.), and calls *fullCB_NN.m* for each of the relevant parameter sets.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Based on CB_NN_Wrapper - abstracts away complexities of CB_NN.m %
% Allows you to call it for a wide range of parameters. %
% %
% 3/28/12 - updating for the full ion suite %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [] = FullsuiteNN_Wrapper(options, datadir, figdir)

% pass in all of the options in a single struct - easier for adding new options
12 tic;

% MUST UPDATE! Change these lines if the data moves or changes name
datapath = 'C:\Documents and Settings\Amy\My Documents\MATLAB\code\feb2012\'
saltpluses_filename = ['31-Mar-2012SingleSaltPlusMixesANNdata-FINAL.mat'];
justmixes_filename = '31-Mar-2012EnvMixesANNdata-FINAL.mat';

% option - don't really want this power passed in, but...
% if you are testing and only want to run a single time, set this to 1.
22 if (isfield(options, 'runtestonly'))
    runtest_only = options.runtestonly;
else
    runtest_only = 0;
end % if

% In order to choose exactly what subset of available targets
% are used, go to: NN_TARGET_SELECTION further down in this file

% Identifying label for files created below
32 this_date = date;

%-----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROCESS INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% indicates whether to make figure plots
42 if (isfield(options, 'plotting'))
    plotting = options.plotting;
else
    plotting = 0;
end % if

% indicates whether to save copies of figs
if (isfield(options, 'savefigs'))
    savefigs = options.savefigs;
else
    savefigs = 0;
52 end % if

% indicates whether to suppress figure windows
if (isfield(options, 'nopopups'))
    nopopups = options.nopopups;
else
    nopopups = 1;
62 end % if

% indicates whether to use charge neutrality constraint
if (isfield(options, 'useCBfcn'))
    useCBfcn = options.useCBfcn;
```



```

else
    useCBfcn = 0;
end % if

% indicates whether to use conductivity constraint
if (isfield(options, 'useECfcn'))
    useECfcn = options.useECfcn;
else
72 end % if
    useECfcn = 0;

% indicates whether to use single-salt data as well
if (isfield(options, 'useExtended'))
    useExtended = options.useExtended;
else
    useExtended = 0;
end % if

% specify the output transfer function (may want to use logsig
82 % to force outputs [0,1] - NOTE you should change map_min also then!
if (isfield(options, 'outputTF'))
    outputTF = options.outputTF;
else
    outputTF = 'purelin';
end % if

% indicates minimum value to be used in mapminmax function,
% to specify different values for inputs and targets, use a
% vector with multiple entries: [inputs targets targetCB targetEC]
92 if (isfield(options, 'map_min'))
    map_min = options.map_min;
else
    map_min = -1;
end % if

% indicates which additional outputs to train to
% options are [14:19]
if (isfield(options, 'extrakeepers'))
102 extrakeepers = options.extrakeepers;
    % throws away any values that are out of range
    extrakeepers = extrakeepers(find((extrakeepers >=14) & (extrakeepers <=19)));
else
    extrakeepers = [];
end % if

% indicate whether to train to [] or log10 []
if (isfield(options, 'traintolog'))
112 traintolog = options.traintolog;
else
    traintolog = false;
end % if

% indicate whether to train to [] or log10 [] for EC
if (isfield(options, 'trainEClog'))
    % if we are not training []s to log, do not train EC to log
    trainEClog = options.trainEClog & traintolog;
else
122 trainEClog = false;
end % if

% 3/30/12 - this option is not yet implemented!
trainEClog = false;

% indicate whether to weight EC, CB outputs in error calc
if (isfield(options, 'errorWeightConstraints'))
    % if we are not training []s to log, do not train EC to log

```

```

    errorWeightConstraints = options.errorWeightConstraints;
132 else
    errorWeightConstraints = false;
end % if

%-----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set options
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
142 if (length(map_min)==1)
    map_min_in = map_min;
    map_min_target = map_min;
    map_min_CB = map_min;
    map_min_EC = map_min;
elseif (length(map_min)==2)
    map_min_in = map_min(1);
    map_min_target = map_min(2);
    map_min_CB = map_min(2);
152 map_min_EC = map_min(2);
elseif (length(map_min)==3)
    map_min_in = map_min(1);
    map_min_target = map_min(2);
    map_min_CB = map_min(3);
    map_min_EC = map_min(3);
else
    map_min_in = map_min(1);
    map_min_target = map_min(2);
    map_min_CB = map_min(3);
162 map_min_EC = map_min(4);
end % if

% collect options to pass to sub-routines
suboptions = struct;
suboptions.plotting = plotting;
suboptions.useECfcn = useECfcn;
suboptions.useCBfcn = useCBfcn;
suboptions.map_min_in = map_min_in;
suboptions.map_min_target = map_min_target;
172 suboptions.map_min_CB = map_min_CB;
suboptions.map_min_EC = map_min_EC;
suboptions.outputTF = outputTF;
suboptions.traintolog = traintolog;
suboptions.trainEClog = trainEClog;

if (nopopups)
    set(0,'DefaultFigureVisible','off'); % don't want it to display so many
    figures
end % if

182 if (useExtended)
    datafilename = saltpluses_filename;
    outputfilename = 'SingleSaltsPlusEnvFreshMixes.mat';
else
    datafilename = justmixes_filename;
    outputfilename = 'EnvFreshMixes.mat';
end

%-----%

192 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set ranges for NN parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
maxEpochs = 10000;
tranFunInd = 5; % 'tansig'

```

```

% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! %
% Code only works for up to 3 layers right now! 11/4/11 %
% Note: 3/28/12 - I think it actually generalizes? haven't
% tested it, though.
202 % !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! %

maxLayers = 3;
layerSizes = [6:3:20];
trainGoals = 10.^(-1*[6]); % 10.^(-1*[4:6]);
mu_opts = [0.1 0.001]; % [0.0001 0.001 0.01 0.1 1 10];
mudec_opts = [0.1 0.5 0.9]; % 0.1*[1:9];
muinc_opts = [1.5 10 50]; % [1.5 5 10*[1:5]];

% calculates the number of permutations required to compare
% results with each of the parameter combinations listed
212 num_iters = (sum(length(layerSizes).^[1:maxLayers])) ...
    *length(trainGoals)*length(mu_opts)*length(mudec_opts)*length(muinc_opts);
display(['Setting up run with ' num2str(num_iters) ' iterations...']);

%-----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set constants for NN parameters
222 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% many listed here are set for default, included in case
% changes later are useful.

blf = 'learnngdm'; % default (base learning function)
           % this learning function updates weights as:  $dW = mc*dW_{prev} +$ 
           %  $(1-mc)*lr*gW$ 
lr = 0.01; % learning rate - default = 0.01 for 'learnngdm'
mc = 0.9; % momentum constant - default = 0.9 for 'learnngdm'
bwf = 'dotprod'; % default (base weight function)
232 % mu = 0.01; % default = 0.001    -> Initial mu
% mu_dec = 0.1; % default = 0.1    -> mu decrease factor
% mu_inc = 50; % default = 10     -> mu increase factor

%-----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set up some options - deprecated
242 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTE: This is not being used any more as of 3/28/12
% Previously was used to divide data, ensuring max/min values
% were in the training set...
train_fraction = 0.7; % approx 70% used for training

% Set this flag at the beginning of the flie if you are testing
% & only want to run a single iteration
252 if (runtest_only)
    warning('RUNNING TEST ONLY!!! IS THIS WHAT YOU WANTED?');
    pause(1);
    num_iters = 1;
end % if

%-----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define charge balance output %
262 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% TargetData is now 19 wide:

```

```

% 'NH4'      'Ca2'      'Na'      'K'      'Cl'      'NO3'      'Mg2'      'SO42'
% 'HCO3'     'CO3'     'H'      'OH'     'HSO4'
% 'SO4-TOT'  'NH4-TOT'  'pH'     'IS'     'g1'     'g2'

% charge balance output is constructed as Alk - carb_Alk
% i.e., SUM(plus_ions) - SUM(neg_ions) - HCO3 - 2*CO3 (= OH - H)
% all other rows should be set to zero
272 CBgammas = [+1 +2 +1 +1 -1 -1 +2 -2 ...
              -1 -2 0 0 -1 ...
              0 0 0 0 0 0]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define EC (electrical conductivity) output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% see notebook pg. 65 for references for these numbers
% (contribution of each ion to EC per milli-equivalent per
282 % liter)
% Listed as: (uS/cm) / (meq/L)
ECgammas = [73.5 119 50.10 73.5 76.35 71.46 106 160 ...
            44.5 138.6 349.6 199.1 52 ...
            0 0 0 0 0 0]';
ECgammas = ECgammas*1000; % (uS/cm) / (M)

%-----%
292 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set error weighting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Relatively weight some parameters as more important
% Should be of size NUM.TARGETS x 1 --> recall that we will
% add in optional targets (for charge balance, EC) below.

302 EW = ones(length(CBgammas),1); % size specified by NN functions

% make NH4+, NO3- relatively more important
EW([1 6]) = 10;

% make some other parameters relatively less important
EW([14 15 17 18 19]) = 0.1;
%EW([9 10 11 12 13 14 15 17 18 19]) = 0.1;

%-----%
312 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Setup directories for saving data, figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[figdir datadir] = setupdirs(figdir, datadir);
masteroutputfile = [datadir this_date '-masterfile-' outputfilename];

%-----%
322 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Section Label: NN_TARGET_SELECTION
% Load data from file, choose parameters as inputs, targets
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTE: NNtools expect one set of data per row

332 load([datapath datafilename]);
% InputData TargetData TargetHCO3

```

```

%      InputDataAncillary      TargetEC      TargetHSO4
%      InputDataAncillaryHeaders  TargetFieldnames  TargetIS
%      InputDataHeaders      TargetGamma1  TargetNH4
%      TargetCB      TargetGamma2  TargetOH
%      TargetCO3      TargetH      TargetSO42

% 'logable' marks points that should be log-transformed (if flag is set)
% concentration data, IS, gammas can be log-transformed
logable = ones(1, length(TargetFieldnames));
342 tmp_pHindex = find(strcmp(TargetFieldnames, 'pH')); % do not transform pH
logable(tmp_pHindex) = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUTS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determine use of one or both EC meters' data
InEC = InputDataAncillary(1,:);

352 % Determine yes/no use of temperature data (comment out if 'no'
% and fix next few lines)
InTemp = InputDataAncillary(3,:);

InData = [InTemp' InEC' InputData']'; % each column is one input set
InDataHeaders = {InputDataAncillaryHeaders{[1 3]} InputDataHeaders{:}};
[in_rows, in_cols] = size(InData);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
362 % TARGETS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Choose the subset of target data to train to.
% 1:13 are required for CB constraint & EC constraint
% all others are optional

% check if HSO4 is included.
% if no -> only need to keep 1:13 + extras
th_index = find(strcmp(TargetFieldnames, 'HSO4'));
372 if (isempty(th_index))
    display('Not using HSO4');
    % user has no way to know if HSO4 has been omitted, so they
    % should pass in references to the 19-length.
    extrakeepers = extrakeepers - 1;
    keepers = [1:12 sort(extrakeepers)]; % want these to be in
    % ascending order
    CBgammas = CBgammas([1:12 14:end]);
    ECgammas = ECgammas([1:12 14:end]);
    EW = EW([1:12 14:end]);
382 else
    keepers = [1:13 sort(extrakeepers)]; % want these to be in
    % ascending order
end % if

% Choose the most important of the outputs (to you)
limitedIonSet = [1:8]; % for computing limited MSE, RMSE

% TargetData is now 19 wide:
% 'NH4' 'Ca2' 'Na' 'K' 'Cl' 'NO3' 'Mg2' 'SO42'
392 % 'HCO3' 'CO3' 'H' 'OH' 'HSO4'
% 'SO4-TOT' 'NH4-TOT' 'pH' 'IS' 'g1' 'g2'
%
% UNLESS HSO4 HAS BEEN OMITTED!!

% Make subset selection
TargetData = TargetData(keepers,:);
CBgammas = CBgammas(keepers);
ECgammas = ECgammas(keepers);

```

```

TargetDataHeaders = TargetFieldnames(keepers);
EW = EW(keepers);
402 logable = logable(keepers); % keep track of which can be log-ed

% run this check - should return nothing!
negnums = sum(find(TargetData < 0));
if (negnums > 0)
    warning('Negative numbers in concentration target data! Pausing...');
    pause();
elseif (sum(find(TargetData < 10^-10))>0)
    % for our purposes, these might as well be zero, so reset.
412    % (Should have been done in creation of training sets, but
    % double-check here in case not.)
    probpoints = find(TargetData < 10^-10);
    TargetData(probpoints) = 0;
end % if

[num_tr,num_tc] = size(TargetData); % num_tc should = in_cols

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
422 % Verify data %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (num_tc ~= in_cols)
    warning('Input and Target data have different number of samples.');
```

```

end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
432 % Make targets for CB, EC constraints
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Note that these MUST be their own independent targets to that
% we can selectively 'turn off' training on these weights

% TargetCB - loaded from file above
% TargetEC - loaded from file above

if (trainEClog)
    % EC should never be <= 0, so shouldn't have errors
    TargetEC = log10(TargetEC);
442 end % if

if (useCBfcn)
    if (errorWeightConstraints)
        EW = [EW; 10]; % add error weight for charge balance
    else
        EW = [EW; 1]; % add error weight for charge balance
    end % if
end % if

452 if (useECfcn)
    if (errorWeightConstraints)
        EW = [EW; 10]; % add error weight for conductivity
    else
        EW = [EW; 1]; % add error weight for conductivity
    end % if
end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
462 %-----%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Normalize data for input to NN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% inputs
[NormInData, psi] = mapminmax(InData, map_min_in, 1); % map_min = 0 or -1

```

```

% targets
if (traintolog)
472   % log function will not work on zeros - need to make them
      % something non-zero
      th_zeros = find(TargetData == 0);
      % do not want to overwrite original data
      tmp_TargetData = TargetData;
      tmp_TargetData(th_zeros) = 10^-14; % cannot interfere with other calcs
      tmp_TargetData(th_zeros) = NaN; % cannot interfere with other calcs
      to_log = find(logable==1);
      tmp_TargetData(to_log,:) = log10(tmp_TargetData(to_log,:));
      [NormTargetData, pst] = mapminmax(tmp_TargetData, ...
482                                     map_min_target,1);
else
  [NormTargetData, pst] = mapminmax(TargetData, map_min_target,1);
end % if

% charge balance calc
[NormTargetCB2, ps_cb] = mapminmax(TargetCB, map_min_CB, 1);

% electrical conductivity calc
492 [NormTargetCond, ps_cd] = mapminmax(TargetEC, map_min_EC,1);

% collect together for passing to sub-routines
normfcns = struct;
normfcns.input = psi;
normfcns.target = pst;
normfcns.CB = ps_cb;
normfcns.EC = ps_cd;

%-----%

502 % Rename per convention set up in code commented out above
NNdata = struct;
NNdata.Inputs = NormInData;
NNdata.Targets = NormTargetData;
NNdata.CB = NormTargetCB2;
NNdata.EC = NormTargetCond;

%-----%

512 % loop through all possibilities for parameterization (set
% above)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% create storage variables for all cases
allMSE = NaN(num_iters,1); % mean square error, all targets
allNRMSE = NaN(num_iters,1); % normalized root mean square error
limitedMSE = NaN(num_iters,1); % mean square error, only ion targets
limitedNRMSE = NaN(num_iters,1);

522 % for passing in parameters to sub-routines
% net parameters not dependent on iteration
netparams = struct;
netparams.blf = blf;
netparams.bwf = bwf;
netparams.lr = lr;
netparams.tranFunInd = tranFunInd;
netparams.maxEpochs = maxEpochs;
netparams.mc = mc;

532 for this_iter=1:num_iters

% get parameters for this run
[layers, trainGoal, mu, mu_dec, mu_inc] = ...
  get_params(this_iter, layerSizes, maxLayers, trainGoals, ...

```

```

                                mu_opts , mudec_opts , muinc_opts);
netparams.layers = layers;
netparams.trainGoal = trainGoal;
netparams.mu = mu;
netparams.mu_dec = mu_dec;
542 netparams.mu_inc = mu_inc;

% create the name to identify this run's files
test_id = [this_date '_iterNo_' num2str(this_iter) ...
           '-CB' num2str(useCBfcn) '-EC' num2str(useECfcn) ...
           '-log' num2str(traintolog)];

if (runtest_only)
    % test it's all still working
    layers = [20 17];
552 trainGoal = 10^-6;
    mu = 0.01;
    mu_dec = 0.1;
    mu_inc = 50;
end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now actually call the NN function to create & train the network %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

562 % FIX THIS - WANT TO HAVE ONLY 1 SUBROUTINE CALL

% sub-routine returns the trained network & training record
[net, trRec] = fullCB_NN(netparams, suboptions, NNdata, ...
                        normfns, EW, CBgammas, ECgammas);

% save results to a file for loading later
outputfile = [datadir test_id '-' outputfilename];
save(outputfile, 'net', 'trRec');

572 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluate (single run) results - save to master matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% for ALL data
Y = sim(net, NormInData);
    % recall that this will produce results with one
    % more row than NormTargetData because of the charge
582    % balance calc, & two more rows with conductivity calc

if (useECfcn)
    Ycond = Y(end,:);
    ECResults = mapminmax('reverse', Ycond, normfns.EC);
    Y = Y(1:end-1,:); % peel off for next section of if statements
end % if

if (useCBfcn)
592    Ycb = Y(end,:);
    CBResults = mapminmax('reverse', Ycb, normfns.CB);
    Y = Y(1:end-1,:); % peel off for next section of if statements
end % if

Ytargets = Y;
Results = mapminmax('reverse', Ytargets, normfns.target);

if (traintolog)
    to_invert = find(logable==1);
    % need to transform results back to concentration space
602    Results(to_invert,:) = 10.^Results(to_invert,:);
end % if

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% error calculated in several ways
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ResultsMSE = mse(TargetData - Results);
ResultsNRMSE = sum(sqrt(mean((TargetData'-Results').^2))./mean(TargetData'));

612  if (plotting)
        % show this info in the command window
        ResultsNRMSE2 = sqrt(mean((TargetData'-Results').^2))./ mean(TargetData')
        % shows for individual targets
        ResultsMSE2 = mean((NormTargetData'-Ytargets').^2) % direct ANN output
        MSE
    end % if

%         keyboard

% store MSE / NRMSE results
allMSE(this_iter) = ResultsMSE;
allNRMSE(this_iter) = ResultsNRMSE;
622  ionsTargets = TargetData(limitedIonSet,:); % just the target ions
    ionsResults = Results(limitedIonSet,:);
    limitedMSE(this_iter) = mse(ionsTargets-ionsResults);
    limitedNRMSE(this_iter) = sum(sqrt(mean((ionsTargets'-ionsResults').^2))./
        mean(ionsTargets'));

    if (plotting | savefigs)
        make_nnet_plots(Results, TargetData, TargetDataHeaders, ...
            savefigs, figdir, test_id);
        632  if (useCBfcn)
                % see how far off the charge balance calcs were
                % note CB calcs can be > or < 0 !! -> do not log-transform
                figure; hold on;
                plot(TargetCB,CBResults,'k*');
                plot(TargetCB,TargetCB,'r'); % one-to-one line
                title(['Results for charge balance output']);
                xlabel('Targets [M]');
                ylabel('Results [M]');
                if (savefigs)
                    642  printplot(gcf,figdir,[test_id '-Results-' 'CB' '.jpg']);
                        close(gcf); % do not want to use up memory
                else
                    nicipy_graph(gcf,gca,18);
                end % if
            end % if useCBfcn

            if (useECfcn)
                % see how far off the conductivity calcs were
                % note EC should be > 0 always, log-transform OK
                figure; hold on;
                652  plot(log10(TargetEC), log10(ECResults),'k*');
                    plot(log10(TargetEC), log10(TargetEC),'r');
                    title(['Results for conductivity output']);
                    xlabel('Targets [log10(uS/cm)]');
                    ylabel('Results [log10(uS/cm)]');
                    if (savefigs)
                        printplot(gcf,figdir,[test_id '-Results-' 'EC' '.jpg']);
                        close(gcf);
                    else
                        nicipy_graph(gcf,gca,18);
                    end % if
                end % if useECfcn
            end % if plotting

            if (mod(this_iter,50)==0)
                % save masterfile occasionally in case of program crash
                save(masteroutputfile,'allMSE','allNRMSE','limitedMSE','limitedNRMSE')
            end % if

```

```

        pause(0.1); % will allow us to break program if necessary
    end % for this_iter
672
    runtime = toc;

    % save overall results
    save(masteroutputfile, 'blf', 'bwf', 'lr', 'mc', 'maxEpochs', 'tranFunInd', ...
        'layerSizes', 'maxLayers', 'trainGoals', 'mu_opts', 'mudec_opts', '
        muinc_opts', ...
        'num_iters', 'runtime', ...
        'datapath', 'datafilename', 'outputfilename', 'this_date', ...
        'CBgammas', 'ECgammas', 'TargetDataHeaders', 'keepers', ...
682        'normfcns', 'options', ...
        'allMSE', 'allNRMSE', 'limitedMSE', 'limitedNRMSE');

    % reverse this if we set it at the top
    if (nopopups)
        set(0, 'DefaultFigureVisible', 'on'); % don't want it to display so many
            figures
    end % if

end % function CB_NN_wrapper

```

fullCB_NN.m: Implements creation and training of the constraint neural network architecture.

```

1 % Purpose: This function gets data & parameters in – creates & trains the NN.
%   Incorporates an output that calculates something related to charge
%   balance, as defined by gamma, and uses this as a net constraint.
% INPUTS:
%
% layers : vector of hidden layer sizes, e.g., [20 10 5] — code appends output
%   layer size
% tranFunInd : integer, indicates which transfer function will be used on hidden
%   layers
%   typically tranFunInd=5, i.e., 'tansig' transfer function
%   output layer automatically set as 'purelin'
% maxEpochs : sets the maximum epochs for training
11 % trainGoal : error goal for training
% plotting : boolean, indicates whether figures, informative windows are displayed
% NormInputData : data to be used as input to the nnet, should already be mapminmax
%   processed
% NormTargetData : data to be used as targets for the nnet, should already be
%   mapminmax processed
% pst : function used to do mapminmax processing on TargetData
% gammas : defines function to be used for charge balance calc (elements are +1 for
%   +1 ions, etc.)
% blf: string, default learning function
% bwf: string, default weight function
% lr: double, learning rate
% mc: double, momentum constant
21 % ps_cb : function used to do mapminmax processing on CB loop
% mu : initial mu value (trainParam)
% mu_dec : mu_dec value (trainParam)
% mu_inc : mu_inc value (trainParam)
% EW : error weights, passed in to the train fcn
% useCBfcn : boolean, determines whether CB calculation is built into the NN
% useCondfcn : boolean, determines whether conductivity constraint is built in
% condgammas : defines function to be used for cond calc
% NormTargetCond : data to be used as targets for cond channel, should already be
%   mapminmax processed
% ps_cd : function used to do mapminmax processing on TargetCond data

```

31

```

% 3/29/12
% updating CB_NN, CB_NN2 to create a single cleaner file
% corresponds to updates made in FullsuiteNN_Wrapper.m
%
% function call is dramatically changed
% variable names are NOT dramatically changed
%
% things that have changed:
41 % condgammas —> ECgammas
% gammas —> CBgammas
% useCondFcn —> useECFcn
%

function [cbNet_trained, training_record] = fullCB_NN(netparams, suboptions, ...
    NNdata, normfcns, EW, CBgammas, ECgammas)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% parse parameters from new input format
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51
% network parameters
blf = netparams.blf;
bwf = netparams.bwf;
lr = netparams.lr;
tranFunInd = netparams.tranFunInd;
maxEpochs = netparams.maxEpochs;
mc = netparams.mc;
layers = netparams.layers;
trainGoal = netparams.trainGoal;
61 mu = netparams.mu;
mu_dec = netparams.mu_dec;
mu_inc = netparams.mu_inc;
clear netparams; % save memory

% options
plotting = suboptions.plotting;
useECFcn = suboptions.useECFcn;
useCBFcn = suboptions.useCBFcn;
map_min_in = suboptions.map_min_in;
71 map_min_target = suboptions.map_min_target;
map_min_CB = suboptions.map_min_CB;
map_min_EC = suboptions.map_min_EC;
outputTF = suboptions.outputTF;
traintolog = suboptions.traintolog;
trainEClog = suboptions.trainEClog;
clear suboptions; % save memory

% data normalization functions
81 pst = normfcns.target;
ps_cb = normfcns.CB;
ps_cd = normfcns.EC;
clear normfcns; % save memory

% data
NormInputData = NNdata.Inputs;
NormTargetData = NNdata.Targets;
NormTargetCB = NNdata.CB;
NormTargetCond = NNdata.EC;
91 clear NNdata; % save memory

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Verify inputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (isempty(blf))
    blf = 'learngdm';
end % if
if (isempty(bwf))

```

```

    bwf = 'dotprod';
101 end % if
    if (lr < 10^-6 | lr > 10)
        lr = 0.01;
    end % if
    if (mc < 0 | mc > 10)
        mc = 0.9;
    end % if

% possible transfer functions
tFarray = {'hardlims', 'logsig', 'poslin', 'satlins', 'tansig', 'tribas'};
111 if (tranFunInd < 1 || tranFunInd > 6)
    tranFunInd = 5;
end % if
tF = tFarray{tranFunInd};

if (useECfcn && useCBfcn)
    numlayers = size(layers, 2) + 3; % three output layers, initialized below
elseif (useECfcn || useCBfcn)
    numlayers = size(layers, 2) + 2; % two output layers, initialized below
else % neither extra constraint
121 numlayers = size(layers, 2) + 1; % just one output layer
end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Process input data - append constraint data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (useECfcn && useCBfcn)
    NormTargetData = [NormTargetData ; NormTargetCB ; NormTargetCond];
    % last rows for target Charge Balance (CB) and target Conductivity data
131 elseif (useCBfcn)
    NormTargetData = [NormTargetData ; NormTargetCB]; % last row is for target
    Charge Balance (CB)
elseif (useECfcn)
    NormTargetData = [NormTargetData ; NormTargetCond]; % last row is for target
    EC
end % if

[input_rows, input_cols] = size(NormInputData);
[num_tr, num_tc] = size(NormTargetData); % num_tc should = input_cols

141
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Build the network %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

myNet = network; % creates a new NN (empty)

% input layer
myNet.numInputs = 1; % one input vector of size 'input_rows' (set below)
myNet.numLayers = numlayers;
151 myNet.inputConnect = zeros(numlayers, 1);
myNet.inputConnect(1) = 1; % inputs only connect to the first hidden layer

% for arbitrary number of layers
myNet.layerConnect = zeros(numlayers, numlayers); % first assume no cxn
for tmp = 1:numlayers-1
    myNet.layers{tmp}.netInputFcn = 'netsum';
    myNet.layers{tmp}.initFcn = 'initnw';
    myNet.layerConnect(tmp+1, tmp) = 1; % layerConnect(toLayer, fromLayer)
    % normal feed forward, except last layer
    % only has output cxn
161 myNet.layerWeights{tmp+1, tmp}.weightFcn = bwf;
myNet.layerWeights{tmp+1, tmp}.learnFcn = blf;
% THESE THROW ERRORS AT THE MOMENT BUT I DO NOT KNOW WHY
%myNet.layerWeights{tmp+1, tmp}.learnParam.lr = lr;

```

```

    %myNet.layerWeights{tmp+1, tmp}.learnParam.mc = mc;
    myNet.biasConnect(tmp) = 1;
    myNet.biases{tmp}.learnFcn = blf;
    myNet.biases{tmp}.learnParam.lr = lr;
    myNet.biases{tmp}.learnParam.mc = mc;
end % for
171
% last layer
myNet.layers{numlayers}.netInputFcn = 'netsum';
myNet.layers{numlayers}.initFcn = 'initnw';
myNet.biasConnect(numlayers) = 1;
myNet.biases{numlayers}.learnFcn = blf;
myNet.biases{numlayers}.learnParam.lr = lr;
myNet.biases{numlayers}.learnParam.mc = mc;

if (useECfcn && useCBfcn)
181    % want to connect these both back to CONCENTRATION
    % pre-outputs, not CB -> EC
    myNet.layerConnect(numlayers, numlayers-2) = 1;
    myNet.layerConnect(numlayers, numlayers-1) = 0;
end % if

% output layer
myNet.outputConnect = zeros(numlayers,1)'; % first assume no cns
if (useECfcn && useCBfcn)
191    myNet.outputConnect(end-2) = 1; % ion concentration outputs
    myNet.outputConnect(end-1) = 1; % single output, calculates charge balance
    myNet.outputConnect(end) = 1; % single output, calculates conductivity
elseif (useCBfcn || useECfcn)
    myNet.outputConnect(end-1) = 1; % ion concentration outputs
    myNet.outputConnect(end) = 1; % single output, calculates CB or EC
else
    myNet.outputConnect(end) = 1; % ion concentration outputs
end % if

201 % {
% BETTER WAY TO WRITE THIS CODE? more efficient, less informative
myNet.outputConnect(end-2) = useECfcn && useCBfcn;
myNet.outputConnect(end-1) = useECfcn || useCBfcn;
myNet.outputConnect(end) = 1;
% }

% initialize the inputs
myNet.inputs{1}.size = input_rows; % should be 11 (up to 14)
myNet.inputWeights{1}.weightFcn = bwf;
211 myNet.inputWeights{1}.learnFcn = blf;
myNet.inputWeights{1}.initFcn = 'initnw';
myNet.inputs{1}.range = ones(myNet.inputs{1}.size,1)*[map_min_in 1];
% note: this m-function takes in manually mapminmax-ed input data

% initialize all of the hidden (non-output connected) layers
for tmp=1:numlayers-(sum(myNet.outputConnect))
    % sum(myNet.outputConnect) = 2 for CB, 3 for cond as well
    myNet.layers{tmp}.size = layers(tmp);
    myNet.layers{tmp}.transferFcn = tF;
221 end % for

% set up output layers and layer weights
% update weight functions
if (useECfcn && useCBfcn)
    % 3rd-to-last layer is ion concentrations (plus other
    % outputs we want) - has linear transformation to output
    % (or logsig coupled to [0 1] mapminmax to avoid <0 outputs)
    myNet.layers{numlayers-2}.size = num_tr-2; % anywhere from 13 up to 19
    % transferFcn: specified how to calculate this layer's
231 % output from its inputs
    myNet.layers{numlayers-2}.transferFcn = outputTF; % calcs signals to outputs

```

```

% 2nd-to-last layer is just for doing charge balance calculation (also goes
to output)
myNet.layers{numlayers-1}.size = 1;
% CB output can be positive or negative...
myNet.layers{numlayers-1}.transferFcn = 'purelin'; % outputTF; % going to
output

% last layer is just for doing conductivity calculation (also goes to output)
myNet.layers{numlayers}.size = 1;
241 % output must be > 0
% myNet.layers{numlayers}.transferFcn = 'logsig'; % going to outputs
myNet.layers{numlayers}.transferFcn = 'purelin'; % going to outputs

% set the cb loop based on a given function (the charge balance calculation)
% Don't want those settings to update!

% Recall:
% net.LW{toLayer,fromLayer} : layer weight
% size(net.LW{toLayer,fromLayer}) = (#rows, #cols = size of fromLayer)
251

% feedback from concentrations to charge balance
if (traintolog)
    myNet.layerWeights{numlayers-1,numlayers-2}.weightFcn = 'cbsum';
end % if
myNet.layerWeights{numlayers-1,numlayers-2}.learn = false; % don't want
feedback values to change
myNet.biases{numlayers-1}.learn = false;

% feedback from concentrations to conductivity
if (traintolog)
261 myNet.layerWeights{numlayers,numlayers-2}.weightFcn = 'cbsum';
end % if
myNet.layerWeights{numlayers,numlayers-2}.learn = false; % don't want
feedback values to change
myNet.biases{numlayers}.learn = false;
elseif (useECfcn) % uses EC function but not CB function
% 2nd-to-last layer is ion concentrations (plus other
% outputs we want) - has linear transformation to output
% (or logsig coupled to [0 1] mapminmax to avoid <0 outputs)
myNet.layers{numlayers-1}.size = num_tr-1;
myNet.layers{numlayers-1}.transferFcn = outputTF; % going to outputs, notes
above
271

% last layer is just for doing conductivity calculation (also goes to output)
myNet.layers{numlayers}.size = 1;
% output must be > 0
% myNet.layers{numlayers}.transferFcn = 'logsig'; % going to outputs
myNet.layers{numlayers}.transferFcn = 'purelin'; % going to outputs

% set the EC loop based on a given function -
% Don't want those settings to update!
if (traintolog)
281 myNet.layerWeights{numlayers,numlayers-1}.weightFcn = 'cbsum';
end % if
myNet.layerWeights{numlayers,numlayers-1}.learn = false; % don't want the
feedback values to change
myNet.biases{numlayers}.learn = false;
elseif (useCBfcn)
% 2nd-to-last layer is ion concentrations (plus other
% outputs we want) - has linear transformation to output
% (or logsig coupled to [0 1] mapminmax to avoid <0 outputs)
myNet.layers{numlayers-1}.size = num_tr-1;
myNet.layers{numlayers-1}.transferFcn = outputTF; % going to outputs, notes
above
291

% last layer is just for doing charge balance calculation (also goes to
output)

```

```

myNet.layers{numlayers}.size = 1;
% CB output can be positive or negative...
myNet.layers{numlayers}.transferFcn = 'purelin'; % going to outputs

% set the cb loop based on a given function (simulating the charge balance
% calculation)
% Don't want those settings to update!
if (traintolog)
    myNet.layerWeights{numlayers, numlayers-1}.weightFcn = 'cbsum';
301 end % if
myNet.layerWeights{numlayers, numlayers-1}.learn = false; % don't want the
% feedback values to change
myNet.biases{numlayers}.learn = false;
else
% last layer is ion concentrations (plus other
% outputs we want) - has linear transformation to output
% (or logsig coupled to [0 1] mapminmax to avoid <0 outputs)
myNet.layers{numlayers}.size = num_tr;
myNet.layers{numlayers}.transferFcn = outputTF;
311 end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize network weights
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

myNet.initFcn = 'initlay';
myNet = init(myNet); % initialize the network with random numbers on weights &
% biases

321 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% set up static weights for constraints
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (useECfcn && useCBfcn)
    EClayer = numlayers;
    CBlayer = numlayers-1;
    concLayer = numlayers-2;
elseif (useECfcn)
331 EClayer = numlayers;
    CBlayer = NaN;
    concLayer = numlayers-1;
elseif (useCBfcn)
    EClayer = NaN;
    CBlayer = numlayers;
    concLayer = numlayers-1;
else % use neither
    EClayer = NaN;
    CBlayer = NaN;
    %concLayer = numlayers; % not needed
341 end % if

if (~isnan(EClayer))
    myNet = setupEClayer(myNet, EClayer, concLayer, pst, ps_cd, ...
                        ECgammas, traintolog, trainEClog);
end % if
if (~isnan(CBlayer))
    myNet = setupCBlayer(myNet, CBlayer, concLayer, pst, ps_cb, ...
                        CBgammas, traintolog);
end % if

351 % set other network parameters
myNet.performFcn = 'mse';
myNet.trainFcn = 'trainlm'; % 'traingdm' 'traingda' — mem
% problems, then 'trainscg' or 'trainrp'
myNet.adaptFcn = 'adaptwb';

```

```

myNet.divideFcn = 'dividerand'; % default
myNet.divideParam.trainRatio = 0.7;
myNet.divideParam.valRatio = 0.15;
361 myNet.divideParam.testRatio = 0.15;

myNet.trainParam.epochs = maxEpochs; % default = 100
myNet.trainParam.goal = trainGoal; % default = 0
myNet.trainParam.max_fail = 6; % default = 5 —> Maximum validation failures
%myNet.trainParam.min_grad = ???; % default = 1e-10 —> Minimum performance
    gradient
myNet.trainParam.mu = mu; % default = 0.001 —> Initial mu
myNet.trainParam.mu_dec = mu_dec; % default = 0.1 —> mu decrease factor
myNet.trainParam.mu_inc = mu_inc; % default = 10 —> mu increase factor
%myNet.trainParam.mu_max = ???; % default = 1e10 —> Maximum mu
371 myNet.trainParam.show = 50; % default = 25
myNet.trainParam.showCommandLine = 0; % —> Generate command-line output
myNet.trainParam.time = 60*60; % default = inf —> Maximum time to train in
    seconds

if (plotting)
    myNet.trainParam.showWindow = 1; % —> Show training GUI
    myNet.plotFcns = {'plotperform','plottrainstate','plotregression'};
    view(myNet)
else
    myNet.trainParam.showWindow = 0; % —> Do not show training GUI
381 end % if

% reformat EW vector to the shape expected by this fcn
if (useECfcn && useCBfcn)
    EW = {EW(1:end-2)*ones(1,num_tc) ; EW(end-1)*ones(1, num_tc) ; EW(end)*ones
        (1, num_tc)};
    % charge balance & conductivity have separate EW columns
elseif (useCBfcn || useECfcn)
    EW = {EW(1:end-1)*ones(1,num_tc) ; EW(end)*ones(1, num_tc)};
    % add on the error weighting for the single CB/EC channel
else
391 EW = {EW(1:end)*ones(1,num_tc)};
end % if

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Train the nnet %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[cbNet_trained, training_record] = train(myNet, NormInputData, NormTargetData, {},
    {}, EW);
%[cbNet_trained, training_record] = train(myNet, NormInputData, NormTargetData); %
    no error weighting
401 end % function CB_NN2

% FIX THIS FUNCTION
function myNet = setupEClayer(myNet, EClayer, concLayer, pst, ps_cd, ...
    ECgammas, traintolog, trainEClog)
% given NN & information, set up constraint for electrical conductivity

% Must calculate final EC weights based on the transformation (mapminmax)
411 % on the entire target data set ("pst" - input param)

if (traintolog)
    % this function depends on whether EC target is log-ed
    if (trainEClog)
        % concentration as log[] and EC as log[]

        % FIX ME FIX ME FIX ME
        % 3/30/12 - not yet implemented!
        warning('Training to log(EC) not yet implemented.');
```



```

421         return

    else
        % concentrations as log[], but EC as regular
        % This looks identical to CB case (next function below).
        % See pages 72–74 in notebook for derivation of these functions.

        % update weighting function for input to EC layer
        %         myNet.layerWeights{EClayer, concLayer}.weightFcn = 'cbsum';

431        % take into account normalization of the target data
        w1s = (pst.xrange)./pst.yrange; % weights w1 in exponent
        beta1 = ((pst.ymin).*(pst.xrange))./(pst.yrange) - pst.xmin; % constant
            multiplier in  $w2*P*10^{const}$ 
        beta1 = 10.^(-1*beta1);
        w2s = ECgammas.*beta1;

        % take into account normalization of the EC data
        w2s = (w2s*ps_cd.yrange)/ps_cd.xrange;
        cond_constant_part = (-1*ps_cd.xmin)*ps_cd.yrange/ps_cd.xrange + ps_cd.
            ymin;

441        % 'cbsum' weight function requires twice as many weight params
        condW = [w1s' w2s']';
    end % if
else
    % take into account normalization of the target data
    alpha = (ECgammas.*pst.xrange)./pst.yrange;
    beta = sum(ECgammas.*pst.xmin - (ECgammas.*pst.ymin).*(pst.xrange)./(pst.
        yrange));

    % take into account normalization of the cond data
    condW = (alpha*ps_cd.yrange)/ps_cd.xrange;
451    cond_constant_part = (beta - ps_cd.xmin)*ps_cd.yrange/ps_cd.xrange + ps_cd.
        ymin;
end % if

% seems like there is no way to disable initialization on all of the weights,
% so reset the EC calc ones afterward (and it has learning set to false so
% shouldn't change)
myNet.LW{EClayer, concLayer} = condW'; % from []s to cond
myNet.b{EClayer} = [cond_constant_part]; % @ cond layer

end % function

461 function myNet = setupCBlayer(myNet, CBlayer, concLayer, pst, ps_cb, ...
    CBgammas, traintolog)
% given NN & information, set up constraint for charge balance

% Must calculate charge balance weights based on the transformation (mapminmax)
% on the entire target data set ("pst" - input param)
%
% If training to log concentrations, need to calculate weights differently.
%
%{
471 % These calculations do not take into account the normalization on the CB data
    itself
    cbW = (CBgammas.*pst.xrange)./pst.yrange;
    cb_constant_part = sum(CBgammas.*pst.xmin - (CBgammas.*pst.ymin).*(pst.xrange)./(
        pst.yrange));
%}

if (traintolog)
    % see pages 72–74 in notebook for derivation of these functions

    % update weighting function for input to CB layer
    %         myNet.layerWeights{CBlayer, concLayer}.weightFcn = 'cbsum';
481

```

```

    % take into account normalization of the target data
    w1s = (pst.xrangle)./pst.yrange; % weights w1 in exponent
    beta1 = ((pst.ymin).*(pst.xrangle))./(pst.yrange) - pst.xmin; % constant
        multiplier in  $w2 * P * 10^{const}$ 
    beta1 = 10.^(-1*beta1);
    w2s = CBgammas.*beta1;

    % take into account normalization of the CB data
    w2s = (w2s*ps_cb.yrange)/ps_cb.xrangle;
    cb_constant_part = (-1*ps_cb.xmin)*ps_cb.yrange/ps_cb.xrangle + ps_cb.ymin;
491
    % 'cbsum' weight function requires twice as many weight params
    cbW = [w1s' w2s']';
else
    % take into account normalization of the target data
    alpha = (CBgammas.*pst.xrangle)./pst.yrange;
    beta = sum(CBgammas.*pst.xmin - (CBgammas.*pst.ymin).*(pst.xrangle)./(pst.
        yrange));

    % take into account normalization of the CB data
    cbW = (alpha*ps_cb.yrange)/ps_cb.xrangle;
501    cb_constant_part = (beta - ps_cb.xmin)*ps_cb.yrange/ps_cb.xrangle + ps_cb.ymin
        ;
end % if

% seems like there is no way to disable initialization on all of the weights,
% so reset the charge balance calc ones afterward (and it has learning set to
    false so shouldn't change)
myNet.LW{CBlayer, concLayer} = cbW';
myNet.b{CBlayer} = [cb_constant_part];

end % function

```

cbcum.m: WeightFcn for implementing constraints on log-transformed data.

```

1 function out1 = cbsum(varargin)
%CBSUM Custom weight function for doing charge balance constraint
%
% Weight functions apply weights to an input to get weighted inputs.
%
% Returns the dot product  $W2 * 10^{(W1 * P)}$  of a weight matrix W and
% an input P.
%
%
% Amy Mueller, 3-23-12
11 % adapted from dotprod(varargin) (MathWorks)

% NOTE: BOILERPLATE SECTION OMITTED, SEE dotprod.m

%% =====
% BOILERPLATE_START
% This code is the same for all Weight Functions.

% BOILERPLATE_END
%% =====
21
function name = function_name, name = 'CB sum'; end
function flag = is_continuous, flag = true; end
function d = p_deriv, d = 0; end
function d = w_deriv, d = 0; end
function param = parameters, param = []; end

function err = check_param(param)
    err = '';
end

```

```

31 function dim = weight_size(s,r,param)
    dim = [s 2*r]; % first r are exponent
                % second r are multiplier
end

function z = apply(w,p,param)
    r = size(p,1);
    s = size(p,2); % gets called with ALL samples simultaneously
    w1 = w(1:r);
41    w2 = w(r+1:end);
    w1s = w1'*ones(1,s); % same size as p
    z = w2*(10.^(w1s.*p));
end

function d = dz_dp(w,p,z,param)
    r = size(p,1);
    s = size(p,2);
    w1 = w(1:r);
    w2 = w(r+1:end);
51    w1s = w1'*ones(1,s); % same size as p
    d = w2 * ((10.^(w1s.*p)) * w1s') * log(10);
    % expect size to be numNodes (1) x size(p,1) (10)
    % (dz_dp)*p -> (1 x 10) x (10 x N) -> 1 X N
end

function d = dz_dw(w,p,z,param)
    r = size(p,1);
    s = size(p,2);
    w1 = w(1:r);
61    w2 = w(r+1:end);
    w1s = w1'*ones(1,s); % same size as p
    for i=1:s
        d1(:,i) = (w2'.*p(:,i)) .* (10.^(w1'.*p(:,i))) * log(10);
    end % for
    d2 = (10.^(w1s.*p));
    d = [d1' d2']';
    % requires output of size [length(w) x size(p,2)]
    % w*(dz_dw) -> (1 x 20) x (20 x N) -> 1 x N
end
71 function p = simulink_params(param)
    p = {};
end

```

Bibliography

- [1] Bakker, E. and Pretsch, E. (2002) *Analytical Chemistry* **74(15)**, 420A–426A.
- [2] Wanichacheva, N. Design and Synthesis of Ionophores and Fluoroionophores for the Detection of Lithium and Ammoniums ions PhD thesis Worcester Polytechnic Institute Worcester, Massachusetts (2006).
- [3] Stauthamer, W., Engbersen, J., Verboom, W., and Reinhoudt, D. (1994) *Sensors and Actuators B* **17**, 197–201.
- [4] Lindner, E., Toth, K., and Pungor, E. (1988) *Dynamic Characteristics of Ion-Selective Electrodes*, CRC Press, .
- [5] Beale, M., Hagan, M., and Demuth, H. *Neural Network Toolbox User's Guide*, 7 The Mathworks, Inc. Natick, MA (2010).
- [6] Atkins, P. and de Paula, J. (2003) *Physical Chemistry*, Oxford Univ. Press, Oxford, U.K. 7th edition.
- [7] Vanysek, P. *CRC Handbook of Chemistry and Physics* chapter Ionic conductivity and diffusion at infinite dilution CRC Press Boca Raton, FL 81st edition (2000).
- [8] Survey, U. S. G. *Usgs water quality samples for usa: Sample data February 2010*.
- [9] Vlasov, Y. and Legin, A. (1998) *Journal of Analytical Chemistry* **361**, 255–260.
- [10] Morf, W., Seiler, K., Lehmann, B., Behringer, C., Hartman, K., and Simon, W. (1989) *Pure Applied Chemistry* **61(9)**, 1613–1618.
- [11] Buhlmann, P., Pretsch, E., and Bakker, E. (1998) *Chemical Reviews* **98**, 1593–1687.
- [12] Grygolicz-Pawlak, E. B. G. C. E., Mistlberger, G., Pawlak, M., and Xie, X. (2011) *Chimia* **65(3)**, 141–149.
- [13] Bakker, E. and Pretsch, E. (2005) *Trends in Analytical Chemistry* **24(3)**, 199–207.
- [14] Puntener, M., Vigazzy, T., Baier, E., Ceresa, A., and Pretsch, E. (2004) *Analytica Chimica Acta* **503**, 187–194.
- [15] Bobacka, J. (2006) *Electroanalysis* **18**, 7–18.
- [16] Michalska, A. (2006) *Analytical and Bioanalytical Chemistry* **384**, 391–406.
- [17] Wilson, D., Hoyt, S., Janata, J., Booksh, K., and Obando, L. (2001) *IEEE Sensors Journal* **1(4)**, 256–274.

- [18] Bobacka, J., Ivaska, A., and Lewenstam, A. (2008) *Chemical Reviews* **108**, 329–351.
- [19] Covington, A. and Sibbald, A. (1987) *Phil. Trans. R. Soc. Lond. B* **316**, 31–46.
- [20] Bakker, E. (2004) *Analytical Chemistry* **76**, 3285–3298.
- [21] Johnson, R. and Bachas, L. (2003) *Anal Bioanal Chem* **376**, 328–341.
- [22] Bott, A. and Heineman, W. (2004) *Current Separations* **20(4)**, 121–126.
- [23] Vlasov, Y., Legin, A., Rudnitskaya, A., DiNatale, C., and D’Amico, A. (2005) *Pure Applied Chemistry* **77(11)**, 1965–1983.
- [24] IUPAC (1995) *Pure Applied Chemistry* **67(3)**, 510–518.
- [25] IUPAC (1994) *Pure Applied Chemistry* **66(12)**, 2527–2536.
- [26] Hopke, P. (2003) *Analytica Chimica Acta* **500**, 365–377.
- [27] Jurs, P., Bakken, G., and McClelland, H. (2000) *Chemical Reviews* **100**, 2649–2678.
- [28] Mas, J. and Flores, J. (2008) *International Journal of Remote Sensing* **29(3)**, 617–663.
- [29] Despagne, F. and Massart, D. (1998) *Analyst* **123**, 157–178.
- [30] Foresee, F. and Hagan, M. (1997) In Proceedings of the 1997 International Joint Conference on Neural Networks : pp. 1930–1935.
- [31] Wold, S., Esbensen, K., and Geladi, P. (1987) *Chemometrics and Intelligent Laboratory Systems* **2**, 37–52.
- [32] Badertscher, M. and Pretsch, E. (2006) *Trends in Analytical Chemistry* **25(11)**, 1131–1138.
- [33] Legin, A., Rudnitskaya, A., and Vlasov, Y. Comprehensive Analytical Chemistry, Volume 39: Integrated Analytical Systems volume **39**, chapter 10 – Electronic tongues: new analytical perspective for chemical sensors Elsevier (2003).
- [34] Gutes, A., Cespedes, F., and del Valle, M. (2007) *Analytica Chimica Acta* **600**, 90–96.
- [35] Ciosek, P. and Wroblewski, W. (2007) *The Analyst* **132**, 963–978.
- [36] A. Riul, J., Dantas, C., Miyazakic, C., and O.N. Oliveira, J. (2010) *Analyst* **10**, 2481–2495.
- [37] del Valle, M. (2010) *Electroanalysis* **10**, 1539–1555.
- [38] Bratov, A., Abramova, N., and Ipatov, A. (2010) *Analytica Chimica Acta* **678**, 149–159.
- [39] Mimendia, A., Gutierrez, J., Leija, L., Hernandez, P., Favari, L., Munoz, R., and del Valle, M. (2010) *Environmental Modeling and Software* **25**, 1023–1030.
- [40] Prien, R. (2007) *Marine Chemistry* **107**, 422–432.

- [41] Hanrahan, G., Patil, D., and Wang, J. (2004) *Journal of Environmental Monitoring* **6**, 657–664.
- [42] Daly, K., Byrne, R., Dickson, A., Gallagher, S., Perry, M., and Tivey, M. (2004) *Marine Technology Society Journal* **38(2)**, 121–143.
- [43] Taillefert, M., Luther, III, G., and Nuzzio, D. (2000) *Electroanalysis* **12(6)**, 401–412.
- [44] Dybko, A. (2001) *Sensors* **1**, 29–37.
- [45] Brett, C. (2001) *Pure Applied Chemistry* **73(12)**, 1969–1977.
- [46] Qin, Y., Peper, S., Radu, A., Ceresa, A., and Bakker, E. (2003) *Analytical Chemistry* **75**, 3038–3045.
- [47] Antonisse, M., Snellink-Ruel, B., Engbersen, J., and Reinhoudt, D. (1998) *Sensors and Actuators B* **47**, 9–12.
- [48] Glud, R., Jensen, K., and Revsbech, N. (1995) *Geochimica et Cosmochimica Acta* **59(2)**, 231–237.
- [49] Luther, III, G., Giblin, A., and Varsolona, R. (1985) *Limnology and Oceanography* **30(4)**, 727–736.
- [50] Luther, III, G., Reimers, C., Nuzzio, D., and Lovalvo, D. (1999) *Environmental Science and Technology* **33**, 4352–4356.
- [51] Johnson, K., Coletti, L., and Chavez, F. (2006) *Deep-Sea Research I* **53**, 561–573.
- [52] Johnson, K. and Coletti, L. (2002) *Deep-Sea Research I* **49**, 1291–1305.
- [53] Sohn, Y., Goodey, A., Anslyn, E., McDevitt, J., Shear, J., and Neikirk, D. (2005) *Biosensors and Bioelectronics* **21**, 303–312.
- [54] Xu, C., Qin, Y., and Bakker, E. (2004) *Talanta* **63**, 180–184.
- [55] Boo, Y. C., Tressel, S., and Jo, H. (2007) *Nitric Oxide* **16**, 306–312.
- [56] Huber, C., Klimant, I., Krause, C., Werner, T., and Wolfbeis, O. (2001) *Analytica Chimica Acta* **449**, 81–93.
- [57] Tengberg, A., Hovdenes, J., Andersson, J., Brocandel, O., Diaz, R., Hebert, D., Arnerich, T., Huber, C., Krtzinger, A., Khripounoff, A., Rey, F., Rnning, C., Schimanski, J., Sommer, S., and Stangelmayer, A. (2006) *Limnology and Oceanography Methods* **4**, 7–17.
- [58] Yamakawa, S. and Yamaguchi, A. (1995) *Sensors and Materials* **7(4)**, 271–280.
- [59] Puyol, M., del Valle, M., Garces, I., Villuendas, F., Dominguez, C., and Alonso, J. (1999) *Analytical Chemistry* **71**, 5037–5044.
- [60] Rosenzweig, Z. and Kopelman, R. (1995) *Analytical Chemistry* **67**, 2650–2654.
- [61] Spichiger-Keller, U. (1997) *Sensors and Actuators B* **38–39**, 68–77.

- [62] McGraw, C., Radu, T., Radu, A., and Diamond, D. (2008) *Electroanalysis* **20(3)**, 340–346.
- [63] Sokalski, T., Ceresa, A., Zwickl, T., and Pretsch, E. (1997) *Journal of the American Chemical Society* **119**, 11346–11348.
- [64] Chumbimuni-Torres, K., Thammakhet, C., Galik, M., Calvo-Marzal, P., Wu, J., Bakker, E., Flechsig, G., and Wang, J. (2009) *Anal. Chem.* **81**, 10290–10294.
- [65] Morf, W. (1981) *The Principles of Ion-Selective Electrodes and of Membrane Transport*, Elsevier Scientific Publishing Company and Akademiai Kiado, The Publishing House of the Hungarian Academy of Sciences, The Netherlands and Budapest, Hungary.
- [66] Sokalski, T., Lingenfelter, P., and Lewenstam, A. (2003) *J. Phys. Chem.* **107**, 2443–2452.
- [67] Bakker, E., Meruva, R., Pretsch, E., and Meyerhoff, M. (1994) *Anal. Chem.* **66**, 3021–3030.
- [68] Sokalski, T. and Lewenstam, A. (2001) *Electrochemistry Communications* **3**, 107–112.
- [69] NICO2000 How ion-selective electrodes work June 2010.
- [70] Wroblewski, W. Ion-selective electrodes June 2010.
- [71] Sudholter, E., van der Wal, P., Skowronska-Ptasinska, M., van den Berg, A., Bergveld, P., and Reinhoudt, D. (1990) *Analytica Chimica Acta* **230**, 59–65.
- [72] Zhang, J., Harris, A., Cattrall, R., and Bond, A. (2010) *Anal. Chem.* **82**, 1624–1633.
- [73] McNaught, A. and Wilkinson, A. (1997) *IUPAC Compendium of Chemical Terminology*, Blackwell Science, .
- [74] Holmin, S., Spangeus, P., Krantz-Rlcker, C., and Winquist, F. (2001) *Sensors and Actuators B* **76**, 455–464.
- [75] Winquist, F., Holmin, S., Krantz-Rlcker, C., Wideb, P., and Lundstrm, I. (2000) *Analytica Chimica Acta* **406**, 147–157.
- [76] Luther, III, G., Brendel, P., Lewis, B., Sundby, B., Lefrancois, L., Silverberg, N., and Nuzzio, D. (1998) *Limnology and Oceanography* **43(2)**, 325–333.
- [77] Ma, S., Luther, III, G., Scarborough, R., and Mensinger, M. (2007) *Electroanalysis* **19–20**, 2051–2058.
- [78] Rozan, T., Theberge, S., and Luther, III, G. (2000) *Analytica Chimica Acta* **415**, 175–184.
- [79] Zen, J., Kumar, A., and Wang, H. (2000) *Analyst* **125**, 2169–2172.
- [80] Winquist, F., Rydberg, E., Holmin, S., Krantz-Rlcker, C., and Lundstrm, I. (2002) *Analytica Chimica Acta* **471**, 159–172.

- [81] Moreno-Baron, L., Cartas, R., Merkoci, A., Alegret, S., del Valle, M., Leija, L., Hernandez, P., and Munoz, R. (2006) *Sensors and Actuators B* **113**, 487–499.
- [82] Gutes, A., Cespedes, F., Alegret, S., and del Valle, M. (2005) *Talanta* **66**, 1187–1196.
- [83] Krantz-Rlecker, C., Stenberg, M., Winqvist, F., and Lundstrm, I. (2001) *Analytica Chimica Acta* **426**, 217–226.
- [84] Ramsing, N., Kuhl, M., and Jorgensen, B. (1993) *Applied and Environmental Microbiology* **59(11)**, 23840–3849.
- [85] Revsbech, N. and Jorgensen, B. (1986) *Advances in Microbial Ecology* **9**, 293–352.
- [86] Revsbech, N., Jorgensen, B., Blackburn, H., and Cohen, Y. (1983) *Limnology and Oceanography* **28(6)**, 1062–1074.
- [87] Charef, A., Ghauch, A., Baussand, P., and Martin-Bouyer, M. (2000) *Measurement* **28**, 219–224.
- [88] Epstein, J. and Walt, D. (2003) *Chemical Society Reviews* **32**, 203–214.
- [89] Wolfbeis, O. (2004) *Analytical Chemistry* **76**, 3269—3284.
- [90] Bakker, E., Buhlmann, P., and Pretsch, E. (1997) *Chemical Reviews* **97(8)**, 3083–3132.
- [91] Goff, T. L., Braven, J., Ebdon, L., and Scholefield, D. (2003) *Journal of Environmental Monitoring* **5**, 353–358.
- [92] Winkler, S., Rieger, L., Saracevic, E., Pressi, A., and Gruber, G. (2004) *Water Science and Technology* **50(11)**, 105–114.
- [93] Muller, B., Buis, K., Stierli, R., and Wehrli, B. (1998) *Limnology and Oceanography* **43(7)**, 1728–1733.
- [94] Glazer, B., Marsh, A., Stierhoff, K., and Luther, III, G. (2004) *Analytica Chimica Acta* **518**, 93–100.
- [95] Hartnett, M. and Diamond, D. (1997) *Analytical Chemistry* **69**, 1909–1918.
- [96] Berg, P., Roy, H., Janssen, F., Meyer, V., Jorgensen, B., Huettel, M., and de Beer, D. (2003) *Marine Ecology Progress Series* **261**, 75–83.
- [97] Lassen, C. and Jorgensen, B. (1994) *FEMS Microbiology Ecology* **15**, 321–336.
- [98] Lassen, C., Ploug, H., and Jorgensen, B. (1992) *FEMS Microbiology Ecology* **86**, 247–254.
- [99] Kounaves, S. and et al. (2009) *Journal of Geophysical Research* **114(E00A19)**, 1–20.
- [100] Hecht, M. and et al. (2009) *Science* **325**, 64–67.
- [101] Kounaves, S. and et al. (2010) *Journal of Geophysical Research* **115(E00E10)**, 1–16.
- [102] Kounaves, S. and et al. (2009) In 40th Lunar and Planetary Science Conference : .

- [103] Weetall, H. (1999) *Biosensors and Bioelectronics* **14**, 237–242.
- [104] Zahran, E., Gavalas, V., Valiente, M., and Bachas, L. (2010) *Anal. Chem.* **82**, 3622–3628.
- [105] Yoshida, Y., Matsui, M., Maeda, K., and Kihara, S. (1998) *Analytica Chimica Acta* **374**, 269–281.
- [106] Legin, A., Rudnitskaya, A., Legin, K., Ipatov, A., and Vlasov, Y. (2005) *Russian Journal of Applied Chemistry* **78(1)**, 89–95.
- [107] Bos, M., Bos, A., and van der Linden, W. (1993) *Analyst* **118**, 323–328.
- [108] Daponte, P. and Grimaldi, D. (1998) *Measurement* **23(2)**, 93–115.
- [109] Bos, M., Bos, A., and van der Linden, W. (1990) *Analytica Chimica Acta* **233**, 31–39.
- [110] Gallardo, J., Alegret, S., Muoz, R., De-Romn, M., Leija, L., Hernandez, P., and del Valle, M. (2003) *Analytical and Bioanalytical Chemistry* **377**, 248–256.
- [111] Gunnlaugsson, T., Glynn, M., Tocci, G., Kruger, P., and Pfeffer, F. (2006) *Coordination Chemistry Reviews* **250**, 3094–3117.
- [112] Peraud, K. and Dodd, G. (1982) *Nature* **299**, 352–355.
- [113] Ikegami, A. and Kaneyasu, M. June 1985 In Proceedings of the 3rd International Conference on Solid-State Sensors and Actuators Philadelphia, PA: . pp. 136–139.
- [114] Otto, M. and Thomas, J. (1985) *Analytical Chemistry* **57**, 2647–2651.
- [115] van der Linden, W., Bos, M., and Bos, A. (1989) *Analytical Proceedings* **26**, 329–335.
- [116] Legin, A., Rudnitskaya, A., Vlasov, Y., Di Natale, C., Mazzone, E., and D’Amico, A. (1999) *Electroanalysis* **11(10–11)**, 814–820.
- [117] Legin, A., Rudnitskaya, A., Vlasov, Y., Di Natale, C., Mazzone, E., and D’Amico, A. (2000) *Sensors and Actuators B* **65**, 232–234.
- [118] Legin, A., Rudnitskaya, A., Lvova, L., Vlasov, Y., Di Natale, C., and D’Amico, A. (2003) *Analytica Chimica Acta* **484**, 33–44.
- [119] Lvova, L., De Angelis, G., Montieri, C., Primadei, T., Martinelli, E., Mazzone, E., Pede, A., Paolesse, R., Di Natale, C., and D’Amico, A. (2004) *Sensors* **1**, 233–235.
- [120] Gallardo, J., Alegret, S., and del Valle, M. (2005) *Talanta* **66(5)**, 1303–1309.
- [121] Lvova, L., Kim, S., Legin, A., Vlasov, Y., Yang, J., Cha, G., and Nam, H. (2002) *Analytica Chimica Acta* **468**, 303–314.
- [122] Moreno i Codinachs, L., Baldi, A., Merlos, A., Abramova, N., Ipatov, A., Jimenez-Jorquera, C., and Bratov, A. (2008) *IEEE Sensors Journal* **8(5)**, 608–615.
- [123] Legin, A., Rudnitskaya, A., Vlasov, Y., Di Natale, C., Davide, F., and D’Amico, A. (1997) *Sensors and Actuators B* **44**, 291–296.

- [124] Winquist, F., Wide, P., and Lundström, I. (1997) *Analytica Chimica Acta* **357**, 21–31.
- [125] Kikkawa, Y., Toko, K., and Yamafuji, K. (1993) *Sensors and Materials* **5(4)**, 83–90.
- [126] Ciosek, P., Augustyniak, E., and Wroblewski, W. (2004) *Analyst* **129**, 639–644.
- [127] Ciosek, P. and Wroblewski, W. (2006) *Talanta* **69**, 1156–1161.
- [128] Toko, K., Murata, T., Matsuno, T., Kikkawa, Y., and Yamafuji, K. (1992) *Sensors and Materials* **4(3)**, 145–151.
- [129] Baxt, W. (1992) *Neural Computation* **4**, 772–780.
- [130] Winquist, F., Bjorklund, R., Krantz-Ricker, C., Lundström, I., Ostergren, K., and Skoglund, T. (2005) *Sensors and Actuators B* **111–112**, 299–304.
- [131] Di Natale, C., Davide, F., Brunink, J., D’Amico, A., Vlasov, Y., Legin, A., and Rudnitskaya, A. (1996) *Sensors and Actuators B* **34**, 539–542.
- [132] Mortensen, J., Legin, A., Ipatov, A., Rudnitskaya, A., Vlasov, Y., and Hjuler, K. (2000) *Analytica Chimica Acta* **403**, 273–277.
- [133] Rudnitskaya, A., Ehlert, A., Legin, A., Vlasov, Y., and Buttgenbach, S. (2001) *Talanta* **55**, 425–431.
- [134] Mourzina, Y., Schubert, J., Zander, W., Legin, A., Vlasov, Y., Luth, H., and Schoning, M. (2001) *Electrochimica Acta* **47**, 251–258.
- [135] Mimendia, A., Legin, A., Merkoci, A., and del Valle, M. (2010) *Sensors and Actuators B* **146**, 420–426.
- [136] Mimendia, A., Gutierrez, J., Opalski, L., Ciosek, P., Wroblewski, W., and del Valle, M. (2010) *Talanta* **82**, 931–938.
- [137] Di Natale, C., Macagnano, A., Davide, F., D’Amico, A., Legin, A., Vlasov, Y., Rudnitskaya, A., and Selezenev, B. (1997) *Sensors and Actuators B* **44**, 423–428.
- [138] Cortina, M., Ecker, C., Calvo, D., and del Valle, M. (2008) *Journal of Pharmaceutical and Biomedical Analysis* **46**, 213–218.
- [139] Mikhaleva, N. and Kulapina, E. (2006) *Electroanalysis* **18(13–14)**, 1389–1395.
- [140] Baret, M., Massart, D., Fabry, P., Conesa, F., Eichner, C., and Menardo, C. (2000) *Talanta* **51**, 863–877.
- [141] Cortina-Puig, M., Muñoz-Berbel, X., Alonso-Lomillo, M., Muñoz-Pascual, F., and del Valle, M. (2007) *Talanta* **72**, 774–779.
- [142] Cortina, M., Gutes, A., Alegret, S., and del Valle, M. (2005) *Talanta* **66**, 1197–1206.
- [143] Gallardo, J., Alegret, S., de Roman, M., Muñoz, R., Hernandez, P., Leija, L., and del Valle, M. (2003) *Analytical Letters* **36(14)**, 2893–2908.
- [144] Chang, C., Saad, B., Srurif, M., Ahmad, M., and Shakaff, A. (2008) *Sensors* **8**, 3665–3677.

- [145] Gallardo, J., Alegret, S., and del Valle, M. (2004) *Sensors and Actuators B* **101**, 72–80.
- [146] Durn, A., Cortina, M., Velasco, L., Rodriguez, J., Alegret, S., and del Valle, M. (2006) *Sensors* **6**, 19–29.
- [147] Gallardo, J., Alegret, S., Munoz, M., Leija, L., Hernandez, P., and del Valle, M. (2005) *Electroanalysis* **17**, 348–355.
- [148] Cortina, M., Duran, A., Alegret, S., and del Valle, M. (2006) *Analytical and Bioanalytical Chemistry* **385**, 1186–1194.
- [149] Calvo, D., Duran, A., and del Valle, M. (2007) *Analytica Chimica Acta* **600**, 97–104.
- [150] Calvo, D., Duran, A., and del Valle, M. (2008) *Sensors and Actuators B* **131**, 77–84.
- [151] Legin, A., Smirnova, A., Rudnitskaya, A., Lvova, L., Suglobova, E., and Vlasov, Y. (1999) *Analytica Chimica Acta* **385**, 131–135.
- [152] Ciosek, P., Brozozka, Z., Wroblewski, W., Martinelli, E., Di Natale, C., and D’Amico, A. (2005) *Talanta* **67**, 590–596.
- [153] Duarte, L., Jutten, C., and Moussaoui, S. (2009) *Lecture Notes in Computer Science* **5441/2009**, 662–669.
- [154] Duarte, L., Jutten, C., and Moussaoui, S. (2009) *IEEE Sensors Journal* **9(12)**, 1763–1771.
- [155] Darder, M., Valera, A., Nieto, E., Colilla, M., Fernandez, C., Romero-Aranda, R., Cuartero, J., and Ruiz-Hitzky, E. (2009) *Sensors and Actuators B* **135**, 530–536.
- [156] Cartas, R., Mimendia, A., Legin, A., and del Valle, M. (2011) *Electroanalysis* **23**, 953–961.
- [157] Kim, D., Goldberg, I., and Judy, J. (2007) *The Analyst* **132**, 350–357.
- [158] Kappes, T., Schnierle, P., and Hauser, P. (1999) *Analytica Chimica Acta* **393**, 77–82.
- [159] Gutierrez, M., Alegret, S., Caceres, R., Casadesus, J., Marfa, O., and del Valle, M. (2007) *Computers and Electronics in Agriculture* **57**, 12–22.
- [160] Gutierrez, M., Alegret, S., Caceres, R., Casadesus, J., Marfa, O., and del Valle, M. (2008) *J. Agric. Food Chem.* **56**, 1810–1817.
- [161] Gutierrez, M., Gutierrez, J., Alegret, S., Leija, L., Hernandez, P., Favari, L., Munoz, R., and del Valle, M. (2008) *Intern. J. Environ. Anal. Chem.* **88(2)**, 103–117.
- [162] Sghaier, K., Barhoumi, H., Maaref, A., Siadat, M., and Jaffrezic-Renault, N. (2011) *Journal of Water Resources and Protection* **3**, 531–539.
- [163] Taboada-Castro, T., Dieguez, A., Lopez, B., and Paz, A. (2000) *Communications in Soil Science and Plant Analysis* **31(11–14)**, 1993–2005.
- [164] Chan, W., Lee, A., Kwong, D., Liang, Y., and Wang, K. (1997) *Analyst* **122**, 657–661.

- [165] Richards, E., Bessant, C., and Saini, S. (2002) *Chemometrics and Intelligent Laboratory Systems* **61**, 35–49.
- [166] Garcia-Villar, N., Saurina, J., and Hernandez-Cassou, S. (2001) *Fresenius Journal of Analytical Chemistry* **371**, 1001–1008.
- [167] Mueller, A. and Hemond, H. (2011) *Analytica Chimica Acta* **590**, 71–78.
- [168] IUPAC (1976) *Pure Applied Chemistry* **48**, 127–132.
- [169] Macca, C. (2004) *Analytica Chimica Acta* **512**, 183–190.
- [170] Allen, J., Florido, A., Young, S., Dotinert, S., and Buchas, L. (1995) *Electroanalysis* **7(8)**, 710–713.
- [171] Srivastav, R., Sudheer, K., and Chaubey, I. (2007) *Water Resources Research* **43(W10407)**, 1–12.
- [172] Almeida, L. and Ludermir, T. (2010) *Neurocomputing* **73**, 1438–1450.
- [173] Abraham, A. (2004) *Neurocomputing* **56**, 1–38.
- [174] Richards, E., Bessant, C., and Saini, S. (2003) *Sensors and Actuators B* **88**, 149–154.
- [175] American Public Health Association (1971) *Standard Methods for Examination of Water and Waste Water*, American Public Health Association, Inc., New York 13th edition.
- [176] Carlson, R. (1978) *Anal. Chem.* **50(11)**, 1528–1531.
- [177] Robert, C. and Casella, G. (2004) *MonteCarlo Statistical Methods*, Springer Science, New York, NY 2nd edition.
- [178] Hayashi, M. (2004) *Environmental Monitoring and Assessment* **96**, 119–128.
- [179] Bakhary, N., Hao, H., and Deeks, A. (2007) *Engineering Structures* **29**, 2806–2815.
- [180] Patel, A. and Kosko, B. (2009) *Neural Networks* **22**, 697–706.